

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA ELETRÔNICA E
DE COMPUTAÇÃO

Sistema para Transferência de Dados Experimentais
Armazenados em Grid

Autor: _____
Rafael Sales Santelli da Silva

Orientador: _____
Carmen Lucia Lodi Maidantchik

Co-orientador: _____
José Manoel de Seixas

Examinador: _____
Antônio Cláudio Gómez de Sousa

Examinador: _____
Marcelo Luiz Drumond Lanza

Examinador: _____
Miriam Gandelman

DEL
Novembro de 2005

1 INTRODUÇÃO.....	IV
1.1 Motivação.....	v
1.2 Realizações do Projeto.....	vi
1.3 Organização do Documento.....	vi
2 A FÍSICA DE ALTAS ENERGIAS.....	7
2.1 O CERN.....	7
2.2 O Detector Atlas e seus Subsistemas.....	8
2.3 Armazenamento e Acesso aos Dados Adquiridos.....	10
2.4 Objetivos do Projeto GridTransfer.....	12
3 AS VERSÕES TILETRANSFER E COMBINEDTRANSFER.....	13
3.1 A Versão TileTransfer.....	13
3.1.1 Análise e Projeto do Sistema TileTransfer.....	17
3.1.1.1 O Módulo Início.....	18
3.1.1.2 O Módulo Busca.....	19
3.1.1.3 O Módulo Download.....	20
3.1.1.4 O Módulo Validação.....	21
3.1.1.5 O Módulo Link_email.....	21
3.1.1.6 O Módulo Link_página.....	22
3.1.1.7 O Módulo Executa_comandos.....	22
3.2 A Versão CombinedTransfer.....	24
3.2.1 Interface CombinedTransfer-AMI.....	25
4 TECNOLOGIAS UTILIZADAS.....	28
4.1 A Linguagem C.....	28
4.2 A Linguagem PHP.....	28
4.3 A Linguagem Javascript.....	29
4.4 O Banco de Dados MySQL.....	29
4.5 A linguagem XML.....	29
5 TRANSFERÊNCIA DE ARQUIVOS.....	31
5.1 BBFTP.....	31
5.2 WGET.....	31
5.3 Getright.....	32
5.4 Download Accelerator.....	32
6 GRIDS COMPUTACIONAIS.....	33
6.1 Projeto Globus.....	33
6.1.1 O Protocolo de Transferência GridFTP.....	34
6.1.2 Segurança GSI.....	34
6.1.2.1 Criptografia Assimétrica.....	35
6.1.2.2 Certificados.....	35
6.1.2.3 Comunicação entre Usuários Certificados.....	35
6.2 Projeto LCG.....	36
6.2.1 Obtendo o Certificado.....	36

6.2.2 ATLAS Data Challenge.....	37
6.3 Ferramentas Grid.....	37
6.3.1 Dirac.....	37
6.3.2 Gilda.....	38
6.3.3 Grenade.....	39
6.3.4 Don Quijote.....	39
6.3.5 Edg (European DataGrid).....	40
7 O SISTEMA GRIDTRANSFER.....	41
7.1 Proposta de Sistema.....	41
7.2 Análise de Requisitos dos Usuários.....	43
7.3 Especificação do Sistema	45
7.3.1 Análise dos Protocolos de Transferência.....	45
7.3.2 Ferramenta de Gerenciamento de Réplicas.....	46
7.3.3 Requisitos para Utilização do Sistema GridTransfer.....	49
7.3.3.1 Opções para Transferência de Arquivos.....	49
7.3.4 Diagramas de Fluxo de Dados.....	52
7.3.4.1 Diagrama de Contexto.....	52
7.3.4.2 Diagrama de Fluxo de Dados Nível 1.....	53
7.3.4.3 Diagrama de Fluxo de Dados Nível 2.....	54
7.3.4.4 Diagrama de Fluxo de Dados Nível 3.....	55
7.3.4.5 Diagrama de Fluxo de Dados Nível 4.....	56
7.4 O Projeto e a Implementação.....	57
7.4.1 Decomposição em Módulos.....	57
7.4.1.1 Arquitetura do Sistema.....	57
7.4.1.2 Sistema GridTransfer.....	58
7.4.1.3 Módulo Início.....	59
7.4.1.4 Módulo Pesquisa.....	60
7.4.1.5 Módulo Acesso.....	61
7.4.1.6 Módulo Resultado _pagina.....	62
7.4.1.7 Módulo Resultado _email.....	63
7.4.2 Projeto Detalhado e Implementação.....	64
7.4.2.1 Módulo Início.....	64
7.4.2.2 Módulo Pesquisa.....	65
7.4.2.3 Módulo Acesso.....	66
7.4.2.4 Módulo Resultado _pagina.....	66
7.4.2.5 Módulo Resultado _email.....	68
7.5 Execução.....	69
8 CONCLUSÕES.....	80
8.1 Trabalhos Futuros.....	80
9 GLOSSÁRIO.....	82
10 REFERÊNCIAS BIBLIOGRÁFICAS.....	84
11 ANEXO A.....	87
12 ANEXO B.....	89

<u>13 ANEXO C.....</u>	<u>92</u>
<u>14 ANEXO D.....</u>	<u>93</u>
<u>15 APÊNDICE.....</u>	<u>97</u>

Resumo

Este projeto final apresenta um sistema Web para a transferência de dados, armazenados em Grids computacionais, resultantes de arranjos experimentais realizados em um detector de partículas para posterior análise dos pesquisadores da comunidade ATLAS do laboratório europeu de física de altas energias (CERN).

O detector ATLAS entrará em funcionamento apenas em 2007 e, portanto, seus módulos e subsistemas devem ser testados e integrados. Para isso, é necessário um trabalho conjunto de testes e análise dos resultados pelos pesquisadores. Como a comunidade é distribuída por vários países, tornou-se necessário um sistema que facilitasse o acesso aos arquivos para posterior análise.

Inicialmente o sistema foi projetado e implementado para transferir dados de apenas um subsistema do detector ATLAS, o calorímetro de telhas. Por esse motivo, o sistema foi denominado TileTransfer. Com o sucesso do sistema, ele foi estendido para oferecer acesso aos dados de experimentos de outros subsistemas como também a testes combinados entre os módulos do detector. Essa versão do sistema foi denominada CombinedTransfer.

Porém, a partir de 2004, com o aumento do volume de dados e a necessidade de maior poder de processamento, foi implementada no CERN uma Grid, cujo o intuito é o compartilhamento de recursos proporcionando maior poder computacional e armazenamento distribuído entre os diversos institutos de pesquisa.

Para fornecer acesso aos dados dos experimentos armazenados em Grid, o sistema CombinedTransfer foi modificado. Os experimentos do Data Challenge, cujo objetivo era testar a infra-estrutura e as ferramentas Grid desenvolvidas nos últimos anos e validar o modelo computacional proposto, foram utilizados para os testes do novo sistema GridTransfer, que é o objeto principal deste projeto final.

Este relatório é composto de uma introdução, onde é explicada a motivação, o ambiente CERN e os experimentos deste laboratório. Depois, um histórico sobre a evolução do sistema TileTransfer até o GridTransfer é mostrado, seguido de um relato sobre as tecnologias estudadas para compor o sistema. Finalmente, a especificação, projeto e resultados obtidos com a implementação do sistema GridTransfer são descritos, seguido de trabalhos futuros e conclusões.

Palavras-chave:

- Sistema Web
- Grid computacional

- Transferência de dados
- Engenharia de Software
- Armazenamento

1 Introdução

Com o aumento da competição global e as profundas alterações que a economia mundial vem sofrendo, as organizações, para sobreviver e atingir seus objetivos, devem investir cada vez mais em inovação e pesquisa. Da mesma forma, pelo fato dos recursos serem escassos, precisam também controlar seus gastos com o desenvolvimento e operação de suas atividades.

Para conseguir responder de forma rápida e eficiente às novas demandas, as organizações vêm sofrendo mudanças estruturais e comportamentais. Uma delas é o surgimento das organizações virtuais, que são novos modelos estruturais através da união e cooperação de esforços inter-organizacionais para atingir um objetivo comum.

Através da tecnologia da informação, as organizações podem compartilhar recursos, dados, tecnologia, riscos, custos e mercado aumentando a sinergia de suas atividades e possibilitando atingir resultados superiores à atuação individual de cada uma delas. Pode-se acrescentar que algumas atividades de tecnologia de ponta podem ter requisitos de recursos com que dificilmente uma única instituição poderia arcar.

As grandes instituições de pesquisa também seguem esta tendência. As suas atividades geram grande volume de dados e, por isso, necessitam de amplo poder computacional para processar e gerar resultados a serem analisados por uma comunidade de cientistas espalhados pelo mundo. Pode-se citar, como exemplo, as pesquisas nas áreas de física de altas energias, medicina, análise do solo, astrofísica, entre outros.

Dentro deste contexto, um ambiente de compartilhamento de recursos (armazenamento, processamento, instrumentos científicos) pode ampliar a capacidade que uma instituição de pesquisa tem para realizar suas atividades e atingir seus objetivos. Além disso, são inúmeros os benefícios, como maior integração entre a comunidade científica, possibilitando intercâmbio de conhecimento, e economia de custos, uma vez que, através da utilização dos recursos de toda a comunidade, a instituição poderia diminuir seus próprios gastos na aquisição de equipamentos. Por exemplo, um cientista poderia compartilhar seu computador para processar uma determinada quantidade de dados enquanto não o estivesse usando, no período da noite.

Uma resposta nesta direção é o conceito de Grid computacional, que é um modelo de computação distribuída. As Grids promovem um ambiente de compartilhamento e integração de recursos autônomos em uma comunidade heterogênea e dinâmica, possibilitando a análise dos resultados das atividades realizadas com produção de dados em larga escala.

A complexidade dos ambientes Grids pode ser analisada pela composição de diferentes institutos, recursos, políticas operacionais e de segurança, além da constante evolução e mudança que sofrem essas organizações. Com a tendência de crescimento no volume de geração e da

necessidade de armazenamento dos dados, de forma distribuída, a busca, acesso e processamento se tornam um problema.

Nestes ambientes complexos de computação, um fato importante a ser considerado é a transferência dos dados, já que, mesmo com a melhoria na velocidade das redes de telecomunicações, o volume de dados a transferir é, tipicamente, enorme e deve trafegar constantemente entre os computadores dos colaboradores geograficamente dispersos. A complexidade no acesso e transferência dos dados gerados nas atividades deve ser minimizada, uma vez que o pesquisador deseja se concentrar nos resultados e não aprendendo formas de utilizar os recursos necessários para a obtenção dos mesmos.

Desta forma, o acesso a bancos de dados, que contêm informações das atividades realizadas, a localização e disponibilidade de recursos e a transferência eficiente de dados são requisitos para a realização das análises científicas.

1.1 Motivação

Este projeto final está inserido na colaboração internacional que a UFRJ tem com uma instituição que realiza pesquisa em física de altas energias (CERN). O CERN é a instituição líder na Europa para o desenvolvimento de modelos computacionais baseados em Grids. Através de um acelerador de partículas e quatro detectores, dentre eles o ATLAS, o CERN se prepara para realizar experimentos com colisões de partículas com níveis de energia jamais alcançados.

O detector ATLAS entrará em operação em 2007. O processo de construção é bastante complexo, sendo necessários testes individuais e em conjunto dos subsistemas, gerando a necessidade de calibração dos módulos do ATLAS. Os dados resultantes devem ser analisados pelos quase 2000 colaboradores espalhados pelo mundo.

A incidência de feixes de partículas no detector ATLAS gera um enorme fluxo de dados quando se realizam os testes experimentais. O detector é subdividido em vários subsistemas e possui dezenas de milhões de canais de aquisição de dados. O volume de dados anual gerado pelo detector ATLAS será equivalente a 6 milhões de cds de 600 Megabytes. Para analisar esses dados, seria necessário um poder computacional equivalente a cem mil computadores pessoais trabalhando o dia todo. A comunidade deve compartilhar seus recursos computacionais, provenientes de diferentes plataformas, para proporcionar o processamento da informação necessária para a realização das análises dos experimentos realizados.

A complexidade deste processo exige que toda a comunidade participe do processo de análise dos resultados dos testes experimentais. Como o volume de dados atinge proporções muito grandes, observou-se a necessidade de descentralizar o armazenamento dos arquivos entre outros servidores espalhados em uma Grid computacional, o que aumentou a complexidade para a busca, acesso e transferência dos arquivos resultantes dos arranjos experimentais. Além disso, o tráfego de dados pela rede foi intensificado uma vez que o fluxo de dados entre os recursos da Grid é muito grande. Desta forma, deve-se buscar formas mais eficientes de transferência de arquivos.

1.2 Realizações do Projeto

Acompanhando o desenvolvimento do ATLAS e analisando os problemas citados, foi feito um estudo do ambiente e da complexidade, relacionando ferramentas e tecnologias capazes de solucioná-lo. Após este estudo, foi implementado um sistema para a transferência dos arquivos gerados nos arranjos experimentais no detector e armazenados em Grid. Como a comunidade se encontra espalhada entre diversos institutos de pesquisa e possui recursos heterogêneos e diferentes plataformas, o sistema foi desenvolvido para ser independente da plataforma, de modo que todos possam utilizá-lo.

O sistema promove, de forma amigável, o acesso às bases de dados que contêm as informações e condições dos arranjos experimentais realizados nos testes e na calibração do ATLAS, através de ferramentas de busca. Além disso, o sistema possui diversas funcionalidades como a busca, acesso e seleção dos arquivos de dados gerados nos arranjos experimentais armazenados em Grid, além de permitir métodos de transferência mais eficientes, uma vez que a quantidade de dados a ser transferida é volumosa.

1.3 Organização do Documento

Este relatório é composto de uma introdução onde é mostrado o contexto geral e específico em que se situa o projeto e a motivação no seu desenvolvimento. Depois é feita uma introdução sobre a física de altas energias, o CERN e o detector ATLAS. Neste capítulo também é citado o problema do armazenamento e acesso aos dados e, depois, os objetivos do sistema. No próximo capítulo será feito um histórico apresentando as versões anteriores do sistema, o contexto e as tecnologias que foram utilizadas nos seus desenvolvimentos. No capítulo seguinte, é relatado o estudo do ambiente Grid e das ferramentas e tecnologias estudadas a fim de implementar o sistema. Depois, é relatada a análise, especificação e implementação do sistema GridTransfer identificando o que foi feito no projeto. No próximo capítulo são analisados os próximos trabalhos a serem desempenhados e as conclusões deste projeto final. Este relatório é composto também de um glossário, referências bibliográficas, anexos e um apêndice contendo o que foi publicado durante o estudo e desenvolvimento do sistema.

2 A Física de Altas Energias

Nesta seção é apresentada uma introdução ao contexto do projeto e ao ambiente para o qual o sistema computacional foi desenvolvido.

2.1 O CERN

O CERN 10 (European Laboratory for Particle Physics), situado na fronteira da Suíça com a França, é um laboratório científico onde são realizados grandes experimentos em física de partículas. Este conta com a participação de uma comunidade de cientistas de diversos institutos e universidades espalhados pelo mundo, que analisam os experimentos realizados por este laboratório. São vinte países membros e vinte e oito não membros envolvidos nos projetos de pesquisa do CERN. Com a colaboração desta comunidade de cientistas, são construídos equipamentos para que os físicos realizem seus experimentos.

Com o objetivo de estudar as partículas que compõem a matéria e as forças de interação entre elas, o CERN realiza experimentos de colisão de partículas. Para isso, constrói aceleradores e detectores de partículas. Os aceleradores são máquinas que levam as partículas a velocidades muito próximas à da luz e, então, as fazem colidir contra outras partículas. Os detectores são instrumentos feitos para captar os resultados das colisões para que os físicos possam observar o que aconteceu durante os experimentos.

Atualmente, o CERN está engajado em um grande projeto: o LHC (Large Hadron Collider), um enorme acelerador de partículas. Acoplado ao LHC, funcionará o detector ATLAS 10 (A Toroidal LHC Apparatus), projeto que está sendo desenvolvido com a participação da UFRJ, através da COPPE, da Escola de Engenharia e do Instituto de Física.

Os físicos usam os aceleradores para concentrar mais energia em partículas muito pequenas para elevar seu momento linear, reduzindo assim seu comprimento de onda a ponto de penetrar no núcleo dos átomos. Campos eletromagnéticos aceleram as partículas carregadas que irão colidir contra um alvo fixo ou com partículas de um outro feixe em sentido contrário. Cada colisão resulta em um evento físico a ser analisado de acordo com o interesse do experimento. Os eventos são extremamente complexos, pois produzem muitas partículas, das quais a grande maioria não pode ser detectada diretamente, devido ao seu curto tempo de vida. Por isso, os físicos tipicamente precisam observar os produtos do decaimento dessas partículas e, a partir destes, deduzir sua existência. Um conjunto de detectores é então colocado ao redor dos pontos de colisões para permitir a leitura das partículas resultantes e registrar os eventos ocorridos.

Os detectores são subdivididos em várias camadas envolvendo o ponto de colisão. Cada camada é especializada na detecção de um determinado grupo de partículas. O LHC levará nuvens de prótons a energia de aproximadamente 14TeV (Terá-elétron-Volts). Isso permitirá aos cientistas se aprofundar ainda mais na estrutura da matéria e reproduzir condições semelhantes as que teriam ocorrido no Big Bang, a grande explosão que deu origem ao universo. O LHC está situado próximo à cidade de Genebra na Suíça e tem aproximadamente 27Km de circunferência. A Figura 1 ilustra o LHC com seus detectores, dentre eles o ATLAS.

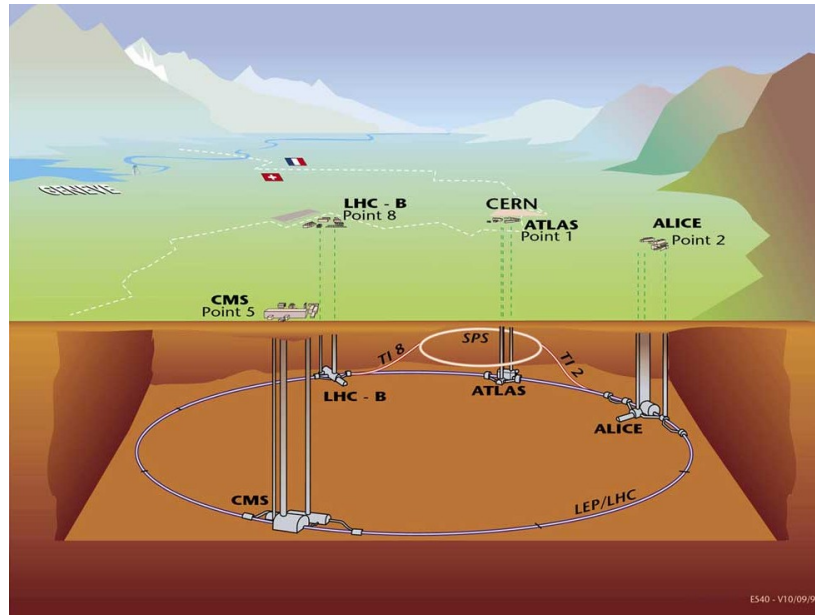


Figura 1 - O LHC e seus detectores. Extraído de 10

2.2 O Detector Atlas e seus Subsistemas

O Detector ATLAS, experimento científico que irá operar no LHC, é uma colaboração internacional envolvendo aproximadamente 2000 físicos de 150 institutos de 34 países do mundo. Seu objetivo básico é realizar novas descobertas acerca da natureza fundamental da matéria e das forças básicas existentes no universo. Uma das metas do detector ATLAS é verificar a existência do bóson de Higgs, partícula que explicaria a diferença de massa entre as partículas elementares.

O detector ATLAS tem o tamanho aproximado de um prédio de cinco andares. Ele é composto por vários sub-detectores, cada um com uma função específica no experimento, tais como medição da energia incidente do feixe de partículas, ângulo de incidência, entre outros. O diagrama esquemático da figura 2 mostra o detector e seus principais componentes. Cada um deles capta um conjunto de partículas diferentes. Eles estão organizados de dentro para fora no detector, assim as partículas os atravessam sequencialmente. Alguns destes componentes do ATLAS, bem como suas funções, são:

- Calorímetro eletromagnético: mede a energia total de elétrons, pósitrons e fótons.
- Calorímetro hadrônico: mede a energia total de hádrons.

- Detector de múons: encontra-se na parte mais externa do detector, onde apenas múons e neutrinos conseguem chegar.

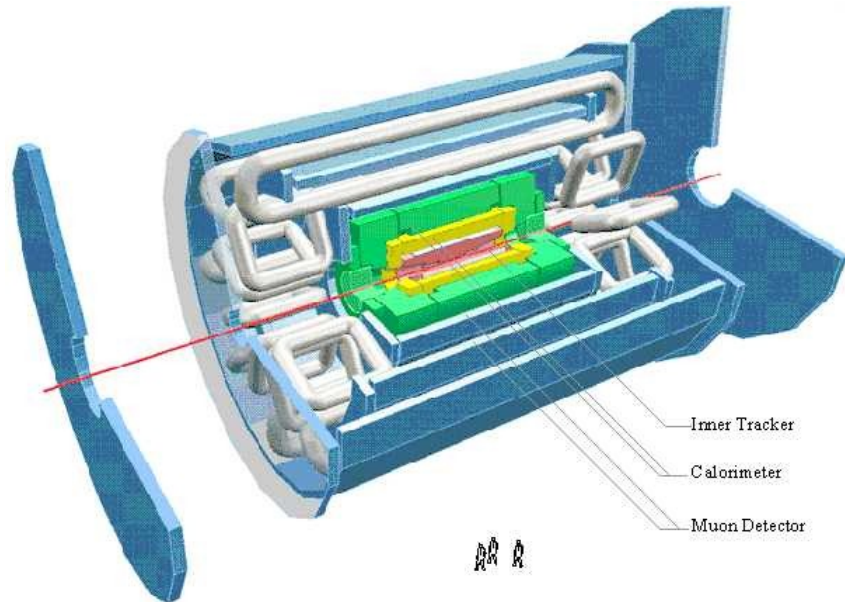


Figura 2 - Diagrama esquemático do detector ATLAS. Extraído de 10

O calorímetro de telhas 10 é um dos sub-detectores do ATLAS, projetado para operar no LHC. É um calorímetro hadrônico que usa aço como material absorvedor de energia e telhas cintilantes como material ativo. O calorímetro de telhas é constituído de uma estrutura cilíndrica com um raio interno de 2280 mm e um raio externo de 4230 mm. Ele é subdividido em um barril central (barrel) de 5640 mm de comprimento e dois barris estendidos (extended barrels) de 2910 mm de comprimento como mostra a figura 3.

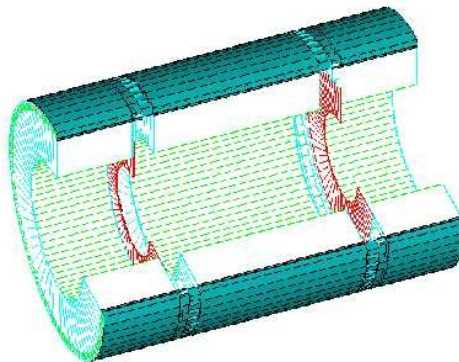


Figura 3 - Visão do Barril Central e Barris Estendidos do calorímetro de telhas Extraído de 10

Cada barril é composto por 64 módulos em formato de cunha. Entre os barris externos e o barril central existe um vão de 600 mm para o acesso de cabos e dispositivos eletrônicos aos sub-detecores internos. As cascatas de partículas, resultantes da interação de uma partícula com o absorvedor, causam emissão de luz pelas telhas cintilantes. Para cada telha, há um par de fibras (uma em cada extremidade) que coletam a luz emitida pela telha. Células de leitura são, então, definidas pelo agrupamento de um conjunto de fibras em um foto-multiplicador (PMT) 10.

O processo de construção do detector ATLAS é complexo. Cada subsistema é testado individualmente e depois calibrado para funcionar de acordo com as especificações. Por exemplo, se um feixe de partículas incidiu em um campo magnético e foi desviado de forma diferente do ângulo determinado, o subsistema deverá ser recalibrado e retestado. Depois dos testes individuais, são feitos testes combinados, ou seja, com subsistemas sendo testados conjuntamente. Depois dos testes em conjunto, a próxima fase é a de comissionamento, onde os subsistemas são postos dentro do detector ATLAS na caverna onde ele entrará em operação. Nesta fase também serão feitos testes. Só depois o detector será posto em operação em 2007. Em todas as fases, os pesquisadores, espalhados pelo mundo, devem acessar e analisar os dados resultantes dos testes.

2.3 Armazenamento e Acesso aos Dados Adquiridos

O grande volume de dados gerados e a complexidade do processo de análise exigem que os dados resultantes dos arranjos experimentais realizados no detector sejam analisados por toda a comunidade, independente da sua localização geográfica e disponibilidade de recursos.

Os dados resultantes dos testes de feixe de partículas são adquiridos por sistemas de aquisição de dados, analisados previamente por físicos e colocados em arquivos chamados "ntuples" 10. O armazenamento das ntuples era feito em um servidor especial chamado CASTOR (CERN Advanced STORAge Manager) 10. Dessa forma, os físicos de todo mundo acessavam o servidor CASTOR para adquirir os dados experimentais. O CASTOR é um servidor com um sistema de fitas e um robô que ao ser acessado se dirige à fita que contém o arquivo de dados requisitado e copia seu conteúdo para uma memória existente nesse servidor. A quantidade de fitas é enorme já que são armazenados cerca de 35 milhões de arquivos num total de aproximadamente 5000 Terabytes. A figura 4 mostra a evolução da demanda por armazenamento de dados no CASTOR apresentando o número de arquivos e espaço necessário para armazená-los.

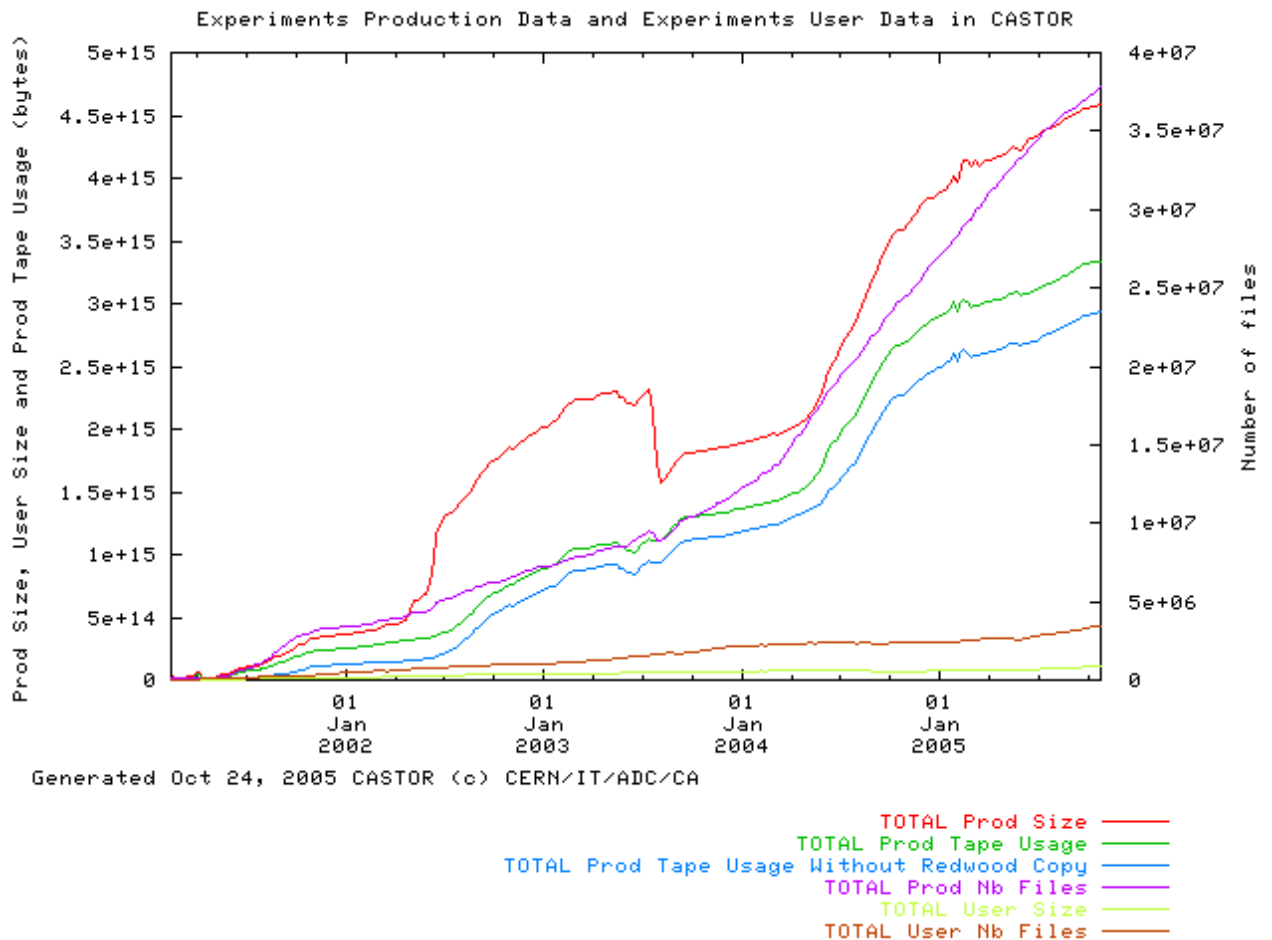


Figura 4 – Demanda por armazenamento de dados. Extraído de 10

A operação deste servidor é complexa resultando em uma grande quantidade de comandos específicos a serem executados. Pode-se perceber que um físico, inicialmente, não tem esse conhecimento e precisaria de algumas horas de seu trabalho para entender o funcionamento do CASTOR e obter os arquivos que necessita para fazer sua análise.

A partir de 2004, conforme o volume de dados foi atingindo proporções cada vez maiores, observou-se a necessidade de descentralizar o armazenamento dos arquivos entre outros servidores espalhados em uma Grid computacional (Capítulo 6), o que aumentou a complexidade para a busca e acesso aos arquivos resultantes dos testes. Além disso, o tráfego de dados pela rede foi intensificado. Desta forma, deve-se buscar formas mais eficientes de transferência de arquivos.

Cada fase de construção do detector gera um conjunto de testes a serem realizados e cada teste é constituído de um conjunto de eventos. Os testes são caracterizados pelas suas condições, tais como tipo de feixe, energia, ângulos de incidência entre outras características a serem determinados durante o processo de calibração. Os dados adquiridos nos testes e, futuramente,

durante o funcionamento do detector estarão armazenados de forma distribuída em uma Grid computacional. Serão inúmeros arquivos espalhados entre diversos institutos que irão armazená-los. A dificuldade começa no processo de busca, ao escolher qual arquivo deverá ser analisado. O acesso aos dados também é outro problema uma vez que cada instituto possui recursos e políticas de segurança diferentes. Além disso, o volume de transferência será bem maior, já que maior quantidade de dados será gerada, principalmente quando o detector entrar em funcionamento, e também os dados estarão armazenados não mais de forma centralizada (CASTOR) e sim espalhados.

2.4 Objetivos do Projeto GridTransfer

O principal objetivo do sistema GridTransfer é proporcionar acesso aos dados resultantes dos testes armazenados de forma descentralizada em Grids, realizados no detector ATLAS, de forma amigável, através de ferramentas de busca, seleção e transferência dos arquivos.

Alguns aspectos são importantes para atingir esse objetivo. Conforme mencionado, a comunidade espalhada entre diversos institutos de pesquisa possui recursos heterogêneos e diferentes plataformas. Sendo assim, o sistema deve ser independente de plataforma para que todos possam utilizá-lo. Além disso, fornecer acesso, através de uma ferramenta de busca, às informações das condições dos testes realizados é fundamental para que o pesquisador possa escolher qual teste deseja analisar.

Outro objetivo a ser alcançado corresponde à otimização do processo de transferência dos arquivos. Isso consiste em oferecer eficiência na transferência, recuperação de arquivos, se houver falha na conexão (reiniciar a transferência de onde parou), e melhores taxas de transferência com maior utilização da banda de transmissão.

3 As Versões TileTransfer e CombinedTransfer

Antes de apresentar o sistema GridTransfer, será feito nesta seção, um histórico do desenvolvimento das versões anteriores ao sistema. Uma breve descrição do sistema TileTransfer, sua composição em módulos e sua evolução para o sistema CombinedTransfer serão apresentadas.

Deve-se destacar que a versão TileTransfer foi desenvolvida no ano de 2003, trabalho realizado no contexto de iniciação científica. A versão CombinedTransfer porém, foi desenvolvida por outro colega, Kaio Karam ¹⁰, também em trabalho de iniciação científica. Foi necessário também realizar um estudo das ferramentas desenvolvidas no projeto CombinedTransfer, para depois agregar o estudo feito no desenvolvimento do GridTransfer.

3.1 A Versão TileTransfer

Foi desenvolvido inicialmente um sistema web, chamado Tiletransfer, que visava a facilitar o acesso aos arquivos que contém os dados resultantes dos testes de feixe de partículas no calorímetro de telhas do detector ATLAS do CERN.

A motivação para o projeto TileTransfer pode ser ilustrada pelas figuras 5 e 6, que apresentam as situações antes e depois da implementação do sistema. A figura 5 mostra a sequência de passos que o pesquisador deve seguir para obter o arquivo desejado. Ele deve consultar as páginas do CERN e CASTOR para obter instruções de como utilizar esse servidor, verificar os testes realizados decidindo qual deseja analisar, acessar o banco de dados que contém as informações das condições realizadas nos testes, conectar-se remotamente através do programa “ssh” para efetuar os comandos para disponibilizar e transferir os arquivos do CASTOR e, finalmente, efetuar a transferência dos arquivos pelo protocolo “ftp”.

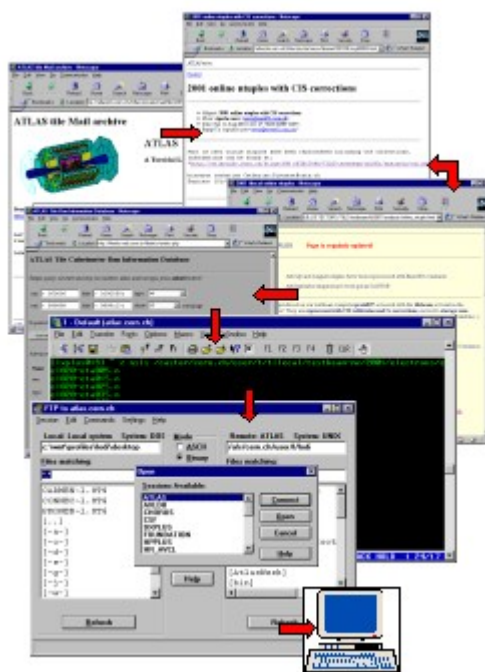


Figura 5 - Situação anterior ao surgimento do TileTransfer

A figura 6 ilustra a melhoria que o sistema TileTransfer proporcionou aos físicos e engenheiros, que precisam ter acesso aos dados resultantes das experiências no calorímetro de telhas do detector ATLAS.

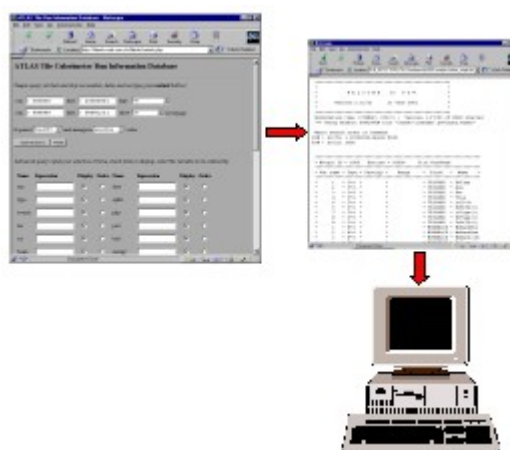


Figura 6 – Situação após a implementação do sistema

O TileTransfer é um sistema com interface Web e pode ser encontrado no endereço (<http://tileinfo.web.cern.ch/tileinfo/lps>). O processo de desenvolvimento, inicialmente, teve a participação do professor Marcelo Lanza 10. Em seguida, o projeto foi continuado por mim em contexto de iniciação científica. Ele foi instalado em uma máquina chamada pcata007.cern.ch que

contém servidor web e acesso ao CASTOR. Os arquivos podem ser transferidos do mesmo para esta máquina para fornecer um link hipertextual ao final do processo. Os arquivos são mantidos por algum tempo nesta máquina, uma vez que o tempo de acesso e transferência no CASTOR são demorados e mais de um pesquisador podem querer transferir este arquivo. Este processo de transferência do CASTOR é demorado porque este servidor possui os arquivos armazenados em fita que, ao ser solicitado, transfere os dados para disco. Esse processo é denominado “stage out”.

Inicialmente, o usuário acessa uma página HTML de busca onde pode inserir as condições dos testes, ou seja, as características que deseja encontrar nos arquivos resultantes dos experimentos físicos. A figura a seguir mostra a página de pesquisa do sistema TileTransfer. Pode-se verificar que são muitos parâmetros de condições de teste já que existem vários campos na página para inserção valores.

ATLAS Tile Calorimeter Run Information Database

Simple query: set start and stop run number, dates and run type, press **submit** button!

run > 050000 date > 20000501 type All

run < 999999 date < 99991231 show 25 rows/page

tb period Aug-2002 and arrange in ascending order

Submit Query Reset

Advanced query: input your selection criteria, check items to display, select the variable to be ordered by.

Name	Expression	Display	Order	Name	Expression	Display	Order
run		☒	☐	date		☒	☐
type		☒	☐	spills		☒	☐
events		☒	☐	phy		☒	☐
las		☒	☐	ped		☒	☐
cis		☒	☐	bad		☒	☐
beam		☒	☐	energy		☒	☐
mod		☒	☐	eta		☒	☐

Figura 7 – Página de pesquisa do sistema TileTransfer

Ao submeter a pesquisa, o sistema consulta o banco de dados Runinfo, que contém informações sobre os “runs” e disponibiliza no navegador os resultados das ntuples que atendem as condições desejadas em uma tabela. A figura a seguir mostra os resultados de uma pesquisa efetuada no sistema.

ATLAS Tile Run Information Database - Netscape

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location http://tileinfo.web.cern.ch/tileinfo/lps/v1.1/runselect.php

Instant Message Internet Lookup New&Cool Netcaster RealPlayer

What's Related

ATLAS Tile Calorimeter Run Information Database

SQL query string: select * from runinfo where run> 050000 and date> 20000501 and type="phys" and run< 999999 and date< 99991231 and tb="Aug-2002" order by run asc

Your query returned 590 rows with 24 fields. Displaying results from 1 to 25.

Click on a row number to view/modify run parameters. Click on a column name to sort data by that parameter.

	run	date	type	spills	events	phy	las	pcd	cis	bad	beam	energy	mod	eta	theta	cell	y	z	cap	dac	chl	ch2	tb	comment
1	220034	2002-08-15 02:09:02	phys	7	50000	49807	0	137	23	13	e	180.0	EB+	90.0	1	0.0	1.0	5	50	0.02	0.07	Aug-2002	WRONG RUN (events have w	
2	220035	2002-08-15 02:26:40	phys	7	50000	49833	0	137	17	13	e	180.0	EB+	90.0	1	0.0	1.0	5	50	0.02	0.07	Aug-2002	Cher2 pressure =	
3	220036	2002-08-15 03:08:38	phys	7	50000	49633	0	159	197	11	e	180.0	EB+	90.0	2	0.0	0.0	5	50	0.02	0.05	Aug-2002	Cher2 pressure =	
4	220037	2002-08-15 03:12:19	phys	7	50000	49598	0	154	237	11	e	180.0	EB+	90.0	3	0.0	0.0	5	50	0.02	0.05	Aug-2002		
5	220038	2002-08-15 03:20:34	phys	6	50000	49717	0	115	160	8	e	180.0	EB+	90.0	4	0.0	0.0	5	50	0.02	0.05	Aug-2002		
6	220039	2002-08-15 03:28:00	phys	7	50000	49621	0	156	214	9	e	180.0	EB+	90.0	5	0.0	0.0	5	50	0.02	0.05	Aug-2002		
7	220040	2002-08-15 03:31:08	phys	7	50000	49629	0	148	212	11	e	180.0	EB+	90.0	6	0.0	0.0	5	50	0.02	0.05	Aug-2002		
8	220041	2002-08-15 03:34:21	phys	7	50000	49669	0	138	181	12	e	180.0	EB+	90.0	7	0.0	0.0	5	50	0.02	0.05	Aug-2002		
9	220042	2002-08-15 03:37:28	phys	7	50000	49673	0	136	181	10	e	180.0	EB+	90.0	8	0.0	0.0	5	50	0.02	0.05	Aug-2002		
10	220043	2002-08-15 03:42:39	phys	6	50000	49717	0	120	155	8	e	180.0	EB+	90.0	9	0.0	0.0	5	50	0.02	0.05	Aug-2002		
11	220044	2002-08-15 03:52:40	phys	6	50000	49637	0	129	202	12	e	180.0	EB+	90.0	10	0.0	0.0	5	50	0.02	0.05	Aug-2002		
12	220045	2002-08-15 03:56:56	phys	7	50000	49698	0	118	174	10	e	180.0	EB+	90.0	10	0.0	1.0	5	50	0.02	0.05	Aug-2002		
13	220046	2002-08-15 04:30:28	phys	7	50000	49668	0	128	196	8	e	180.0	EB+	90.0	11	0.0	0.0	5	50	0.02	0.05	Aug-2002		
14	220047	2002-08-15 04:40:16	phys	7	50000	49609	0	152	222	17	e	180.0	EB+	90.0	11	0.0	0.0	5	50	0.02	0.05	Aug-2002		
15	220049	2002-08-15 05:02:26	phys	2	10000	9920	0	32	43	5	e	180.0	EB+	90.0	11	0.0	0.0	5	50	0.02	0.05	Aug-2002		

Document: Done

Figura 8 – Tabela com os resultados da pesquisa

O sistema oferece duas formas para a obtenção dos arquivos. Uma delas é através do próprio navegador, quando o usuário seleciona “download now”. Ao efetuar essa operação, o sistema irá primeiramente verificar se o arquivo já existe na máquina pcata007 o que ocorrerá se alguém já requisitou esse arquivo anteriormente. Isso é feito para poupar tempo já que as transferências de arquivo do CASTOR são demoradas. Caso contrário, o sistema executará todos os comandos necessários para se conectar ao CASTOR e transferir os arquivos pedidos para a máquina pcata007. O usuário pode acompanhar todas as operações realizadas durante o processo já que o sistema apresenta mensagens no navegador. Finalizada a operação, o sistema cria ligações hipertextuais (links) onde o usuário pode fazer o download dos arquivos. A figura 9 mostra a página do sistema ao finalizar a operação.

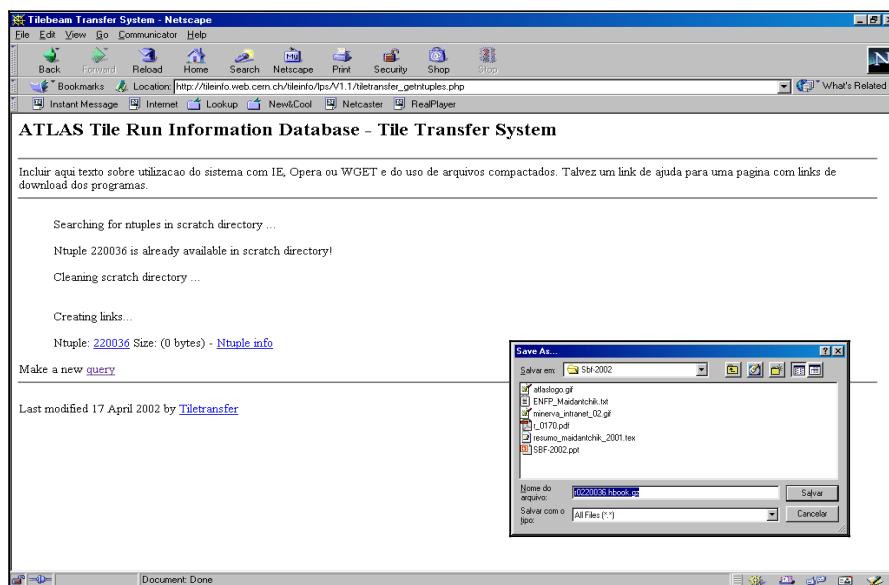


Figura 9 – O sistema fornece o link ao usuário

A outra forma de obtenção dos arquivos é por e-mail. O usuário seleciona os arquivos que deseja e preenche no campo e-mail seu endereço eletrônico. Ao selecionar “download later”, o sistema cria um processo filho no sistema operacional que irá executar as mesmas operações que em “download now” só que o usuário não precisará esperar a execução dos comandos, podendo inclusive fechar a janela do navegador. Os resultados (mensagens da execução e links) serão recebidos em seu e-mail. A grande vantagem dessa funcionalidade é evitar o tempo de processamento através do navegador.

Pode-se concluir que o sistema TileTransfer integra todos os passos necessários para armazenar, procurar e acessar os dados resultantes dos experimentos físicos do calorímetro de telhas do detector ATLAS do CERN, facilitando o processo inicial da análise ao evitar que os usuários precisem lembrar da sintaxe e executem cada comando separadamente. O TileTransfer é integrado ao banco de dados, denominado “Runinfo” que contém todas as características de cada experimento. O resultado é apresentado através de uma página hipertextual onde o usuário pode selecionar mais de um arquivo. Todos os atributos são apresentados em uma tabela o que facilita a seleção dos arquivos desejados. Como os arquivos são comprimidos pelo sistema, o tempo de transferência é reduzido. Ao oferecer uma interface amigável na Web, o sistema também facilita a sua utilização e oferece ligações hipertextuais que descrevem as características observadas nos dados e informações relacionadas de utilidade aos pesquisadores. A arquitetura do TileTransfer integra diferentes recursos de forma transparente ao usuário. O sistema também permite a paralelização do processamento para obtenção dos arquivos uma vez que é criado um processo filho para cada arquivo na opção “download later”.

3.1.1 Análise e Projeto do Sistema TileTransfer

Nesta seção será descrita a análise e o projeto em módulos do sistema TileTransfer. A figura a seguir mostra a arquitetura do sistema. Este proporciona acesso à base de dados Runinfo que contém as informações dos “runs” e aos arquivos de dados (ntuples) adquiridos nos testes de feixe de partículas, armazenados no CASTOR. Os resultados são apresentados em uma página na Internet para que o usuário possa fazer a transferência do arquivo.

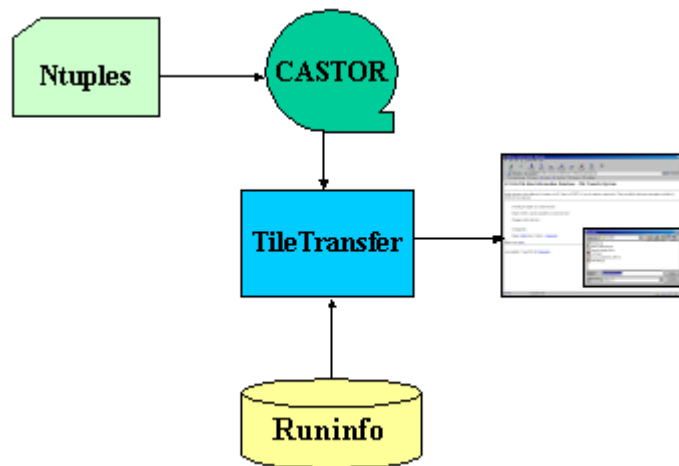


Figura 10 – Arquitetura do Sistema TileTransfer

3.1.1.1 O Módulo Início

O módulo início lê os atributos dos experimentos na base de dados runinfo, tabela Tilecal, que contém os parâmetros das condições dos testes realizados e os apresenta ao usuário. É através desses parâmetros que ele poderá fazer uma pesquisa por experimentos. É a página inicial do sistema TileTransfer. A figura 11 mostra a arquitetura do módulo início.

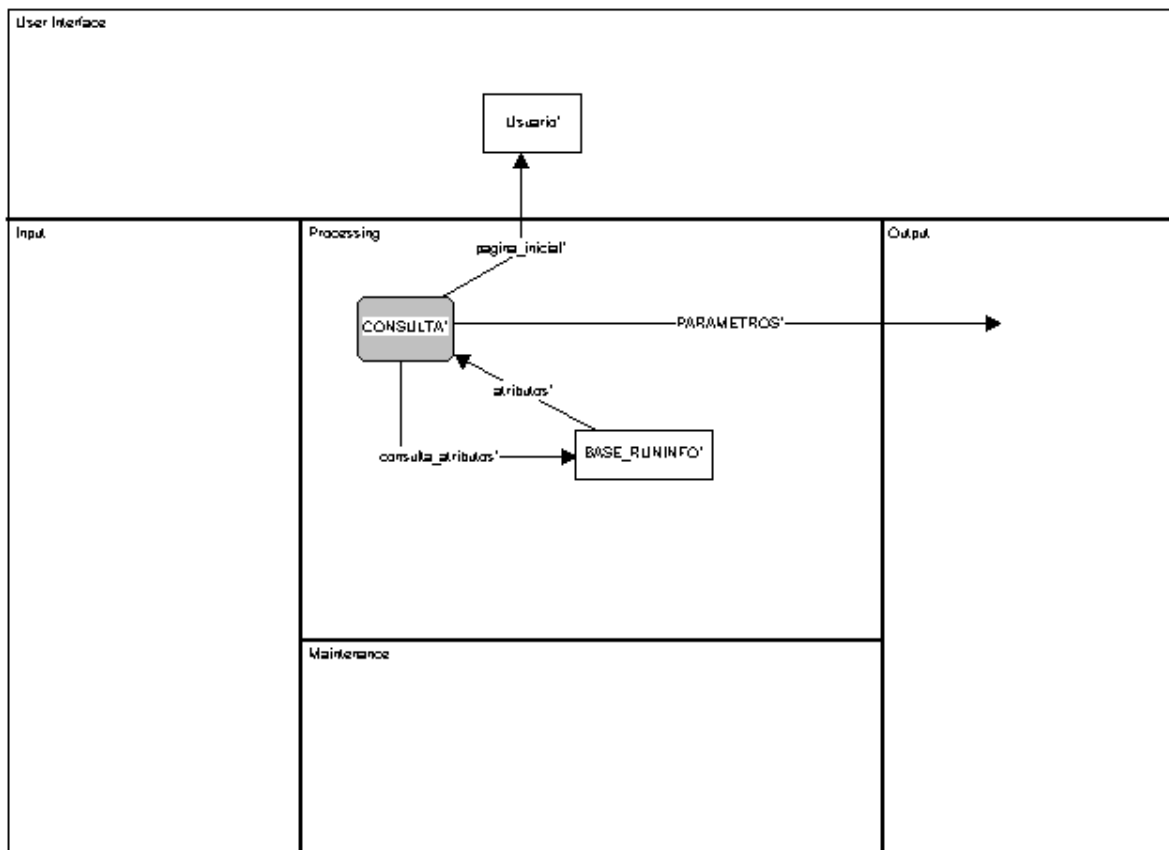


Figura 11 – Módulo Início

3.1.1.2 O Módulo Busca

O Módulo Busca recebe os parâmetros enviados pelo Módulo Início e os valores inseridos pelo usuário que procura por testes e faz uma consulta ao banco de dados na tabela Tilecal. As definições dos parâmetros da tabela Tilecal podem ser encontradas no Glossário (capítulo 9). O sistema recebe os resultados da consulta e monta uma tabela para que o usuário possa visualizar as ntuples retornadas. Um exemplo poderia ser o usuário fazendo busca por aquisições feitas no dia 02/11/2003 (os testes foram realizados em 2003).

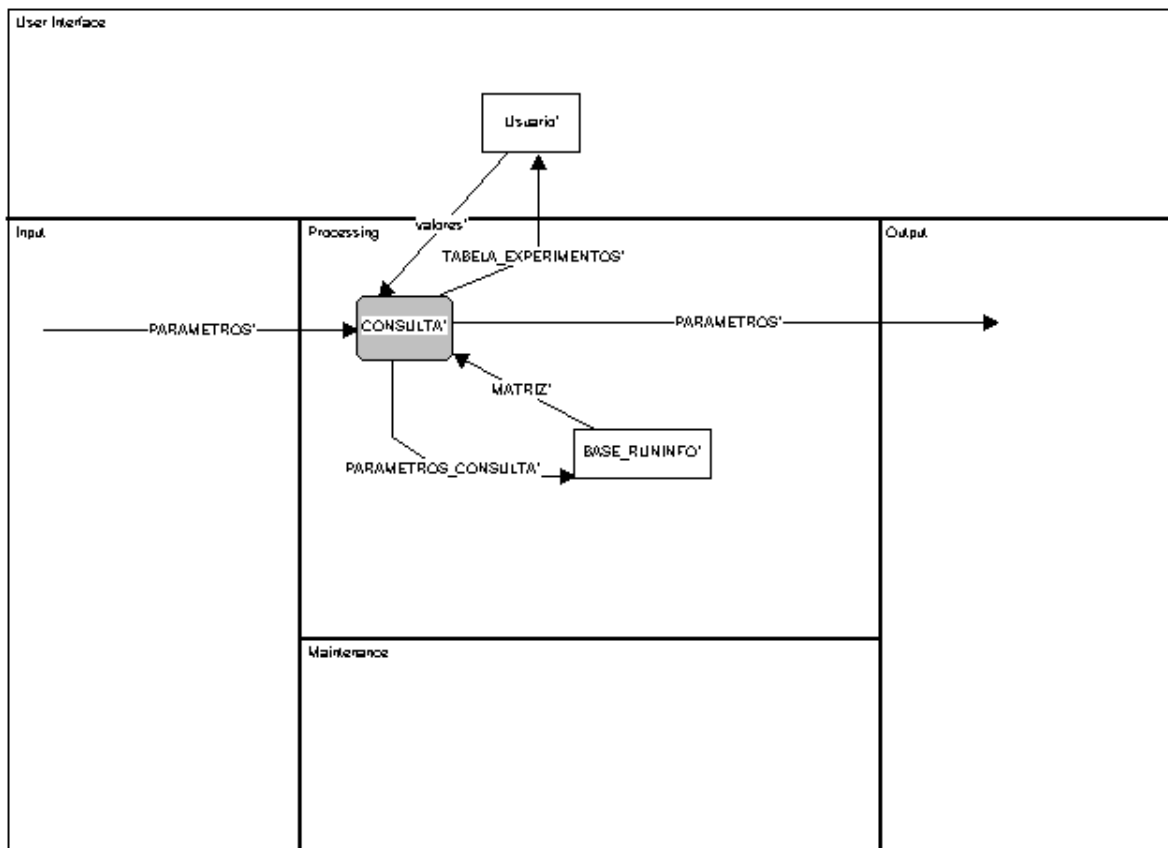


Figura 12 – Arquitetura do módulo Busca

3.1.1.3 O Módulo Download

O Módulo Download recebe a escolha do usuário que é feita através da tabela que o mesmo recebe do módulo Busca. O usuário faz três seleções:

- Arquivos (ntuples)
- Download now / later
- Arquivo comprimido ou não

Sendo assim, o usuário escolhe que arquivos de testes ele quer receber, se ele quer esperar o processamento no navegador, acompanhando as mensagens de processamento, ou receber o resultado por e-mail, e se ele quer o arquivo comprimido ou não. Antes de iniciar o processamento, os dados enviados pelo usuário são validados pelo Módulo de Validação. Se estiverem incorretos o usuário recebe uma mensagem.

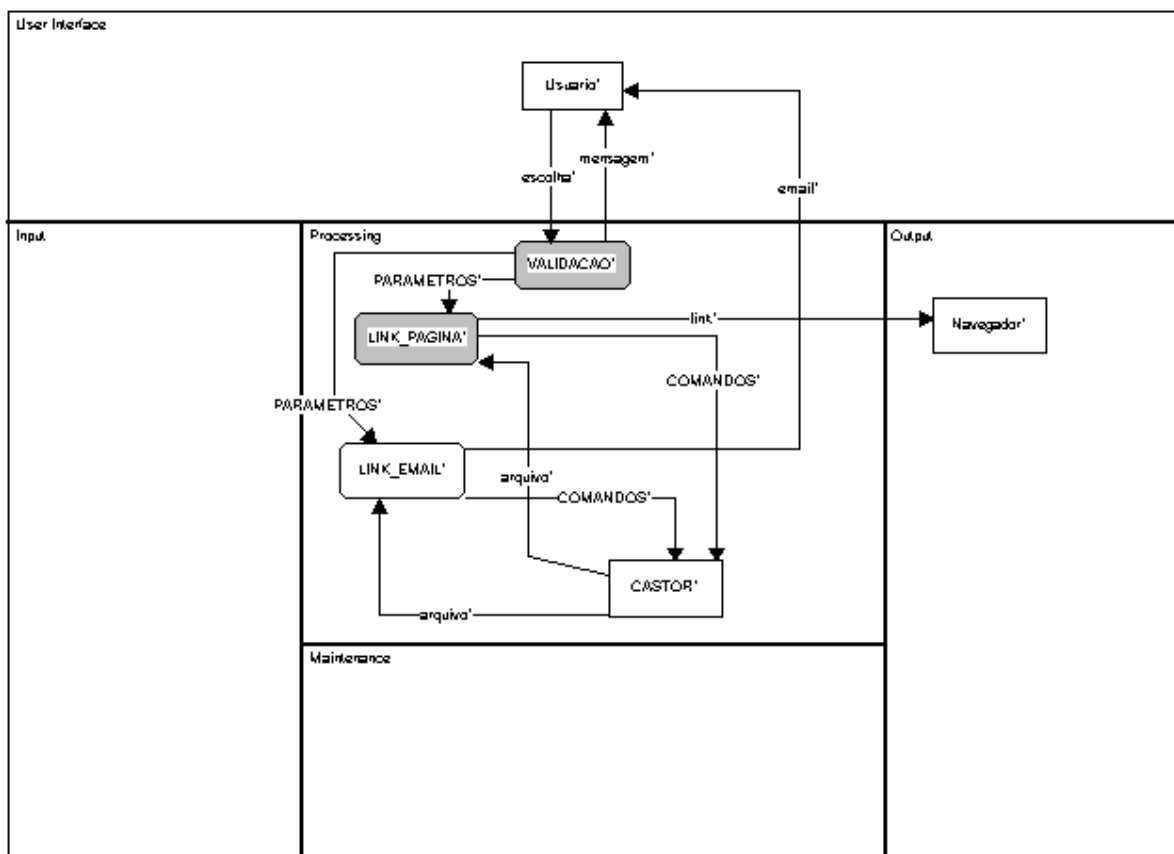


Figura 13 – Módulo Download

3.1.1.4 O Módulo Validação

O módulo de validação verifica se o usuário inseriu corretamente os parâmetros para que o sistema possa buscar e transferir os arquivos. Ele verifica se o usuário escolheu ou não um experimento (se não escolheu envia um alerta ao usuário). Verifica também, quando o usuário escolhe receber o link por e-mail, se o e-mail foi preenchido corretamente já que o e-mail tem que ter “@” e “.”. É importante ressaltar que o módulo validação foi feito em Javascript, linguagem apropriada para executar código na máquina do cliente, evitando processamento no servidor.

3.1.1.5 O Módulo Link_email

O Módulo Link_email é executado quando o usuário escolhe receber os resultados por e-mail. O sistema cria um processo filho e lhe passa os parâmetros de escolha do usuário. Ele irá executar os comandos no servidor de arquivos (CASTOR) e daí chamará o programa sendmail, que enviará ao usuário os resultados de sua requisição.

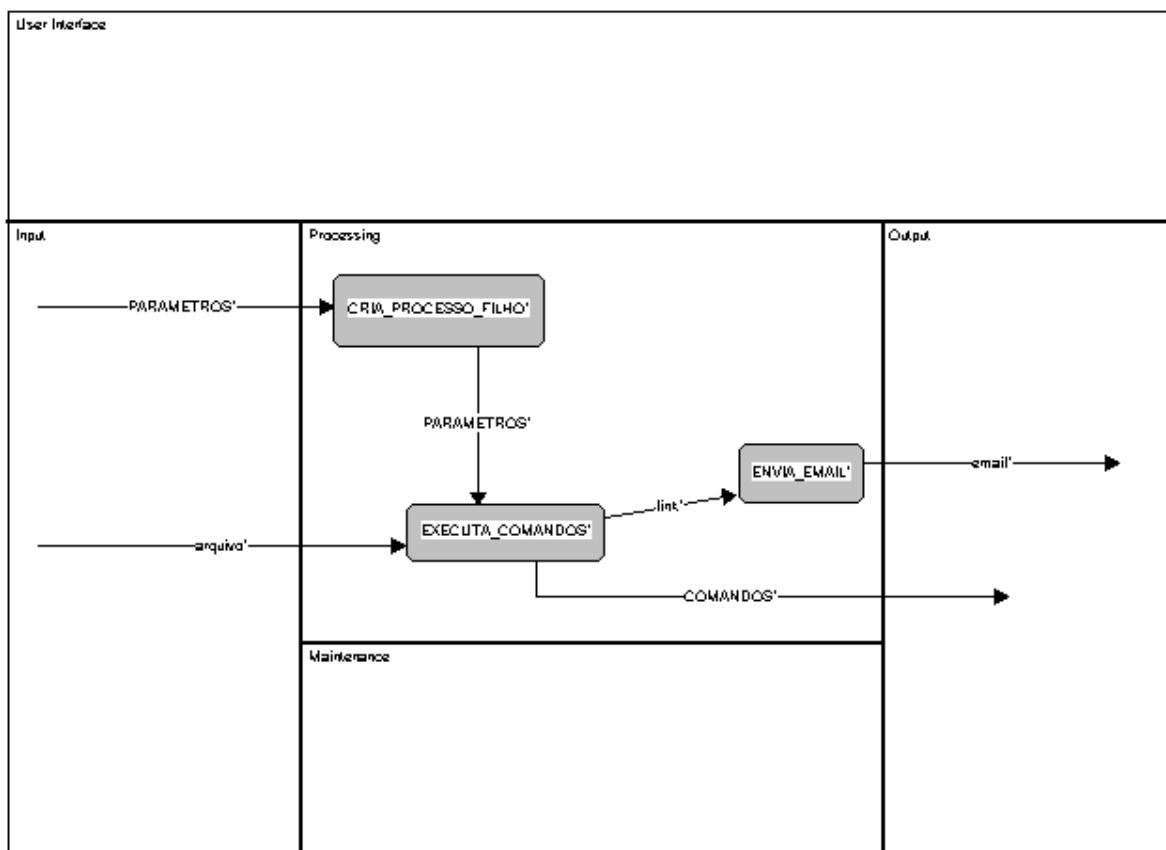


Figura 14 – Módulo: Link_email

3.1.1.6 O Módulo Link_página

O Módulo Link_página recebe os parâmetros escolhidos pelo usuário e chama o Módulo Executa_comandos, porém se diferencia do módulo link_email por retornar os resultados da solicitação do usuário na própria página do navegador.

3.1.1.7 O Módulo Executa_comandos

O módulo executa_comandos é o responsável pela conexão com o servidor de arquivos (CASTOR) e pela execução de programas e funções com o intuito de disponibilizar o arquivo desejado pelo usuário. Todos os comandos executados para disponibilizar o arquivo se encontram em uma página de manual do CASTOR. Primeiramente, ele procura se o arquivo existe no servidor executando o comando **nsIs**. Depois disso, executa o script **get_ntuple** para fazer o “stage out” e tornar possível a realização de uma cópia desse arquivo do servidor. Com o comando **rfcp**, é possível transferir o arquivo do servidor CASTOR para a máquina pcata007.

Abaixo está a lista de comandos a serem executados para se obter um arquivo. Essa lista foi retirada de uma url que contém um manual de instrução do CASTOR 10.

```
nsls /castor/cern.ch/user/t/tilebeam/online/jun02/r0200xxx.hbook  
~tilebeam/scripts/get_ntuple 200xxx ONL jun02  
rfcp /castor/cern.ch/user/t/tilebeam/online/jun02/r0200xxx.hbook .
```

Os protótipos das funções estão listados abaixo:

void setTilebeamEnvironment(void);

Seta as variáveis de ambiente.

int tiletransfer_searchNtuple(char *, char *, char *);

Procura por arquivos que já estão armazenados na conta TileTransfer. Acontece quando um outro usuário já tenha requisitado anteriormente o mesmo experimento. Isso é feito para poupar tempo uma vez que se o arquivo já estiver na conta TileTransfer não será necessário conectar ao CASTOR para obtê-lo.

int tiletransfer_deleteNtuples(char *);

Deleta arquivos antigos na conta TileTransfer. É necessário pois o espaço em disco é limitado.

int tiletransfer_nsls(char *, char *, char *);

Executa o comando nsls no servidor CASTOR.

int tiletransfer_getNtuple (char *, char *, char *);

Executa o comando get_ntuple no servidor CASTOR.

int tiletransfer_rfcpNtuple(char *, char *, char *);

Executa o comando rfcp no servidor CASTOR.

int tiletransfer_compressNtuple(char *, char *);

Comprime ou não o arquivo de acordo com a opção do usuário.

int tiletransfer_createLinks(char *, int, char *, char *);

Cria os links que serão disponibilizados na página.

3.2 A Versão CombinedTransfer

Ao longo do ano de 2004, foram realizados no CERN testes com feixes de partículas para se avaliar o correto funcionamento dos subsistemas do ATLAS atuando em conjunto (Combined TestBeam). Estes testes geraram dados brutos que foram gravados em arquivos no CASTOR. Depois, programas são executados para reconstruir os eventos ocorridos durante os testes, gerando, a partir dos dados brutos, arquivos com formato conveniente para a análise dos dados por parte dos físicos participantes do experimento.

Além dos testes com dados reais, existem pacotes de software que permitem a simulação de eventos. Os resultados das simulações podem, então, ser comparados com os resultados obtidos com os dados reais. Assim, muitos de arquivos de dados são gerados tanto pela reconstrução quanto pela simulação e são armazenados no CASTOR. Estes arquivos precisam ser acessados pelos pesquisadores para que possam realizar as observações e análises pertinentes.

Dessa forma, surgiu a necessidade de estender o sistema TileTransfer para proporcionar a transferência destes novos testes do detector ATLAS do CERN. A proposta foi o sistema CombinedTransfer. É importante ressaltar que o sistema CombinedTransfer foi desenvolvido pelo aluno de graduação Kaio Karam e que, apesar de não ter desenvolvido o sistema CombinedTransfer, tive que estudar o que foi implementado, já que novas tecnologias, como o AMI e XML apresentados a seguir, foram inseridas e a próxima versão as manterá.

O CombinedTransfer se apresenta como um guia para a escolha dos arquivos, com uma interface na Web que permite a busca pelos dados de condições e apresenta estes dados junto com um mecanismo para a seleção dos arquivos. O sistema, naturalmente, realiza todos os passos necessários à transferência dos arquivos, de forma totalmente transparente ao usuário final, apresentando-os prontos para serem transferidos da Web via protocolo HTTP.

A figura 15 ilustra melhor o serviço oferecido pelo sistema CombinedTransfer mostrando os processos que geram dados para análise e a interface que o sistema realiza entre estes dados e os usuários. Cada módulo do CombinedTransfer trata dos dados referentes a um destes processos. Cada um também possui um banco de dados diferente. A interface AMI aparece na figura e é um sistema utilizado como bookkeeping oficial (armazenamento de dados sobre os experimentos realizados) de vários destes processos ocorridos no CERN, no contexto do detector ATLAS.

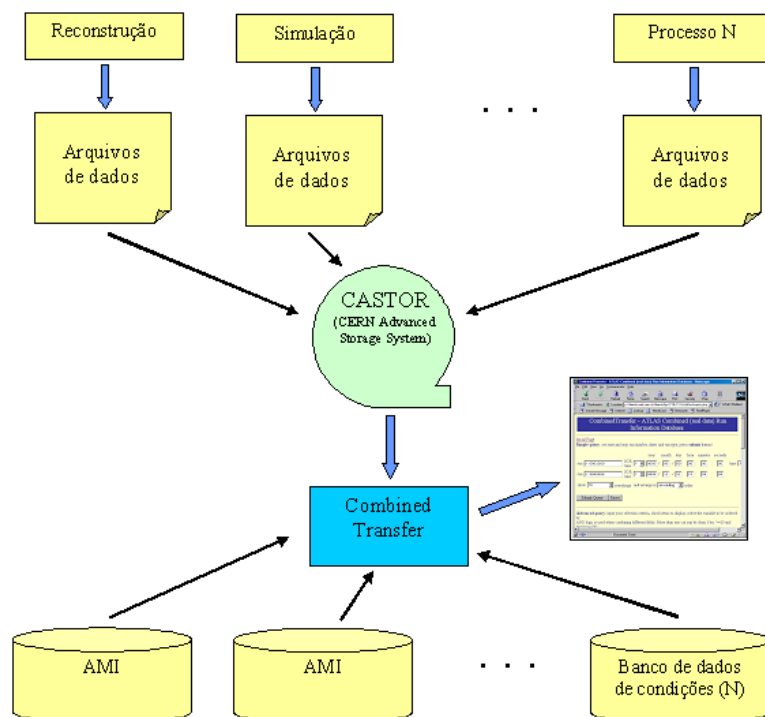


Figura 15 - Diagrama esquemático do sistema

O CombinedTransfer é um sistema com interface Web. O sistema está atualmente na versão 2.0 e pode ser acessado pelo endereço <http://tileinfo.web.cern.ch/tileinfo/lps/CTB/>. Na página inicial, o usuário pode selecionar o subsistema do ATLAS ou tipo de teste/experimento do qual deseja obter os dados. Em seguida, é mostrada uma página de busca onde o usuário insere os dados com as características dos experimentos físicos dos quais deseja analisar os arquivos resultantes. Ao submeter a busca, o sistema consulta o banco de dados correspondente ao módulo do sistema escolhido na tela inicial e disponibiliza no navegador os resultados da pesquisa. Estes resultados, juntamente com o mecanismo que permite a seleção de arquivos a serem transferidos. O sistema CombinedTransfer, da mesma forma que o TileTransfer, oferece duas formas para a obtenção dos arquivos. Uma delas é através do próprio navegador e a outra por e-mail.

3.2.1 Interface CombinedTransfer-AMI

O AMI (ATLAS Metadata Interface) 10 é um conjunto de interfaces (interface web, interface de linha de comando) projetadas para guardar e recuperar informações sobre experimentos do ATLAS. Na realidade, o AMI é bem genérico e pode ser adaptado para várias aplicações de base de dados. Seus autores são Solveig Albrand (solveig.albrand@lpsc.in2p3.fr) e Jérôme Fulachier (jerome.fulachier@lpsc.in2p3.fr).

Os dados do AMI são organizados em projetos 10. Um projeto consiste em um conjunto de dados que são produzidos para um determinado propósito, seja por critérios físicos ou computacionais. Um exemplo de projeto é o ATLAS Data Challenge, um conjunto de testes de

feixe de partículas realizado no detector ATLAS com o intuito de testar a nova infra-estrutura Grid. Os dados de cada projeto passam por etapas de processamento, seja para corte de pedaços desnecessários ou para transformação dos dados aos interesses dos físicos. Cada projeto é, então, dividido em etapas de processamento (“processing steps”), que apontam para uma base de dados. Cada etapa de processamento é uma transformação aplicada aos dados, a qual tem um nome, como “simulation” ou “reconstruction”. Os dados dos projetos são organizados em entidades. Cada entidade representa uma tabela da base de dados, que corresponde a um conjunto de testes de feixe de partícula efetuado no detector ATLAS. Sendo assim, cada teste possui uma entrada em uma determinada tabela. É através da interface com o AMI que o sistema CombinedTransfer realiza a busca determinada pela pesquisa do usuário, que seleciona os atributos dos experimentos desejados e o sistema monta uma consulta sql e acessa uma tabela relacionada a esses testes, retornando o resultado da consulta em uma tabela HTML.

A maioria das bases de dados do AMI podem ser acessadas e pesquisadas pela Web Interface. Ao selecionar o “Project”, a página é atualizada para permitir a seleção do “Process”. Selecionado o “Process”, a página é novamente atualizada para permitir a seleção de “Entity”. Ao inserir os valores e selecionar “Find” teremos o resultado da pesquisa mostrado em uma tabela. Ao final da tabela, aparece uma caixa de texto, com a SQL Query, e o link “AMI Command”. Selecionar esse link nos remete a uma nova página onde podemos alterar a Query e executá-la novamente.

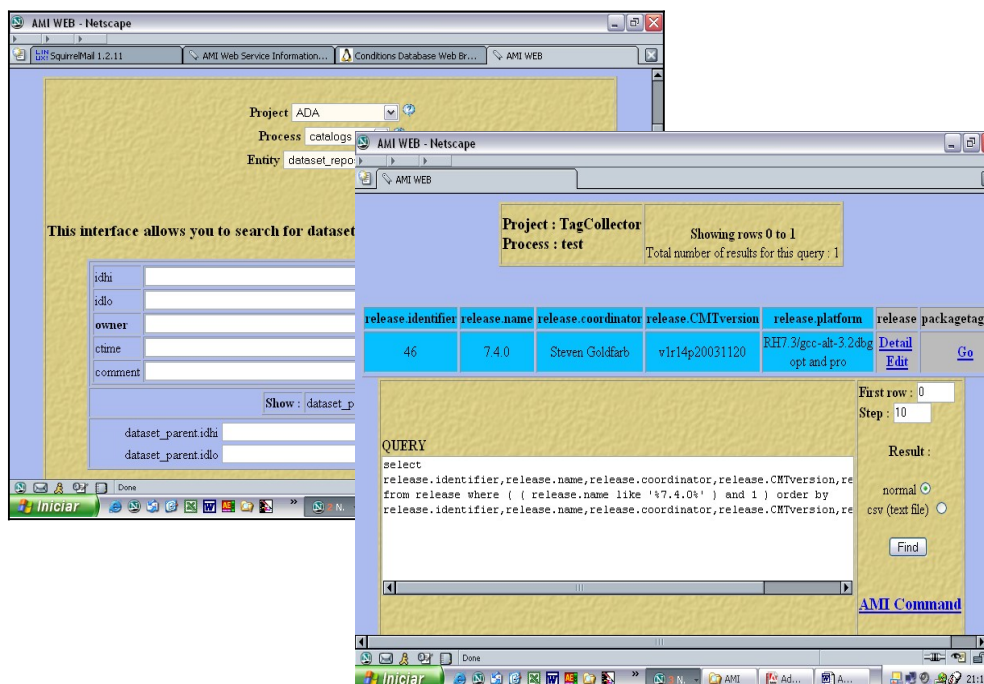
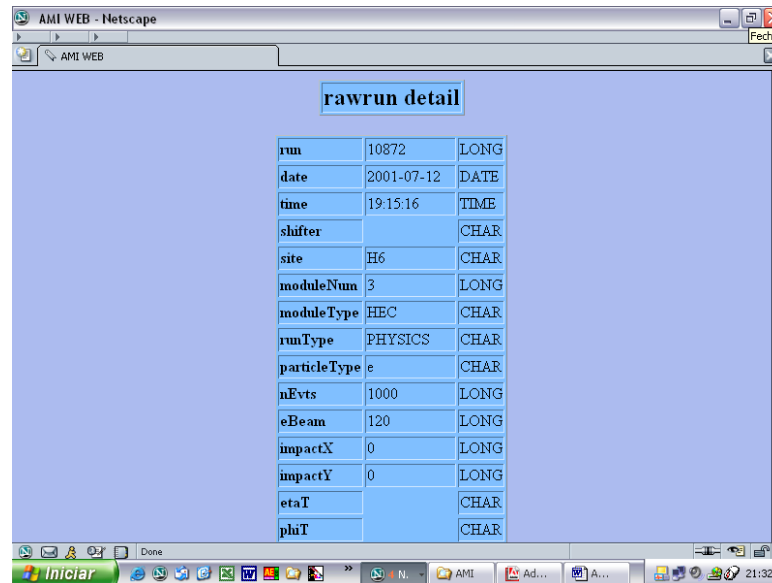


Figura 16 - Páginas de busca e resultados

Os resultados da busca podem ser exibidos nos formatos XML, HTML ou texto.

Podemos visualizar, na figura 17 uma página com uma tabela que mostra a estrutura da tabela relacionada à entidade escolhida.



The screenshot shows a Netscape browser window titled 'AMI WEB - Netscape'. The address bar shows 'AMI WEB'. The main content area displays a table titled 'rawrun detail'. The table has 15 rows, each representing an attribute and its value. The attributes are: run, date, time, shifter, site, moduleNum, moduleType, runType, particleType, nEvts, eBeam, impactX, impactY, etaT, and phiT. The values are: 10872, 2001-07-12, 19:15:16, H6, 3, HEC, PHYSICS, e, 1000, 120, 0, 0, etaT, and phiT. The table is displayed on a light blue background.

Attribute	Value	Attribute	Value
run	10872	shifter	CHAR
date	2001-07-12	site	H6
time	19:15:16	moduleNum	3
shifter	CHAR	moduleType	HEC
site	H6	runType	PHYSICS
moduleNum	3	particleType	e
moduleType	HEC	nEvts	1000
runType	PHYSICS	eBeam	120
particleType	e	impactX	0
nEvts	1000	impactY	0
eBeam	120	etaT	CHAR
impactX	0	phiT	CHAR
impactY	0		
etaT	CHAR		
phiT	CHAR		

Figura 17 - Tabela com os atributos da entidade

É com esses valores que o sistema CombinedTransfer monta a página de pesquisa para que o usuário possa inserir seus valores de busca.

Uma outra forma de acessar as bases de dados é através de um cliente Java executado pela linha de comando. Devem ser enviados alguns parâmetros como projeto, processing step e a sqlquery. O cliente acessa a base e retorna um arquivo em Xml com o resultado da consulta.

O código em linguagem C abaixo mostra o sistema CombinedTransfer montando o comando que irá ser executado na interface AMI:

```
snprintf(command, 10*SIZE, "export CLASSPATH=%s:$CLASSPATH; %s %s
-command=SearchQuery -project=%s -processingStep=%s -sql=\"%s\" > tmp%s", AMI_CLASS,
JAVA, AMI, argv[1], argv[2], argv[3], filename);
```

As constantes são definidas em um arquivo '.h' e os argumentos argv[] são o projeto, o processing step e a sqlquery respectivamente.

Após receber o resultado em XML no arquivo 'filename', o sistema executa um programa xsl para montar a tabela de resultados para o usuário.

4 Tecnologias Utilizadas

Esta seção descreve a tecnologia aplicada no desenvolvimento do sistema GridTransfer e versões anteriores. A escolha da tecnologia considerou as características da colaboração ATLAS, geograficamente dispersa e heterogênea. Portanto, o sistema foi desenvolvido para a Web com o objetivo de facilitar o acesso remoto e podendo ser utilizado em diferentes plataformas.

4.1 A Linguagem C

Usada inicialmente para a programação de sistemas, viria pela sua flexibilidade e poder, a tornar-se numa linguagem de uso geral nos mais diversos contextos, sendo atualmente muito utilizada pelos programadores. A linguagem C pode ser considerada uma linguagem de médio nível, porque para além de possuir instruções de alto nível e ser tão estruturada, também possui instruções de baixo nível. As suas principais características são: potencialidades e conveniência das linguagens quer de alto nível quer de baixo nível, pequeno conjunto de palavras reservadas, permite a criação de novos tipos de dados pelo utilizador, portabilidade, já que os programas escritos em linguagem C podem ser transportados entre máquinas de diferentes arquiteturas, com um número muito reduzido de alterações e eficiência, através de uma boa ligação aos recursos da máquina.

Existem algumas particularidades na linguagem C que a tornam especialmente interessante. Permite conversões de quase todos os tipos de dados. (Os tipos caracter e inteiro podem ser misturados na maioria das expressões). Não possui nenhuma verificação de erros em tempo de processamento (fim de vetor ou compatibilidade do tipo de argumento), cabe inteiramente ao programador saber o que o programa está fazendo.

A linguagem C é amplamente utilizada em todas as versões do sistema uma vez que as principais funções do sistema foram feitas utilizando esta linguagem. Por exemplo, o sistema utiliza a potencialidade da linguagem C ao criar um processo-filho no sistema operacional e executar os comandos necessários para disponibilizar os arquivos no CASTOR. Além disso, a linguagem proporcionou integração com outras ferramentas utilizadas no sistema que serão apresentados mais adiante.

4.2 A Linguagem PHP

PHP 10 é uma linguagem que permite criar sites WEB dinâmicos, possibilitando uma interação com o usuário através de formulários, parâmetros da URL e links. A diferença de PHP com relação a linguagens semelhantes a Javascript é que o código PHP é executado no servidor, sendo enviado para o cliente apenas o código HTML puro. Desta maneira é possível interagir com bancos de dados e aplicações existentes no servidor, com a vantagem de não expor o código fonte para o cliente. Isso pode ser útil quando o programa está lidando com senhas ou qualquer tipo de informação confidencial.

O que diferencia PHP de um script CGI escrito em C é que o código PHP fica embutido no próprio HTML, enquanto no outro caso é necessário que o script CGI gere todo o código HTML, ou leia de um outro arquivo.

PHP também tem como uma das características mais importantes o suporte a um grande número de bancos de dados, como dBase, Interbase, mSQL, MySQL, Oracle, Sybase, PostgreSQL e vários outros. Construir uma página baseada em um banco de dados torna-se uma tarefa extremamente simples com PHP. Além disso, PHP tem suporte a outros serviços através de protocolos como IMAP, SNMP, NNTP, POP3 e, logicamente, HTTP. Ainda é possível abrir *sockets* e interagir com outros protocolos.

O PHP é utilizado no projeto como interface com o usuário. É também com essa linguagem que é feita a comunicação com o banco de dados Mysql.

4.3 A Linguagem Javascript

JavaScript é uma linguagem orientada a objetos voltada para a criação de trechos de código (*scripts*) embutidos dentro de páginas HTML, interpretados e executados pelo programa navegador, na máquina do usuário. Esses *scripts* proporcionam uma maior dinâmica às páginas HTML, visto que possibilitam uma interação maior com o usuário.

As páginas do sistema utilizam JavaScript para validar os formulários HTML de entrada de dados dos usuários no cliente, impedindo que os mesmos sejam submetidos de forma incorreta para o servidor e, portanto, evitando o tráfego desnecessário de dados pela rede.

4.4 O Banco de Dados MySQL

O MySQL é um RDBMS (Relational Data Base Management System) de código fonte aberto (Open Source), o que significa que qualquer usuário pode estudar seu código fonte e modificá-lo de acordo com suas necessidades. É distribuído de acordo com uma licença GPL (GNU General Public License), a qual define suas normas de utilização, ou seja, o que pode e o que não pode ser feito com o software em diferentes situações. Para fins não comerciais, o MySQL é totalmente livre.

O MySQL é um banco de dados muito rápido, confiável e de fácil utilização. Para a manipulação e definição de dados, o MySQL permite o uso da SQL (Structured Query Language), a linguagem mais utilizada para acesso a bancos de dados relacionais.

Há versões do MySQL para diversas plataformas de hardware e software. O MySQL é distribuído em formato binário para os sistemas operacionais mais difundidos (Microsoft Windows 95, 98 e NT, Linux e Unix), podendo também ser obtido na forma de código fonte para compilação.

4.5 A linguagem XML

O XML (eXtensible Markup Language) é uma linguagem de marcação de texto simples, muito parecida com o HTML. Foi criado para servir à descrição de dados. A diferença em relação ao HTML é que, enquanto este dá ênfase à apresentação dos dados, o XML permite descrever os dados concentrando-se na estrutura destes. Esta linguagem é utilizada na troca de dados entre sistemas computacionais, já que permite a formatação dos dados num padrão que pode ser entendido por qualquer sistema, e na publicação de dados, com o auxílio de linguagens e programas que são capazes de apresentar dados armazenados em XML de maneira amigável ao usuário final.

A linguagem utilizada para a apresentação dos dados é o XSL 10. É uma linguagem baseada em XML que define transformações sobre os documentos XML que resultam em novos documentos XML ou em código HTML, que é o código utilizado para a apresentação em navegadores Web. Todos os dados apresentados ou recuperados pelo sistema no formato XML são submetidos a uma transformação XSL antes de serem enviados ao navegador do usuário.

5 Transferência de Arquivos

O volume de dados gerados nos testes é muito grande e é armazenado em arquivos, que devem ser transferidos para que os pesquisadores possam analisá-los. Sendo assim, notou-se a necessidade de encontrar meios mais eficientes para a transferência dos arquivos. A necessidade de otimizar a transferência refere-se à rapidez, e recuperação da transmissão no caso de interrupção. Características como segurança na autenticação, recuperação de arquivos, se houver falha na conexão (reiniciar a transferência de onde parou), e melhores taxas de transferência com melhor utilização da banda serão investigadas nessa pesquisa, já que os atuais protocolos utilizados (HTTP e ftp) não oferecem tais facilidades.

A seguir, são apresentadas algumas características dos protocolos e softwares encontrados.

5.1 BBFTP

BBFTP 10 é um aplicativo livre utilizado para transferir arquivos. Ele é roda em Unix e possui protocolo próprio (BBFTP) que encripta somente login e a senha não encriptando os dados que estão sendo transferidos. Suporta tráfego paralelo possibilitando a abertura de vários “sockets” de TCP para transferência de dados, permitindo que consiga atingir larguras de banda maior do que o ftp, e portanto, maior velocidade. O BBFTP também executa uma nova tentativa automática caso haja perda de conexão e a transferência se reinicia de onde parou.

A conexão no BBFTP é feita da forma tradicional cliente-servidor. O servidor deve executar o programa BBFTPD e o cliente deve executar o BBFTP.

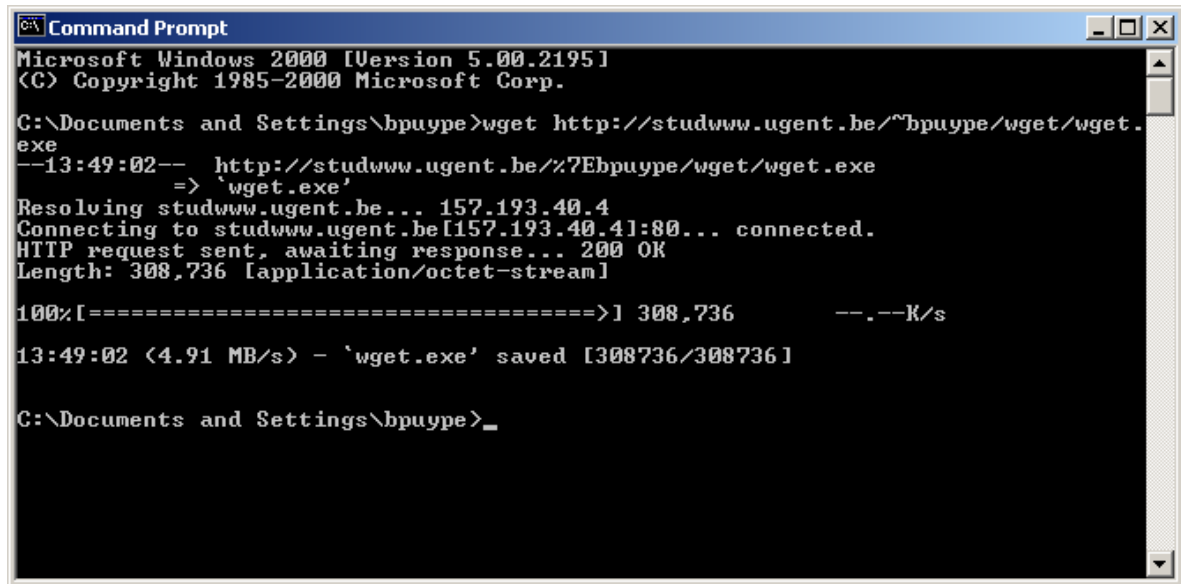
BBFTP permite dois métodos de conexão ao host. Uma conexão direta na porta de controle do daemon remoto do BBFTPD ou começar remotamente um daemon do BBFTPD através de um túnel do ssh. Para o primeiro método, o daemon do BBFTPD tem que ser inicializado (pelo inetd ou como um daemon autônomo) no host e toda informação do usuário (username, senha) é transmitida encriptada ao daemon; para o segundo, o BBFTPD tem que ser acessível em algum lugar no host e todos os dados de controle são transmitidos através do túnel do ssh.

5.2 WGET

GNU Wget 10 é um software livre para transferência de arquivos e funciona tanto em Unix como Windows. Ele utiliza os protocolos HTTP e ftp, os mais utilizados para transferência de arquivos. Uma das principais vantagens é que ele recupera os arquivos se a conexão falhar. É uma ferramenta não interativa (pode rodar em background) permitindo que o usuário comece uma recuperação e desconecte do sistema, deixando Wget terminar o trabalho. Roda em linha de comando, assim pode ser facilmente executado através de scripts, rotinas e terminais. O software é simples de usar e um exemplo é apresentado a seguir:

```
wget [option]... [URL]...
```


A figura 18 apresenta o uso do Wget para transferir o arquivo “wget.exe” do site studwww.ugent.be.



```
Microsoft Windows [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\Documents and Settings\bpuype>wget http://studwww.ugent.be/~bpuype/wget/wget.exe
--13:49:02-- http://studwww.ugent.be/~?Ebpuype/wget/wget.exe
=> 'wget.exe'
Resolving studwww.ugent.be... 157.193.40.4
Connecting to studwww.ugent.be[157.193.40.4]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 308,736 [application/octet-stream]

100%[=====>] 308,736      --.-K/s

13:49:02 (4.91 MB/s) - 'wget.exe' saved [308736/308736]

C:\Documents and Settings\bpuype>
```

Figura 18 -Exemplo de uso do Wget

5.3 Getright

O software GetRight funciona somente no sistema operacional Windows utilizando o protocolo HTTP e é “shareware”. Também é capaz de recuperar um arquivo se houver falha na conexão. É basicamente um programa gerenciador de transferência, oferecendo uma interface amigável onde o usuário pode mover o link do arquivo para a janela do GetRight para iniciar o processo de transferência.

5.4 Download Accelerator

O software “shareware” Download Accelerator funciona em Windows e também utiliza o protocolo HTTP. Assim como o Getright, recupera o arquivo caso haja falha na conexão. E ainda possui uma vantagem pois, assim como o BBFTP, possibilita maior velocidade ao paralelizar a transferência quebrando o arquivo em pedaços e enviando simultaneamente.

6 Grids Computacionais

As Grids computacionais são sistemas de suporte à execução de aplicações paralelas que acoplam recursos heterogêneos e geograficamente dispersos, possibilitando que sejam utilizados em grupo para executar aplicações de larga escala [10].

As Grids são ambientes de colaboração, onde os usuários se beneficiam dos recursos desta grande rede, utilizando a capacidade de processamento, compartilhamento de bases de dados, arquivos e informações, sendo que, o usuário também colabora, compartilhando os recursos de seu computador para benefício da rede.

As tecnologias Grid estão sendo utilizadas pela comunidade de cientistas do experimento do LHC uma vez que os experimentos irão gerar uma enorme quantidade de dados e irão requerer um ambiente computacional muito poderoso para o processamento desses dados. Dessa forma, o experimento LHC pode contar com os recursos computacionais da comunidade científica dispersos pelo mundo. De imediato podemos verificar uma vantagem econômica, uma vez que não seriam necessários gastos maiores com a compra de equipamentos para o processamento dos dados dos experimentos científicos.

As Grids estão sendo bem vistas também nos ambientes corporativos já que irá permitir redução de custos, tempo, aumento de produtividade, compartilhamento de recursos e informações, dentre outras possibilidades.

As Grids se tornaram possíveis nos últimos anos devido à grande melhora no desempenho e redução do custo das redes computacionais e dos microprocessadores. O software também desempenha papel importante em uma Grid, visto que é o responsável pelo gerenciamento dos recursos, compartilhamento de dados, autenticação, autorização, descoberta e acesso a recursos. Os métodos aplicados nesse contexto devem respeitar os quesitos de rapidez, confiabilidade, segurança e portabilidade.

A rapidez é importante uma vez que uma enorme quantidade de dados transita pela rede. Um protocolo de transmissão de dados mais eficiente é necessário. Confiabilidade e segurança são fundamentais, pois em um ambiente de compartilhamento de recursos um intruso poderia causar grandes problemas. Portabilidade também é importante porque os usuários estão dispersos pelo mundo e apesar de haver uma busca por padronização sempre existem usuários que utilizam ferramentas diferentes.

Conclui-se, portanto, que as Grids computacionais apresentam as seguintes vantagens: integração de recursos computacionais geograficamente distribuídos, maximização da utilização de recursos computacionais e integração de grupos de pesquisadores.

6.1 Projeto Globus

O Globus Toolkit produto desenvolvido pela Globus [10] é um software que implementa a maioria dos componentes e ferramentas necessárias para o funcionamento de uma Grid computacional.

A ferramenta GSI para autenticação e segurança, o protocolo GRIDFTP, mais eficiente na transmissão de dados, a aplicação MDS para monitoração e descoberta de recursos e a aplicação GRAM para a gerência de recursos são exemplos de ferramentas Globus desenvolvidas para Grids.

6.1.1 O Protocolo de Transferência GridFTP

O protocolo GridFTP 10 foi implementado para possibilitar a transferência de grande quantidade de dados em Grids, sendo inclusive usado e recomendado por projetos como DataGrid e Globus. O projeto Globus utiliza esse protocolo em sua principal ferramenta, o Globus Toolkit. Além disso, está havendo uma busca pela padronização da transferência de dados nas Grids e o protocolo a ser utilizado será o GridFTP.

Atualmente, os grandes sistemas de armazenamento utilizam protocolos diferentes, o que dificulta a eficiência na transferência de dados entre esses sistemas (dificulta a formação da Grid), além de dificultar a ação do usuário que tem que utilizar diferentes protocolos e bibliotecas para fazer a transferência. A proposta de projetos como o DataGrid e Globus é padronizar a transferência utilizando o protocolo GridFTP. Ele fornece além de segurança, eficiência na transferência de dados nos ambientes Grids. Ele é uma extensão do protocolo ftp com diversas vantagens, tais como:

- Oferece infra-estrutura de segurança Grid (GSI), significando que a autenticação é encriptada.
- Permite transferência paralelizada (múltiplos “sockets” TCP), implicando maior utilização da banda e aumentando a taxa de transferência.
- Permite transferência de dados que estão divididos entre diversos servidores (striped data transfer).
- Permite transferência parcial de um arquivo. Então, se a transferência for interrompida por qualquer motivo, pode-se reiniciar de onde parou.
- Adapta o buffer para transferência automaticamente de acordo com o tamanho do arquivo (buffer grande para arquivos grandes e buffer pequeno para grande quantidade de arquivos pequenos).

6.1.2 Segurança GSI

Em uma Grid, recursos computacionais são compartilhados com a comunidade científica. Para assegurar que esses recursos sejam utilizados de maneira correta e por usuários que tenham permissão para uso, a Globus criou a infra-estrutura GSI (Grid Security Infrastructure) 10. A segurança GSI promove a autenticação de usuários, bem como a confidencialidade na comunicação através do uso de criptografia assimétrica e certificação digital.

6.1.2.1 Criptografia Assimétrica

Em uma comunicação que utiliza o algoritmo de criptografia assimétrica, uma mensagem criptografada com a chave pública só poderá ser descriptografada pela sua chave privada correspondente. Da mesma forma, uma mensagem criptografada com a chave privada pode ser somente descriptografada pela sua chave pública correspondente.

Os algoritmos de criptografia assimétrica podem ser utilizados para garantir autenticidade e confidencialidade à comunicação. Para confidencialidade, a chave pública é usada para criptografar mensagens, com isso só o dono da chave privada pode descriptografá-la. Para autenticidade, a chave privada é usada para criptografar a mensagem, garantindo que apenas o dono da chave privada poderia ter criptografado a mensagem que foi descriptografada pela chave pública.

6.1.2.2 Certificados

A objeto central na autenticação GSI é o certificado. Todos os usuários e serviços de uma Grid são identificados por um certificado que contém informações para identificação e autenticação do usuário ou serviço. Um certificado GSI contém quatro campos:

- O nome do usuário, que identifica o dono do certificado.
- A chave pública pertencente ao usuário.
- A identidade da autoridade de certificação (CA) que assinou o certificado do usuário.
- A assinatura digital da CA.

A CA (Certification Authority) é uma instituição que emite certificados digitais para seus usuários. Os mesmos são codificados no formato X 509, um formato padrão para certificados estabelecido pela Internet Engineering Task Force (IETF).

6.1.2.3 Comunicação entre Usuários Certificados

Se dois usuários possuem certificados e confiam na autoridade emissora do certificado do outro usuário, então ambos podem provar para o outro sua identidade. Isso é conhecido como autenticação mútua. O GSI usa SSL (Secure Sockets Layer) para esse tipo de autenticação.

Em uma comunicação entre 'A' e 'B', o procedimento de autenticação é descrito a seguir. Primeiramente A envia seu certificado para B. Esse certificado contém a identidade de 'A', a chave pública de 'A' e a assinatura digital da CA emissora do certificado. 'B' irá verificar se a assinatura digital da CA é válida. Sendo válida, 'B' envia um pacote contendo uma mensagem qualquer para que 'A' a criptografe com sua chave privada. Recebendo a mensagem criptografada por 'A', 'B' utiliza a chave pública de A para descriptografá-la. Se a mensagem for idêntica a original, 'B' estará certo de que está se comunicando com 'A'. 'A' fará o mesmo procedimento com 'B'. Após a autenticação mútua os usuários estarão habilitados para a comunicação.

É importante mencionar que a chave privada deve ser armazenada em um diretório onde somente o proprietário da chave possa lê-la. A segurança GSI também criptografa o arquivo através de uma senha. Sempre que o usuário for usar o certificado e, por consequência, a chave privada, ele terá que digitar sua senha. Para evitar que o usuário tenha que digitar muitas vezes a senha, uma vez que trabalhando numa Grid pode acessar muitos serviços, a Globus criou um programa que atua como um proxy. Sempre que requisitado para enviar o certificado, o proxy atua na comunicação quando for necessário o uso da chave privada. Isso evita que o usuário tenha que digitar a senha várias vezes.

6.2 Projeto LCG

O objetivo do projeto LCG (LHC Computing Grid) 10 é construir a infra-estrutura computacional para simulação, processamento e análise dos dados resultantes dos experimentos no LHC. Isso inclui toda a gama de bibliotecas e ferramentas necessárias ao suporte e desenvolvimento de aplicativos computacionais que irão promover o armazenamento e processamento de dados 10 10.

O LCG deve atender os requisitos de desempenho, poder computacional, capacidade de armazenamento e recursos humanos para operação e suporte. Agrupar toda essa estrutura em um único local é muito difícil. Sendo assim, os serviços computacionais do projeto LCG serão implementados de forma geograficamente distribuída em uma Grid computacional. Isso significa que os serviços utilizarão recursos de armazenamento e computacionais distribuídos por diversos institutos da colaboração LHC, conectados por redes de alta velocidade. Softwares foram desenvolvidos com o intuito de minimizar a complexidade do ambiente para o usuário, parecendo que todos os recursos estão disponíveis em um único centro computacional.

A interface de acesso à Grid LCG é a User Interface (UI), uma máquina que funciona como um “gateway” para os serviços da Grid e onde os usuários possuem uma conta na qual seus certificados estão instalados. Através da UI, o usuário pode ser autenticado e autorizado a utilizar os recursos e as funcionalidades da Grid. Ela fornece uma interface através de linha de comando onde o usuário pode executar operações de submissão de jobs, listar recursos disponíveis, cancelamento de jobs, verificação de status e de resultados, além do gerenciamento dos dados.

6.2.1 Obtendo o Certificado

Para ter acesso à Grid LCG o usuário precisa obter um certificado emitido por uma autoridade certificadora. O primeiro passo, então, é fazer um pedido à CA que no nosso caso é o ATLAS. De acordo com as instruções 10, deve-se executar o comando `~Gridca/scripts/cert-req` na máquina Lxplus. O programa criará interativamente com o usuário um arquivo (`userreq.pem`) com seus dados e pedirá que o mesmo escolha uma senha para a chave privada que será gerada pelo programa. Essa chave ficará armazenada em um arquivo chamado `userkey.pem` e deve ser armazenada no diretório `~/globus` com acesso de somente leitura para o dono da conta. Depois disso, o usuário deve mandar um e-mail para a CA com o arquivo de requisição `userreq.pem`. Na

CA irão verificar a validade dos dados e emitirão um certificado através de outro arquivo, o `usercert.pem`, que deverá ser armazenado junto com a chave privada no diretório `.globus`. Depois de três dias recebe-se o arquivo `usercert.pem`, com o certificado e a chave pública.

O passo seguinte é registrar-se no LCG 10. A primeira tarefa é importar o certificado no navegador, para que o LCG tenha certeza de que é a própria pessoa que está se registrando. Como o certificado foi obtido em formato ‘pem’ e determinados navegadores só aceitam o formato `pkcs12`, devemos executar um comando para convertê-lo:

```
openssl pkcs12 -export -inkey userkey.pem -in usercert.pem -out my_cert.p12 -name "My certificate"
```

Após a importação do certificado no navegador, deve-se preencher um formulário no site com dados pessoais e escolher uma Virtual Organization (VO) que no nosso caso é o ATLAS. Isso significa que, ao acessar a Grid, teremos acesso aos recursos da comunidade ATLAS, mas não de toda Grid.

Após o pedido de registro recebe-se um e-mail da comunidade ATLAS informando que o mesmo foi efetuado com sucesso e que pode-se utilizar os recursos da comunidade em Grid.

6.2.2 ATLAS Data Challenge

O ATLAS Data Challenge 10 corresponde a um conjunto de experimentos realizados no detector ATLAS cujo intuito, além do propósito original de realizar os experimentos de colisão de partículas e de testar os compartimentos do detector, foi de testar a infra-estrutura e as ferramentas Grid desenvolvidas nos últimos anos e validar seu modelo computacional 10. Dessa forma, os experimentos DC (Data Challenge) foram armazenados em mais de vinte institutos espalhados dentre três Grids computacionais que se interconectam: LCG, US Grid3 e NorduGrid. Outro objetivo alcançado pelo Data Challenge foi gerar dados simulados para o “High Level Trigger”. O trigger é formado por três níveis de filtros com o intuito de selecionar os eventos mais importantes no detector, eliminando ruído e resultados fora do esperado.

6.3 Ferramentas Grid

Nesta seção são apresentadas algumas ferramentas Grid pesquisadas.

6.3.1 Dirac

O Dirac 10 (Distributed Infrastructure with Remote Agent Control) é um software, desenvolvido em Python, que agrega ferramentas de comunicação entre os usuários e a Grid para download e upload de jobs. O poder computacional necessário para processar todos os dados resultantes dos experimentos científicos é muito grande. Sendo assim, uma solução encontrada foi distribuir esse processamento entre os computadores dos usuários da comunidade científica.

Assim, quando um usuário se conecta a uma Grid, ele recebe a incumbência de processar determinados dados e enviá-los ao final do processo. O Dirac é um software amigável que gerencia todos esses passos.

6.3.2 Gilda

O portal Gilda (Grid Infn Laboratory Dissemination Activities) ¹⁰ é um laboratório virtual para demonstração e disseminação do uso de Grids. Ele foi desenvolvido, pelo Istituto Nazionale di Fisica Nucleare (INFN), para facilitar o aprendizado e a utilização de Grids pelos pesquisadores. Baseado na ferramenta Genius (Grid Enabled web eNvironment for site Independent User Job Submission) ¹⁰, que proporciona diversas funcionalidades Grid, o portal Gilda é também autoridade certificadora, organização virtual (VO) e possui um sistema de monitoramento de recursos.

Dentre as funcionalidades da ferramenta Genius, podemos citar o gerenciamento de arquivos de usuários (criar, editar, visualizar, renomear, remover), a verificação de dados de sua virtual organization (como por exemplo verificar os usuários cadastrados na sua VO), ferramentas para visualização de resultados. Porém, sua principal vantagem é a submissão de trabalhos (jobs) em uma Grid. O portal Gilda possui ferramentas de edição de jobs, sendo que o usuário deve utilizar a linguagem jdl para editá-lo. Além disso, o usuário pode verificar a fila de jobs que foram submetidos na Grid. Dessa forma, o usuário pode transferir uma réplica para um recurso computacional, processá-lo submetendo um job e obter o resultado para fazer sua análise física. Anteriormente o usuário tinha que fazer a transferência do arquivo, processá-lo na sua máquina para obter o resultado. Essa é uma das vantagens oferecidas pelas Grids computacionais.

Para utilizar os recursos o usuário deve executar um proxy de comunicação onde o usuário envia seu certificado para o servidor de autenticação da Grid. A pergunta que se faz é como o portal Gilda consegue executar comandos do usuário em uma Grid se a autenticação GSI não permite. A resposta é que o usuário armazena o seu certificado em um servidor ‘Myproxy’ com uma senha. Ao autenticar no portal Genius, o usuário informa a senha e o servidor. Sendo assim, o Genius importa o certificado do usuário ficando capaz de executar as ordens do usuário na Grid.

Isso é possível fazer usando a ferramenta Globus Toolkit através dos comandos myproxy-init e myproxy-get-delegation ¹⁰. A figura 19 mostra a execução do myproxy-init. O usuário envia seu certificado ao servidor ‘Myproxy’ e atribui uma senha para posterior importação de certificado.

```
$ myproxy-init -a -s myproxy.ncsa.uiuc.edu
Your identity: /C=US/O=National Computational Science Alliance/CN=Jim Basney
Enter GRID pass phrase for this identity:
Creating proxy ..... Done
Your proxy is valid until Fri Sep 13 13:52:56 2002
Enter MyProxy Pass Phrase:
Verifying password - Enter MyProxy Pass Phrase:
A proxy valid for 168 hours (7.0 days) for user jbasney now exists on myproxy.ncsa.uiuc.edu.
```

Figura 19 – Execução do comando myproxy-init (Extraído de 10)

A figura 20 mostra a execução do comando myproxy-get-delegation para importar o certificado

```
$ myproxy-get-delegation -s myproxy.ncsa.uiuc.edu
Enter MyProxy Pass Phrase:
A proxy has been received for user jbasney in /tmp/x509up_u500
```

Figura 20 – Execução do comando myproxy-get-delegation (Execução de 10)

6.3.3 Grenade

O projeto Grenade (Grid ENabled Desktop Environment) 10, desenvolvido por pesquisadores da Universidade de Manchester, tem por objetivo tornar o uso da Grid mais amigável ao usuário através de aplicativos instalados no ambiente de trabalho. Na primeira etapa do projeto foram desenvolvidas ferramentas de gerenciamento de jobs e de arquivos em Grids através do KDE (K Desktop Environment), utilizado em sistemas operacionais como Linux, Solaris, Irix, Aix, entre outros. Uma outra funcionalidade desenvolvida pelo Grenade foi um plug-in para o navegador Konqueror onde o usuário pode colocar o endereço do arquivo da Grid (gsiftp://xxx) no navegador para fazer a transferência. Dessa forma, o usuário não precisa executar via linha de comando o aplicativo globus-url-copy, tornando mais amigável a transferência do arquivo.

6.3.4 Don Quijote

O aplicativo Don Quijote 10, desenvolvido por Miguel Branco da comunidade ATLAS, é utilizado para o gerenciamento de réplicas de arquivos em uma Grid. Essa ferramenta será usada, após os testes do Data Challenge 2, já que mais de um instituto passou a armazenar dados científicos. Dessa forma, o usuário do sistema buscará por arquivos de testes no Don Quijote e este retornará o instituto mais próximo para que ele possa transferir o arquivo para a sua central de armazenamento de seu instituto. Depois da transferência, o arquivo é copiado para o diretório local do usuário. Se o arquivo já estiver na central de armazenamento do instituto, o Don Quijote cria um link ou copia para o diretório local do usuário.

Para usuários que não possuem uma central de armazenamento em seu local de trabalho, o Don Quijote transfere o arquivo para o diretório local usando o protocolo GridFTP. Para isso, o usuário precisa ter certificado digital emitido pelo CERN e a infra-estrutura Grid instalada (Globus).

6.3.5 Edg (European DataGrid)

A ferramenta de gerenciamento de réplicas “Edg” 10, do projeto DataGrid 10, procura arquivos, lista unidades de armazenamento, verifica se a unidade está respondendo (ping), faz o “download” e “upload” de arquivos na Grid, ordena a réplica de um determinado arquivo para outro instituto, entre outros. Possui as mesmas funcionalidades que o Don Quijote.

7 O Sistema GridTransfer

A proposta do novo sistema GridTransfer é permitir que os usuários possam ter acesso aos arquivos de dados que se encontram armazenados em Grid, uma vez que após a validação do modelo computacional proposto pelo LCG, os arquivos passarão a ser armazenados de forma distribuída entre os diversos institutos de pesquisa da colaboração ATLAS. Para validar o sistema, foram utilizados os testes do Data Challenge, armazenados em Grid. Além disso, o sistema oferece acesso aos testes realizados no calorímetro de telhas, calorímetro de argônio líquido, testes de feixes combinados (Combined TestBeam), já implementados nas versões anteriores do sistema. Sendo assim, o sistema utiliza as tecnologias estudadas e implementadas nas versões anteriores (Capítulo 4). O usuário pode pesquisar por experimentos, através do processo de bookkeeping, já que o sistema serve de interface entre o usuário e a base de dados AMI. Após submeter a busca o sistema retorna uma tabela para que o usuário possa escolher os arquivos que deseja. Nesse momento, decide pelo método de transferência (HTTP, Wget, GridFTP ou Don Quijote), se quer o arquivo comprimido ou não e se quer receber os resultados na página ou por e-mail. A escolha de comprimir ou não o arquivo se restringe ao acesso HTTP e Wget, uma vez que no GridFTP e Don Quijote, a transferência será feita diretamente dos institutos que armazenam os arquivos. Sendo assim, o sistema, que não tem posse dos arquivos, não poderá comprimi-los. Após tal decisão, o sistema localizará, usando Don Quijote, o instituto mais próximo que esteja armazenando o arquivo e transferirá o arquivo para o diretório local do sistema de modo a oferecer acesso HTTP. Como os arquivos são grandes, o sistema deve apresentar um algoritmo de controle de espaço em disco, deletando os arquivos antigos à medida que o espaço em disco esteja chegando ao seu limite.

Já para transferência GridFTP, o sistema oferece um script Shell para executar via linha de comando a transferência. Esse script contém comandos para a verificação de espaço em disco, autenticação na Grid (Grid-proxy-init), além do globus-url-copy (cliente GridFTP) ou Don Quijote, que irá buscar e transferir os arquivos da Grid para a máquina do usuário. Os usuários que optarem pela transferência GridFTP devem ter os pré-requisitos citados anteriormente (certificado, software globus).

No caso de escolha do Wget, o sistema realizará o mesmo algoritmo do acesso HTTP, porém, ao final do processo, em vez de apresentar o link hipertextual, apresentará um script com o comando Wget e o link do arquivo como argumento.

7.1 Proposta de Sistema

Antes da existência das Grids, o pesquisador, para fazer sua análise dos dados adquiridos nos detectores, tinha que acessar um recurso de armazenamento, que no caso do ATLAS é o CASTOR, fazer a transferência dos dados resultantes dos testes, processá-los localmente em seu computador e, finalmente, realizar a sua análise através de ferramentas de análise física.

Com o advento das Grids, o pesquisador pode optar entre realizar o processamento dos arquivos localmente, conforme descrito no parágrafo anterior, ou processá-los através de algum

recurso computacional disponível na Grid. Em um ambiente Grid, o usuário pode, em vez de esperar a transferência dos arquivos e o processamento em seu computador, solicitar a transferência de uma réplica dos arquivos para algum recurso de processamento da rede, submeter um job de execução e, então, obter os resultados para fazer sua análise física. O resultado pode ser transferido para a máquina do usuário ou até mesmo para outro recurso de análise na Grid.

Para a realização desta tarefa o conhecimento de informática requerido é muito grande, devido a enorme gama de comandos específicos. Um sistema que oferecesse, de forma amigável e automatizada, as ferramentas para que os físicos pudessem fazer suas análises, seja através de processamento local, ou através da submissão de um job para processamento na Grid, é importante.

Um sistema deveria, inicialmente, apresentar duas opções ao usuário: processamento local ou processamento via Grid. No processamento local, o sistema deve oferecer ao usuário acesso aos dados dos experimentos para que possa fazer a transferência. Deve permitir que o usuário possa transferir o arquivo usando protocolos mais eficientes como o GridFTP, desde que possua os requisitos para efetuar tal transferência. Já para o processamento em Grid, o sistema deve reunir todas as ferramentas necessárias para que o usuário se autentique na Grid, envie seu script de job, monitore a execução do processamento dos arquivos e obtenha os arquivos resultantes. Além disso, ferramentas de análise física pela web também são necessárias caso o usuário opte por fazer a análise usando recursos da Grid.

A figura a seguir apresenta a cadeia de atividades que o pesquisador deve realizar para analisar os dados gerados no detector ATLAS e permite observar o que o sistema GridTransfer pode oferecer a um experimento do porte do ATLAS.

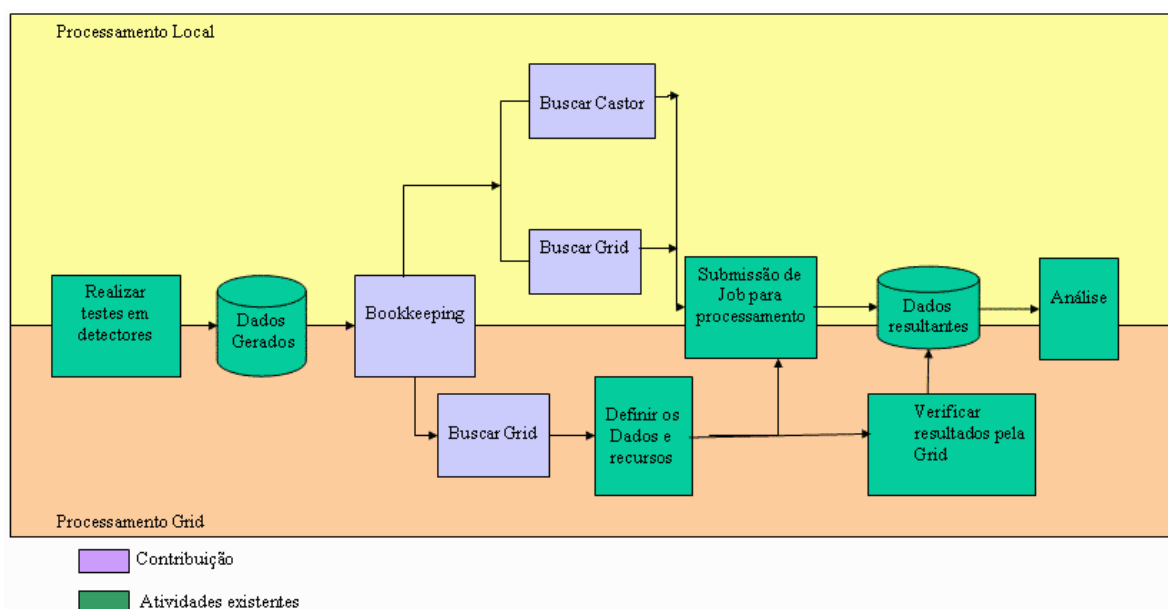


Figura 21 – Cadeia de atividades

Vejamos uma análise da cadeia de atividades de um pesquisador que deseja analisar um experimento do detector ATLAS, primeiramente, tem que coletar os dados gerados nos experimentos realizados nos detectores e processá-los para que tenham um significado físico. Dessa forma escolhe se fará o processamento local ou na Grid. Para a escolha dos dados do teste que ele deseja analisar, o físico consulta o bookkeeping de testes de experimentos, que é um banco de dados que contém informações sobre os arranjos experimentais realizados. Assim, durante os testes,, se algum pesquisador solicitou que fosse feito um experimento com determinadas características de seu interesse, ele deve consultar o bookkeeping para saber se seu experimento já foi efetuado. Pode-se também analisar os arranjos experimentais já realizados ao longo do programa de testes. Sendo assim, precisam consultar o bookkeeping para fazer a escolha. No caso de processamento local, ao escolher o arranjo experimental específico, o pesquisador deve buscar os dados no Castor ou na Grid. Após fazer a transferência dos dados, o pesquisador roda um programa que irá processar os dados na sua máquina, ou submete um job para o processamento no conjunto (cluster) de máquinas disponíveis localmente. Já para o processamento na Grid, basta que o pesquisador submeta um job especificando os dados que quer processar, para que algum recurso computacional disponível na Grid o faça. Cabe ressaltar que o usuário pode especificar quais recursos deseja utilizar e quais os requisitos serão necessários, como por exemplo, espaço em disco e a capacidade de processamento. Neste caso, o usuário pode acompanhar o status do job submetido. Após todas essas etapas, no caso do processamento local, o físico já possui o resultado na sua máquina para fazer a sua análise. No processamento em Grid, o pesquisador pode transferir o resultado para sua máquina utilizando um protocolo como o GridFTP, ou mesmo transferir para um recurso de armazenamento da Grid. Sendo assim, a análise pode ser feita utilizando programas em sua própria máquina ou até mesmo utilizando outros recursos e ferramentas de análise remota que a Grid oferece.

7.2 Análise de Requisitos dos Usuários

Esta seção mostra o estudo e a verificação do trabalho de dois físicos do Instituto de Física da UFRJ que utilizam a Grid para fazer suas análises físicas através de programas que simulam o funcionamento do detector Lhcb. Depois, no final da seção, é feita uma referência a um e-mail mandado por Solveig Albrand, da colaboração ATLAS, na interface AMI. Nesse e-mail ela mostra a necessidade de integrar as ferramentas Grid para proporcionar ao usuário uma interface amigável para a obtenção dos dados.

Inicialmente serão descritos os conhecimentos adquiridos durante a reunião com Bruno Souza de Paula, um aluno de doutorado do instituto de física da UFRJ, que esteve no CERN trabalhando no projeto Lhcb. Ele é um usuário da Grid e tem feito diversas testes em programas que simulam os detectores. A sua atividade é gerar os dados que caracterizam os experimentos, executar o programa que simula o detector e analisar o resultado obtido. Para executar suas simulações, ele submete um job na Grid. Pedimos então que ele demonstrasse como utiliza a Grid, os comandos e os jobs.

Primeiramente ele se loga na máquina Lxplus através do ssh. Depois ele executa um script que seta as variáveis de ambiente Grid através do comando `source /afs/CERN.ch/project/gd/LCG-share/2.4.0/sl3/etc/profile.d/Grid-env.csh`. O próximo passo é

autenticar na Grid criando um proxy (Grid-proxy-init). Normalmente ele configura a validade do proxy para três dias que é o tempo que seus jobs são executados na Grid. É importante ressaltar que, de acordo com Bruno, se o proxy terminar antes do fim do job, este será deletado.

Após a autenticação, ele submete os jobs para execução na Grid. Isso é feito através do comando `edg-job-submit [arquivo]`. Os jobs são descritos em linguagem jdl (job description language). Um exemplo pode ser visto, Anexo A (Capítulo 11), um job para submeter uma simulação, ou seja, o usuário faz uma requisição para utilizar um determinado programa que simula um detector, que no caso é o Lhcb, com os atributos físicos determinados pelo usuário no job.

No script, as bibliotecas e os dados são especificados como entrada (inputs), assim como os requisitos (variável requirements), como por exemplo memória e espaço em disco.

O status do job pode ser conferido através do comando `edg-job-list-match` e o resultado pode ser visto pelo comando `edg-job-get-output`.

Ao final do job, o próximo passo é analisar o arquivo resultante do processo e utilizar as ferramentas de análises físicas. Todas essas tarefas são executadas via linha de comando. Seria interessante oferecer uma interface mais amigável ao usuário poupando tempo e o trabalho de memorizar todos esses comandos.

Agora serão descritos os conhecimentos adquiridos na reunião com a professora Miriam, que também é pesquisadora do instituto de física da UFRJ e realiza análises físicas de simulações do projeto Lhcb. Primeiramente ela nos mostrou o software Dirac, que coordena a submissão de jobs na Grid. Este verifica os recursos disponíveis e encaminha os jobs para os mesmos. Dessa forma, ao se logar na Grid o Dirac roda um programa daemon que verifica se existe algum job para máquina do usuário executar. Se tiver, o Dirac transfere o job e os dados necessários para a execução do mesmo. Este é um exemplo em que o pesquisador colabora com a Grid compartilhando seus recursos computacionais para o processamento dos arquivos.

Outro software mostrado foi o LCG, usado para submissão de jobs também, porém mais eficiente, pois o usuário pode escolher o recurso e evitar que o job seja destinado a uma máquina mais lenta. Ao submeter um job para simulação e indicar os parâmetros e dados o LCG encontra um recurso computacional disponível com o programa que roda a simulação (chamado Root). Ele gera como resultado um arquivo `.root` que é o arquivo desejado para a análise física.

Existe uma página do Lhcb onde se faz o bookkeeping, ou seja, a escolha dos dados para a simulação ([HTTP://lhcbdata.home.CERN.ch/lhcbdata/bkk/](http://lhcbdata.home.CERN.ch/lhcbdata/bkk/)). O Dirac, já insere automaticamente na base de dados do bookkeeping os resultados dos jobs de simulação, para que os pesquisadores possam identificá-los posteriormente.

O código mostrado no Anexo B (Capítulo 12) é um outro exemplo de submissão de job.

Quanto ao e-mail da Solveig Albrand, presente no Anexo C (Capítulo 13), cita a necessidade de integrar as ferramentas Grid para proporcionar ao usuário uma interface amigável para a obtenção dos dados. Sugere a utilização da interface AMI e do Don Quijote e cita o que foi feito

no passado com o sistema TileTransfer dizendo que seria interessante estender as mesmas funcionalidades desenvolvidas anteriormente para os experimentos armazenados em Grid.

7.3 Especificação do Sistema

Nesta seção serão especificados as escolhas de métodos de transferência e ferramenta de gerenciamento de réplicas a serem utilizados na implementação do sistema. Além disso, serão definidos os requisitos para uso de cada uma delas.

7.3.1 Análise dos Protocolos de Transferência

Ao analisar as informações adquiridas no processo de pesquisa, comparando os softwares e protocolos apresentados anteriormente e respeitando as características do ambiente CERN, o protocolo GridFTP surge como melhor opção pelas razões apresentadas a seguir.

Os protocolos BBFTP e GridFTP possuem algumas características importantes para o sistema tais como maior utilização da banda de conexão permitindo taxas de transferência maiores, recuperação caso haja perda de conexão e também, a segurança na conexão (login e senha são transmitidos encriptados). O GridFTP vai mais além, permitindo transferência de dados que estão divididos entre diversos servidores (striped data transfer), além de outras vantagens para otimizar a transferência como adaptar o buffer para transferência automaticamente de acordo com o tamanho do arquivo.

Outra vantagem do GridFTP é a questão da padronização na questão da transferência de arquivos. Projetos como DataGrid e Globus e instituições que possuem grandes sistemas de armazenamento estão buscando melhor eficiência e conformidade usando um protocolo comum para transferência de dados e o GridFTP é utilizado pela maioria delas.

A tabela a seguir apresenta uma comparação dos softwares e protocolos estudados com relação ao seu funcionamento.

	GridFTP	BBFTP	Wget	Getright	Download Accelerator
sistema operacional	unix	unix	unix/win	win	win
protocolo	GridFTP	BBFTP	HTTP e ftp	HTTP	HTTP
encriptação na autenticação	sim	sim	não	não	não
conexões tcp paralelas	sim	sim	não	não	sim
recupera dados em falha	sim	sim	sim	sim	sim
striped data transfer	sim	não	não	não	não
adaptação do buffer	sim	não	não	não	não

Tabela 1: Comparação entre os programas de transferência de dados

7.3.2 Ferramenta de Gerenciamento de Réplicas

As duas ferramentas pesquisadas, EDG e Don Quijote, possuem as mesmas características e funcionalidades. Como o Don Quijote foi, recentemente, escolhido como a ferramenta oficial de “Data Management” (gerenciamento de dados) da comunidade ATLAS, esta ferramenta também será integrada ao sistema GridTransfer.

Os nomes das réplicas dos arquivos armazenados em Grid podem ser encontrados na base de dados AMI. Porém, existe uma diferença com relação ao nome dos arquivos. Antes, os arquivos eram obtidos através de seus nomes físicos de armazenamento, como por exemplo `/castor/cern.ch/grid/atlas/some_file.pool.root`. Só que ao implementar uma Grid, vários institutos passaram a armazenar os dados adquiridos. Como existem cópias de um mesmo arquivo espalhadas pela Grid, adotou-se uma convenção de atribuir a todas as replicas, um mesmo nome, o LFN (Logical File Name) que seria utilizado nas tabelas de dados em vez dos nomes físicos. É importante ressaltar que, em termos de nomenclatura, os nomes físicos passaram a ser chamados de SURL (Storage URL). O LFN é um nome amigável para um arquivo e suas réplicas. Um mesmo arquivo pode ser armazenado em diversos lugares, sob diferentes nomes físicos (nome real). O mesmo arquivo pode ser `/castor/CERN.ch/Grid/atlas/some_file.pool.root` no Castor@CERN e `/castor/ific.uv.es/Grid/atlas/123456.pool.root` no instituto IFIC na Espanha. Embora os arquivos tenham nomes físicos distintos, existe um catálogo que os mapeia em uma mesma LFN, assegurando que os arquivos são de fato cópias idênticas. A LFN poderia ser, por exemplo, `some_pool_root_file`. Sendo assim, todas as réplicas podem ter nomes físicos diferentes já que são mapeados pela LFN. Ao usar o Don Quijote, o usuário se abstrai dos nomes físicos passando a usar as LFNs. Saber a LFN é o suficiente para achar uma réplica de um arquivo numa Grid. Além da LFN, duas outras nomenclaturas são comuns, as LCN e LPN. A LCN (Logical Collection Name) é o diretório onde as LFN estão armazenadas, por exemplo `/datafiles/dc2/evgen/`. A LPN (Logical Path Name) é o nome completo de um arquivo em uma Grid. É a concatenação da LCN com a LFN. Os usuários devem usar as LPNs para localizar um arquivo via linha de comando. Porém, usando o Don Quijote é necessário apenas a LFN como parâmetro. Outra forma de representar o arquivo, porém menos amigável ao usuário, é através da GUID (Grid Unique identifier). O usuário pode identificar qualquer arquivo na Grid através de uma string de 40 bytes, como por exemplo `guid:38ed3f60-c402-11d7-a6b0-f53ee5a37e1d`. Ambas, GUID e LFN, podem ser usados para localizar arquivos na Grid, porém a LFN é mais amigável ao usuário. Para mapear as LFNs e GUIDs nos nomes físicos (SURL), existe um catálogo que possui uma base de dados com todos os GUIDs e suas LFNs associadas às replicas (nomes físicos). Esse catálogo é o RLS (Replica Location Server). O RLS é subdividido em dois catálogos, o Replica Metadata Catalog (RMC) e o Local Replica Catalog (LRC). O primeiro é responsável por mapear as LFNs em GUIDs e o segundo, GUIDs em SURLs. A figura a seguir mostra um esquema do mapeamento e os catálogos utilizados.

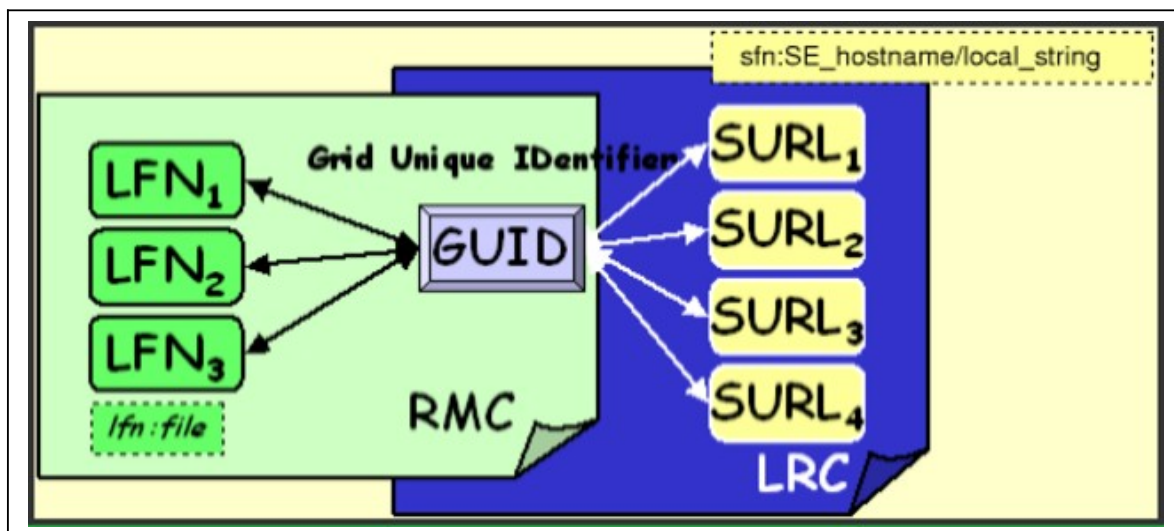
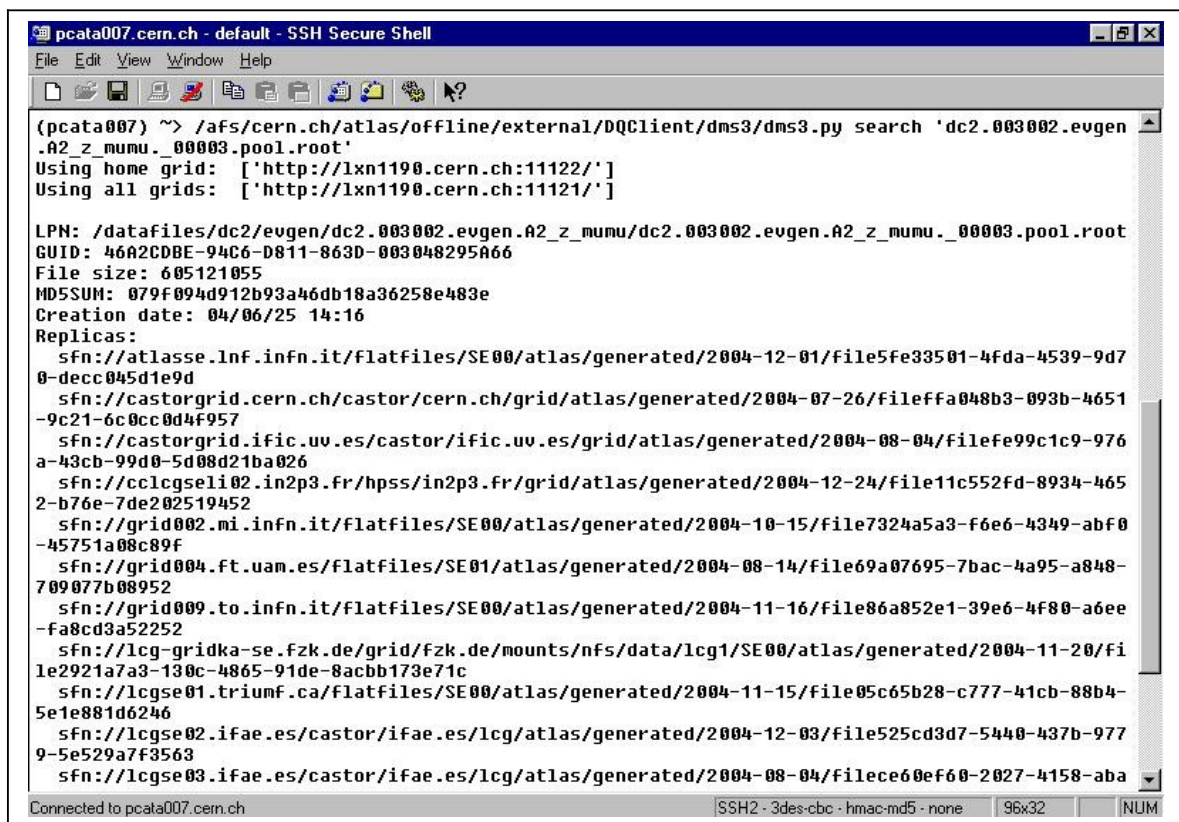


Figura 22 – Mapeamento de arquivos (Extraído de 10)

O usuário obtém a LFN desejada fazendo uma pesquisa no banco de dados AMI. O sistema GridTransfer une todas essas ferramentas de forma amigável ao usuário integrando todos os passos para a obtenção das ntuples armazenadas em Grid. Depois de obter a LFN desejada, o sistema executa o Don Quijote para transferir o arquivo para o diretório local e proporcionar acesso HTTP.

A execução do Don Quijote para procurar experimentos em Grid é apresentada na Figura 23. A nomenclatura é bem simples, tendo como parâmetros a palavra 'search' e o experimento (LFN). O aplicativo retorna as réplicas de arquivos espalhadas nos diversos institutos que armazenam o experimento escolhido.



```
pcata007.cern.ch - default - SSH Secure Shell
File Edit View Window Help

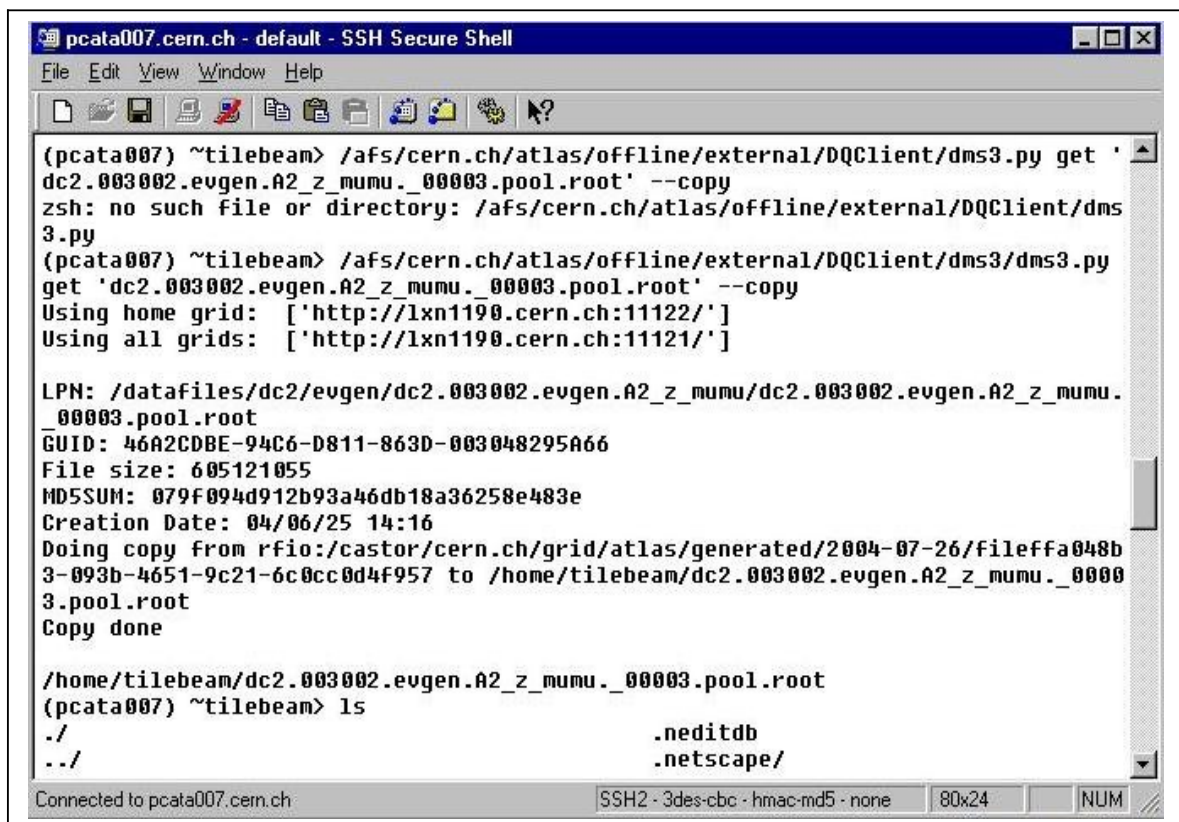
(pcata007) ~> /afs/cern.ch/atlas/offline/external/DQClient/dms3/dms3.py search 'dc2.003002.evgen
.A2_z_mumu._00003.pool.root'
Using home grid: ['http://lxn1190.cern.ch:11122/']
Using all grids: ['http://lxn1190.cern.ch:11121/']

LPN: /datafiles/dc2/evgen/dc2.003002.evgen.A2_z_mumu/dc2.003002.evgen.A2_z_mumu._00003.pool.root
GUID: 46A2CDBE-94C6-D811-863D-003048295A66
File size: 605121055
MD5SUM: 079f094d912b93a46db18a36258e483e
Creation date: 04/06/25 14:16
Replicas:
  sfn://atlasse.lnf.infn.it/flatfiles/SE00/atlas/generated/2004-12-01/file5fe33501-4fda-4539-9d7
  0-decc045d1e9d
  sfn://castorgrid.cern.ch/castor/cern.ch/grid/atlas/generated/2004-07-26/fileffa048b3-093b-4651
  -9c21-6c0cc0d4f957
  sfn://castorgrid.ific.uv.es/castor/ific.uv.es/grid/atlas/generated/2004-08-04/filefe99c1c9-976
  a-43cb-99d0-5d08d21ba026
  sfn://cclcgse.li02.in2p3.fr/hpss/in2p3.fr/grid/atlas/generated/2004-12-24/file11c552fd-8934-465
  2-b76e-7de202519452
  sfn://grid002.mi.infn.it/flatfiles/SE00/atlas/generated/2004-10-15/file7324a5a3-f6e6-4349-abf0
  -45751a08c89f
  sfn://grid004.ft.uam.es/flatfiles/SE01/atlas/generated/2004-08-14/file69a07695-7bac-4a95-a848-
  709077b08952
  sfn://grid009.to.infn.it/flatfiles/SE00/atlas/generated/2004-11-16/file86a852e1-39e6-4f80-a6ee
  -fa8cd3a52252
  sfn://lcg-gridka-se.fzk.de/grid/fzk.de/mounts/nfs/data/lcg1/SE00/atlas/generated/2004-11-20/fi
  le2921a7a3-130c-4865-91de-8acbb173e71c
  sfn://lcgse01.triumf.ca/flatfiles/SE00/atlas/generated/2004-11-15/file05c65b28-c777-41cb-88b4-
  5e1e881d6246
  sfn://lcgse02.ifae.es/castor/ifaes/lcg/atlas/generated/2004-12-03/file525cd3d7-5440-437b-977
  9-5e529a7f3563
  sfn://lcgse03.ifae.es/castor/ifaes/lcg/atlas/generated/2004-08-04/filece60ef60-2027-4158-aba
  NUM

Connected to pcata007.cern.ch SSH2 - 3des-cbc - hmac-md5 - none 96x32
```

Figura 23 – Execução de busca no Don Quijote

Para fornecer acesso HTTP através de um link hipertextual, o sistema precisa copiar o arquivo da Grid para o local onde o sistema GridTransfer é executado. Para isso é necessário executar o Don Quijote com os parâmetros ‘get’, o experimento e a opção –copy. A execução é mostrada na figura 24.



```
(pcata007) ~tilebeam> /afs/cern.ch/atlas/offline/external/DQClient/dms3.py get '
dc2.003002.evgen.A2_z_mumu._00003.pool.root' --copy
zsh: no such file or directory: /afs/cern.ch/atlas/offline/external/DQClient/dms
3.py
(pcata007) ~tilebeam> /afs/cern.ch/atlas/offline/external/DQClient/dms3/dms3.py
get 'dc2.003002.evgen.A2_z_mumu._00003.pool.root' --copy
Using home grid: ['http://lxn1190.cern.ch:11122/']
Using all grids: ['http://lxn1190.cern.ch:11121/']

LPN: /datafiles/dc2/evgen/dc2.003002.evgen.A2_z_mumu/dc2.003002.evgen.A2_z_mumu.
_00003.pool.root
GUID: 46A2CDBE-94C6-D811-863D-003048295A66
File size: 605121055
MD5SUM: 079f094d912b93a46db18a36258e483e
Creation Date: 04/06/25 14:16
Doing copy from rfio:/castor/cern.ch/grid/atlas/generated/2004-07-26/fileffa048b
3-093b-4651-9c21-6c0cc0d4f957 to /home/tilebeam/dc2.003002.evgen.A2_z_mumu._0000
3.pool.root
Copy done

/home/tilebeam/dc2.003002.evgen.A2_z_mumu._00003.pool.root
(pcata007) ~tilebeam> ls
./                               .neditdb
../                              .netscape/

Connected to pcata007.cern.ch: SSH2 - 3des-cbc - hmac-md5 - none 80x24 NUM
```

Figura 24 – Execução de cópia pelo Don Quijote

7.3.3 Requisitos para Utilização do Sistema GridTransfer

Os usuários que optarem por transferir usando GridFTP deverão ter o certificado emitido pela Autoridade Certificadora, além de instalar os pacotes de software indicados pela equipe de tecnologia da informação do LCG. Dentre eles, estão as ferramentas Globus segurança GSI para a autenticação do usuário na Grid e o GridFTP para a transferência do arquivo (globus-url-copy). As instruções e requisitos para configurar o computador para fazer a transferência GridFTP se encontram no site do LCG 10. É recomendado a instalação de quatro pacotes, dentre eles o vdt_globus_essentials-VDT1.2.0rh7gcc3-1.i386.rpm que contém ferramentas do projeto Globus. Além dos pacotes especificados na página do LCG, o usuário pode também instalar o Don Quijote, para procurar e transferir arquivos na Grid. Os pacotes de instalação podem ser encontrados no site do Don Quijote no CERN 10. Para efetuar a autenticação na Grid o usuário precisa também de acesso a uma User Interface (UI) 10. Para os usuários que desejam transferir usando Wget, devem ter o programa instalado em sua máquina 10.

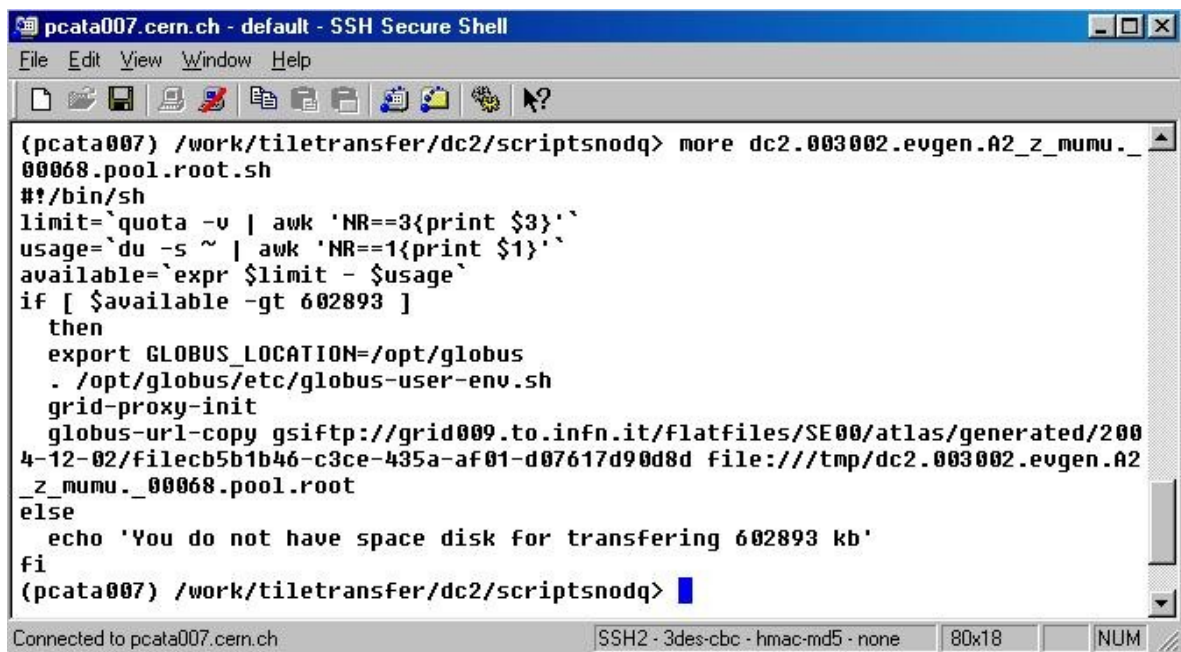
7.3.3.1 Opções para Transferência de Arquivos

O sistema oferece quatro formas de transferir os arquivos: HTTP, Wget, GridFTP e Don Quijote.

Para transferência através do protocolo HTTP, o usuário seleciona o(s) arquivo(s) e a opção de realizá-la imediatamente ou receber o link para o arquivo por e-mail. O usuário pode selecionar a opção para comprimir ou não o arquivo antes da transferência. Para a transferências por Wget, GridFTP e Don Quijote, o sistema oferece um script contendo os comandos que o usuário deve executar localmente para efetuar a transferência.

Para executar a transferência pelo GridFTP ou Don Quijote, o usuário deve, primeiramente, configurar as variáveis de ambiente. Depois, iniciar o proxy de comunicação para se autenticar. Finalmente executar o cliente Don Quijote ou globus-url-copy (cliente GridFTP) para a transferência. É importante ressaltar que esses comandos devem ser executados localmente no cliente, uma vez que a segurança GSI não permite que o servidor Web fizesse isso pelo cliente. Uma solução encontrada foi encapsular todos os comandos que devem ser executados no cliente em um script Shell.

A figura 25 apresenta um modelo de script para GridFTP e a figura 26 para Don Quijote. Em ambos os casos, há um teste para verificar se o usuário possui espaço em disco. Isso é realizado através dos comandos “quota” e “du” do Unix. Eles são armazenados em variáveis e subtraídos para verificar se o espaço em disco é suficiente. Após o teste, o script seta as variáveis de ambiente, realiza a autenticação na Grid (globus-proxy-init) e executa a transferência pelos clientes GridFTP (globus-url-copy) ou Don Quijote (/afs/CERN.ch/atlas/offline/external/DQClient/dms3/dms3.py).



```
(pcata007) /work/tiletransfer/dc2/scriptsnodq> more dc2.003002.evgen.A2_z_mumu._
00068.pool.root.sh
#!/bin/sh
limit=`quota -v | awk 'NR==3{print $3}`
usage=`du -s ~ | awk 'NR==1{print $1}`
available=`expr $limit - $usage`
if [ $available -gt 602893 ]
then
    export GLOBUS_LOCATION=/opt/globus
    . /opt/globus/etc/globus-user-env.sh
    grid-proxy-init
    globus-url-copy gsiftp://grid009.to.infn.it/flatfiles/SE00/atlas/generated/200
4-12-02/filecb5b1b46-c3ce-435a-af01-d07617d90d8d file:///tmp/dc2.003002.evgen.A2
_z_mumu._00068.pool.root
else
    echo 'You do not have space disk for transferring 602893 kb'
fi
(pcata007) /work/tiletransfer/dc2/scriptsnodq>
```

Figura 25 – Modelo de script GridFTP

```

pcata007.cern.ch - default - SSH Secure Shell
File Edit View Window Help
[Icons]
(pcata007) /work/tiletransfer/dc2/scripts> more dc2.003002.evgen.A2_z_mumu._0004
2.pool.root.sh
#!/bin/sh
limit=`quota -v | awk 'NR==3{print $3}'`
usage=`du -s ~ | awk 'NR==1{print $1}'`
available=`expr $limit - $usage`
if [ $available -gt 598089 ]
then
  grid-proxy-init
  /afs/cern.ch/atlas/offline/external/DQClient/dms3/dms3.py get 'dc2.003002.evge
n.A2_z_mumu._00042.pool.root' --copy --auto
else
  echo 'You do not have space disk for transferring 598089 kb'
fi
(pcata007) /work/tiletransfer/dc2/scripts>

```

Connected to pcata007.cern.ch SSH2 - 3des-cbc - hmac-md5 - none 80x15 NUM

Figura 26 – Modelo de script Don Quijote

No caso dos experimentos não armazenados em Grid, que é o caso dos Calorímetros de telhas e Argônio Líquido (Lar) e Combined Testbeam, o sistema executa os comandos nsls, stagein para disponibilizar o arquivo no CASTOR. Depois disso, apresenta ao usuário o script Shell com os comandos para a autenticação do usuário com o CASTOR e para executar a transferência GridFTP, que neste caso passa a ser o globus-url-copy. O Don Quijote não seria utilizado nesse caso, pois o experimento não está armazenado em Grid.

A tabela a seguir apresenta as ações do sistema para cada método de transferência escolhido pelo usuário.

	HTTP	WGET	GRIDFTP	DON QUIJOTE
Alternativa de comprimir	Sim	Sim	Não	Não
Envia resultado por e-mail	Sim	Sim	Sim	Sim
Transfere para pcata007	Sim	Sim	Não	Não
Autenticação na Grid	Não	Não	Sim	Sim
Fornece Script (execução pela linha de comando)	Não	Sim	Sim	Sim

Tabela 2: Ações executadas pelo sistema de acordo com a opção do usuário

7.3.4 Diagramas de Fluxo de Dados

Esta seção descreve a especificação do sistema GridTransfer através de diagramas de fluxo de dados.

7.3.4.1 Diagrama de Contexto

A figura 27 apresenta o Diagrama de contexto do sistema GridTransfer, que se relaciona com Usuário, Castor, Don Quijote e Ami_Interface. Primeiramente, deve-se justificar que a conexão dos terminadores com armazenamentos de dados nos diagramas foram feitos com o intuito de facilitar a compreensão do leitor. Isso ocorre porque as entidades Castor, Don Quijote e Ami_interface são externas ao sistema. Desta forma, os processos de obtenção dos dados por essas entidades não foram representados.

O sistema interage com o usuário através de escolhas que o mesmo submete, tais como, tipo de experimento, características, método de transferência, arquivos e forma de download (no momento ou receber o resultado por email). Para fornecer informações sobre os experimentos e dados gerados, o sistema se relaciona com a Ami_Interface, a qual possui acesso à base de dados que contém essas informações dos testes realizados. As mesmas serão apresentadas através de uma tabela em uma página HTML ao usuário. De acordo com sua opção o sistema irá buscar os dados dos experimentos na Grid através do Don Quijote ou no Castor, dependendo do tipo de experimento. Ao final do processo, o sistema produz um link ou um script para que o usuário possa obter os dados.

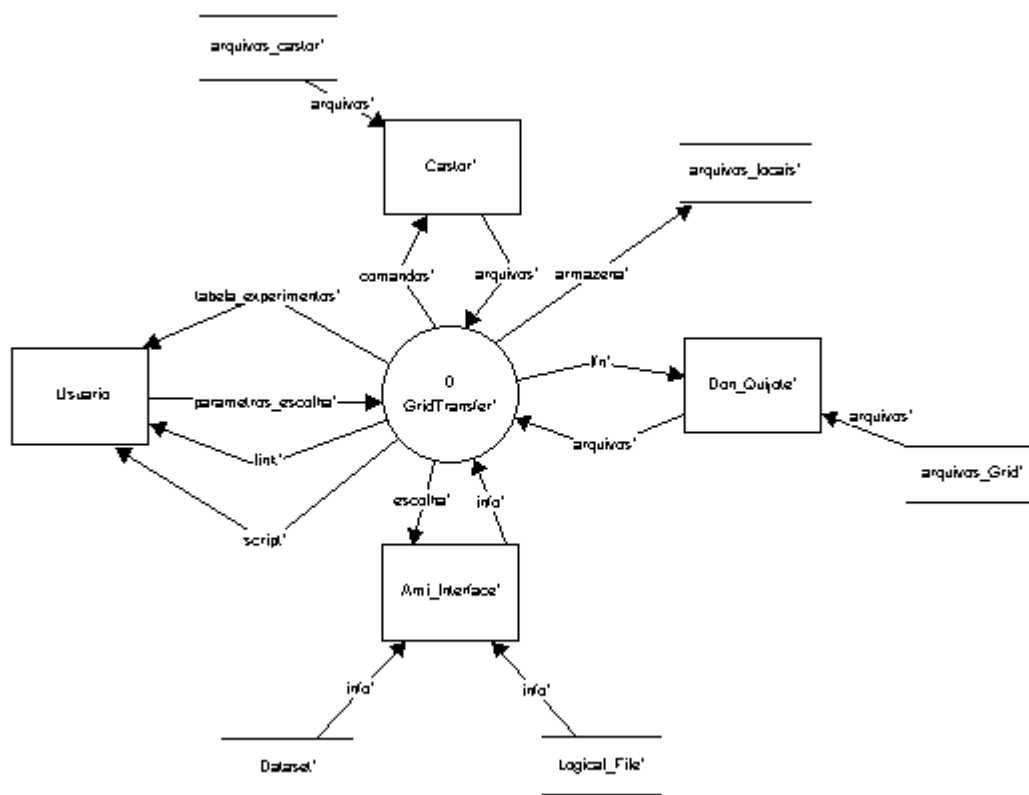


Figura 27 – Diagrama de contexto

7.3.4.2 Diagrama de Fluxo de Dados Nível 1

O sistema GridTransfer se subdivide em quatro módulos. Cada módulo é responsável por um tipo de experimento. Dessa forma, o módulo TileTransfer disponibiliza dados gerados nos experimentos realizados no calorímetro de telhas. Já os módulos LarTransfer, CombinedTransfer e DCTransfer disponibilizam dados do calorímetro de argônio líquido, testes de feixes combinados e os experimentos do DataChallenge2 respectivamente.

Nota-se, na figura 28, que só o DCTransfer se relaciona com Don Quijote, uma vez que somente os experimentos do DataChallenge2 estão armazenados em Grid. Os outros módulos se relacionam com o Castor, onde estão armazenados os dados do calorímetro de telhas, calorímetro Lar e testes de feixe combinados.

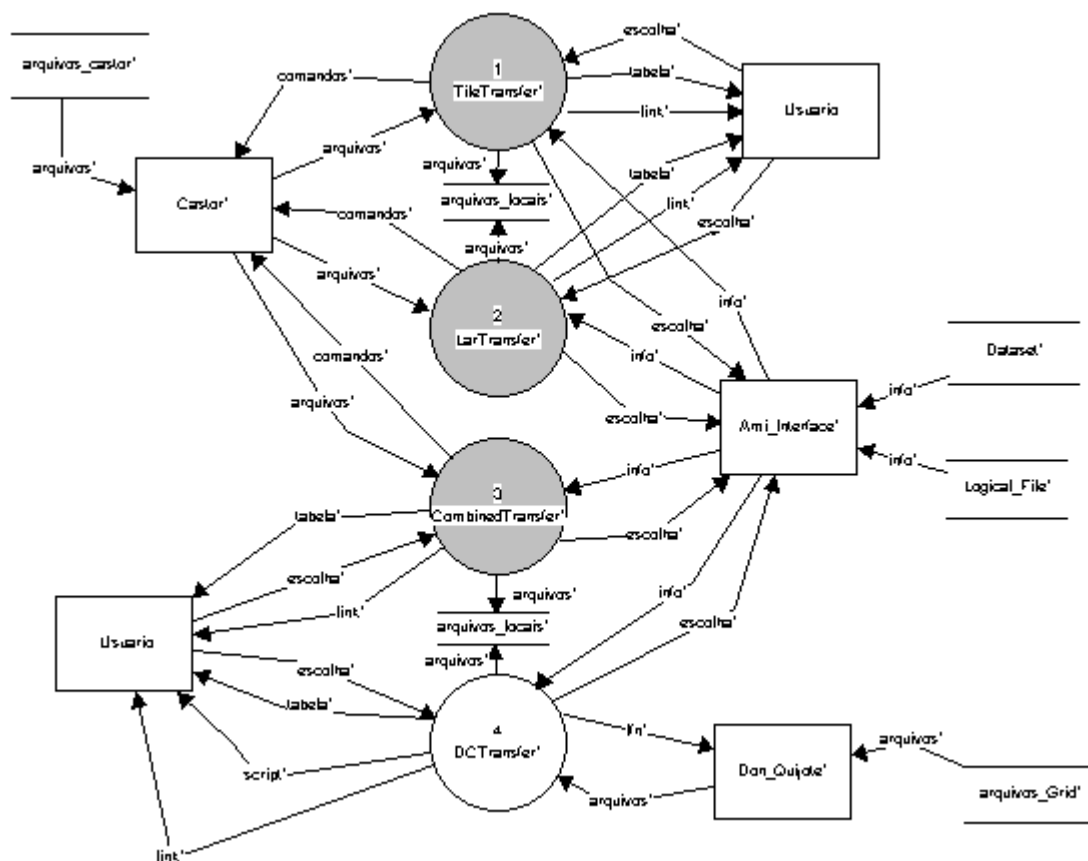


Figura 28 – DFD nível 1

7.3.4.3 Diagrama de Fluxo de Dados Nível 2

A figura 29 apresenta o diagrama de fluxo de dados do módulo DCTransfer. Inicialmente o usuário se relaciona com o módulo Pesquisa, que apresenta uma página com os parâmetros, que são os atributos das entidades “dataset” e “logicalFile”, onde o usuário insere as características dos experimentos ou os dados que deseja analisar. O sistema submete uma busca através da Ami_interface e obtém um arquivo contendo o resultado da pesquisa, que são as informações sobre o conjunto dos testes desejados. Essas informações são apresentadas ao usuário em uma tabela na página HTML. Depois disso, o usuário se relaciona com o módulo Acesso selecionando quais arquivos de dados deseja obter. O módulo Acesso irá enviar a lfn correspondente do arquivo solicitado ao Don Quijote para que possa procurar e transferir o arquivo armazenado na Grid. A lfn do arquivo foi obtida pelas informações adquiridas na busca na tabela de logicalFile pela Ami_interface.

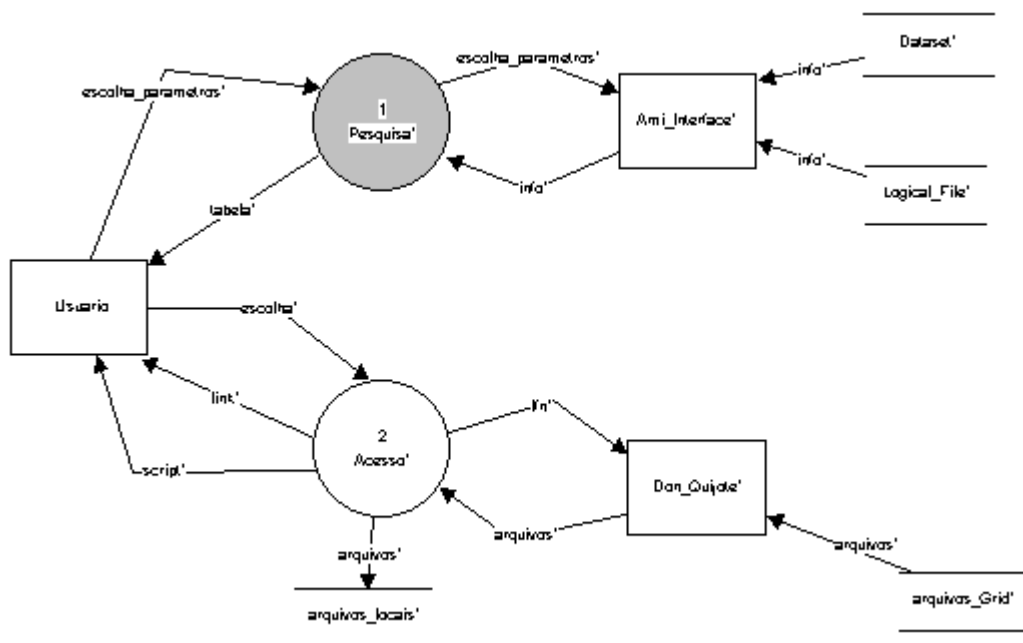


Figura 29 – DFD nível 2

7.3.4.4 Diagrama de Fluxo de Dados Nível 3

Esta seção mostra o diagrama de fluxo de dados do módulo Acesso na figura 30. É neste módulo que o usuário escolhe se os resultados do processo de obtenção dos arquivos serão apresentados na página do navegador ou encaminhados por e-mail. Como o processo é demorado, a opção de receber por e-mail se torna interessante, visto que o usuário pode fechar a janela do navegador e verificar depois na sua caixa de e-mail se o arquivo foi disponibilizado conforme sua solicitação. Caso o usuário queira acompanhar o processo, com mensagens informativas, o sistema oferece a opção de obter o resultado pela página web.

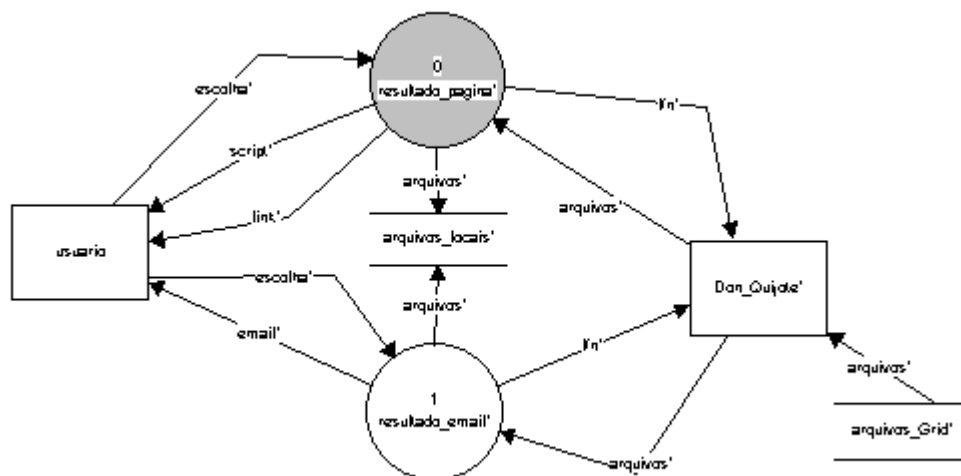


Figura 30 – DFD nível 3

7.3.4.5 Diagrama de Fluxo de Dados Nível 4

Além de escolher se deseja os resultados na página ou por e-mail, o usuário deve escolher também o método de transferência, ou seja, se fará o download utilizando o protocolo HTTP, Wget, GridFTP ou Don Quijote. Dependendo da escolha, o sistema retornará um link ou script ao usuário. Nos módulos HTTP e Wget, o sistema acessa o Don Quijote, que transfere o arquivo da Grid para o diretório local. Dessa forma é possível oferecer acesso HTTP através de um link. Para o Wget, o sistema oferece um script com o comando contendo o link para fazer a transferência. Já no caso do GridFTP e Don Quijote, o sistema acessa o Don Quijote para verificar se o arquivo existe na Grid. Se existir, cria um script com os comandos necessários para que o usuário possa fazer a transferência via GridFTP ou Don Quijote respectivamente.

A figura 31 representa o módulo Resultado_pagina.

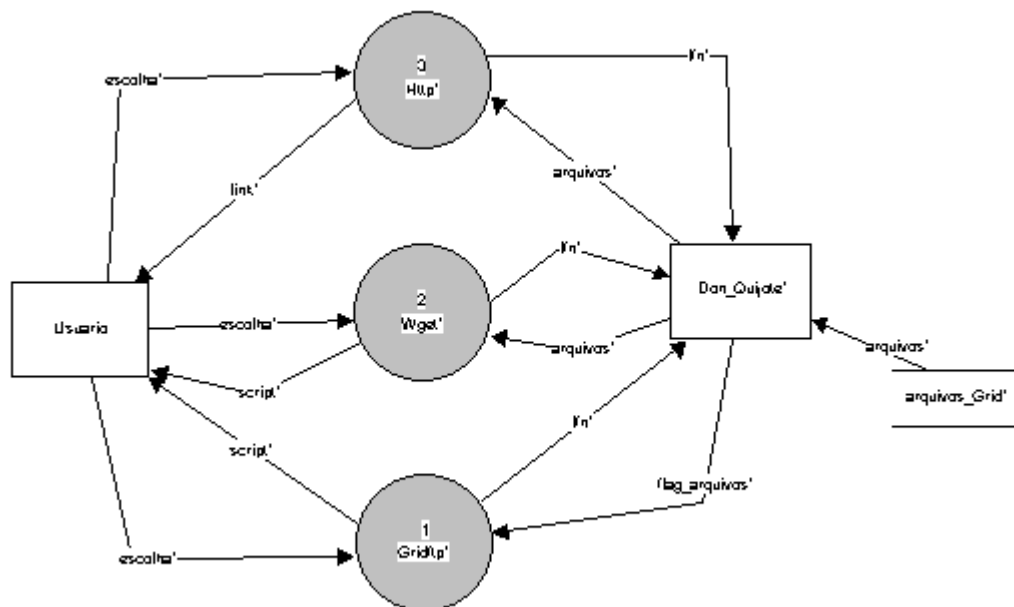


Figura 31 – DFD nível 4 (módulo Resultado_pagina)

O módulo Resultado_email, apresentado na figura 32, inicialmente chama o módulo Processo-filho, que irá criar um processo-filho para executar o processamento em outro programa no sistema operacional. Dessa forma o usuário não receberá o resultado na página. Assim como no Resultado_pagina, o usuário seleciona o método de transferência. O que difere do módulo anterior é que o sistema enviará o link ou script ao usuário por e-mail através do processo Envia_email.

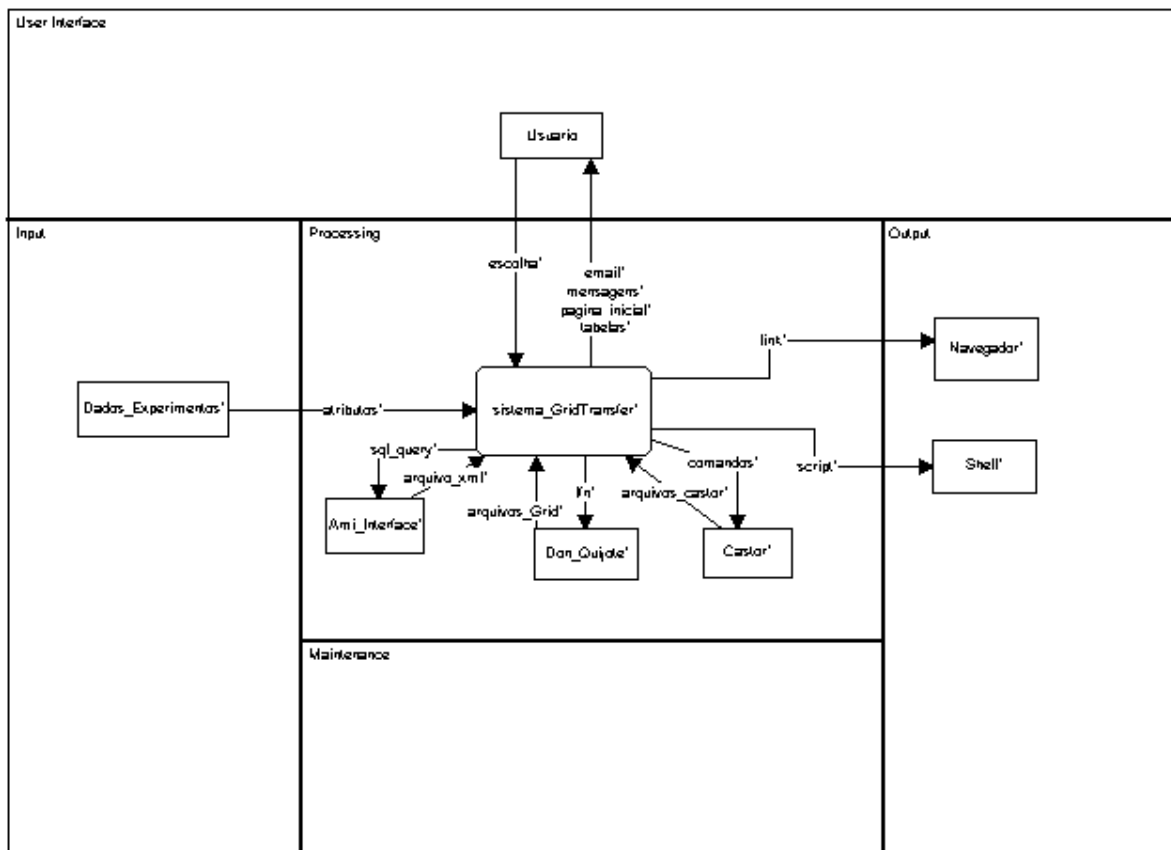


Figura 33 – Arquitetura do sistema GridTransfer

7.4.1.2 Sistema GridTransfer

A figura 34 mostra os módulos que compõem o sistema e como se relacionam entre si, com o usuário e entidades externas. Os módulos são executados exatamente na ordem apresentada: início, pesquisa e acesso. O módulo de pesquisa se relaciona com a **Ami_Interface** e o módulo Acesso com **Don_Quijote** e **Castor**. Ao final do processo, um link ou script é gerado. Nas próximas seções, os módulos que constituem o sistema serão explicados mais detalhadamente.

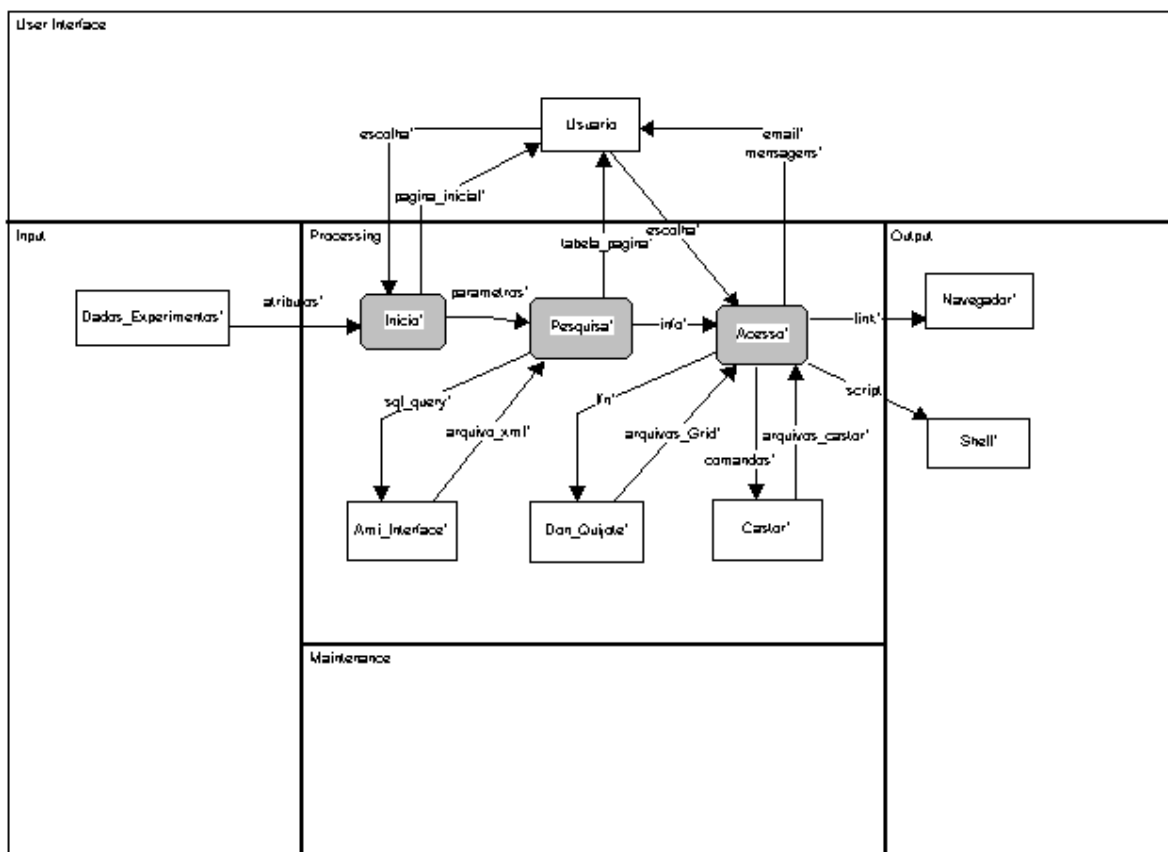


Figura 34 – Módulos do sistema GridTransfer

7.4.1.3 Módulo Início

O módulo Início é o primeiro contato do usuário com o sistema GridTransfer. O sistema apresenta uma página onde o usuário pode escolher qual tipo de experimento (TileCalorimeter, LarCalorimeter, CombinedTransferBeam ou DataChallenge2) deseja analisar. Após a escolha, o sistema consulta um arquivo que contém dados (atributos) sobre o experimento selecionado e os apresenta em uma página HTML com entradas (inputs), onde o usuário poderá inserir seus valores de busca.

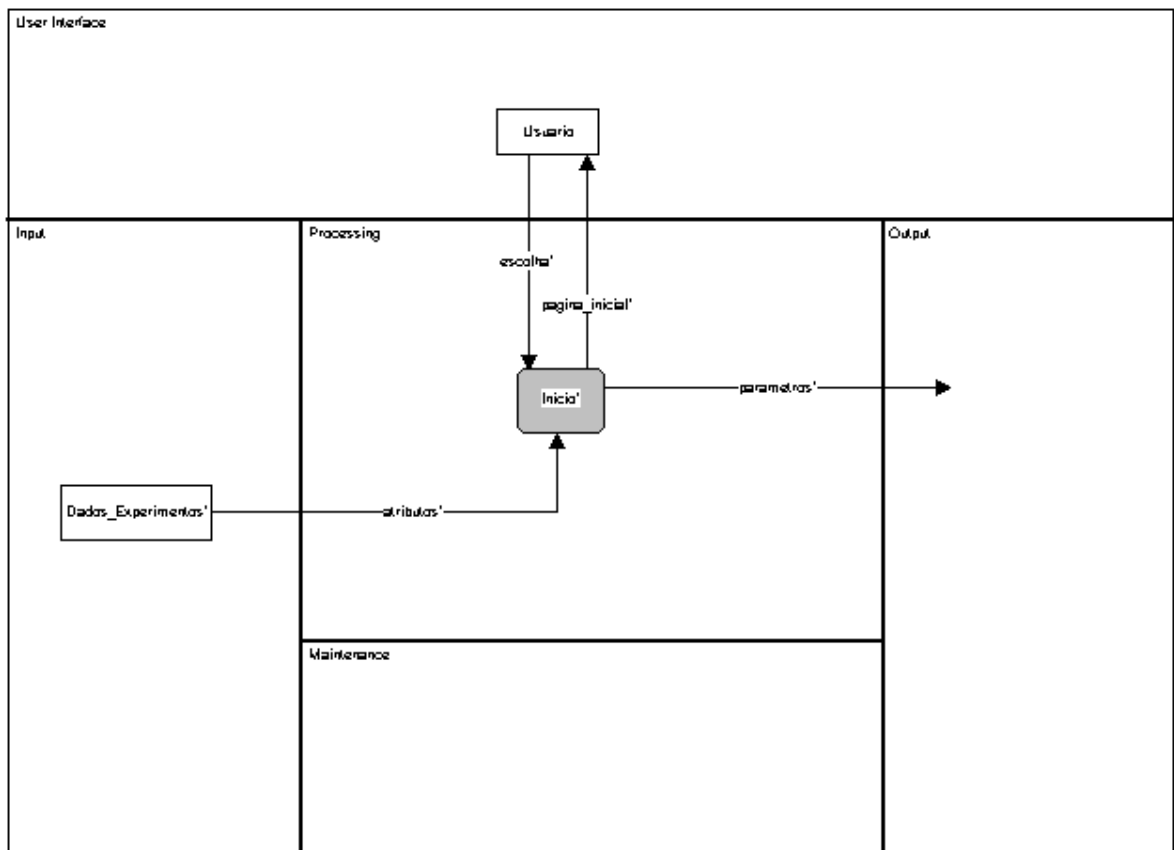


Figura 35 – Arquitetura do módulo Início

7.4.1.4 Módulo Pesquisa

O módulo Pesquisa recebe do módulo Início os parâmetros, que são os valores dos atributos dos experimentos e os valores escolhidos pelo usuário para a busca. O sistema monta a `sql_query` que será enviada à `Ami_Interface` para a obtenção das informações sobre os experimentos desejados, que são recebidas através de um arquivo xml que retorna da `Ami_Interface`. O sistema então converte os dados xml através de um arquivo xsl e monta uma tabela que será apresentada ao usuário na página. É importante ressaltar que esse módulo fornece as informações dos experimentos pesquisados aos próximos módulos.

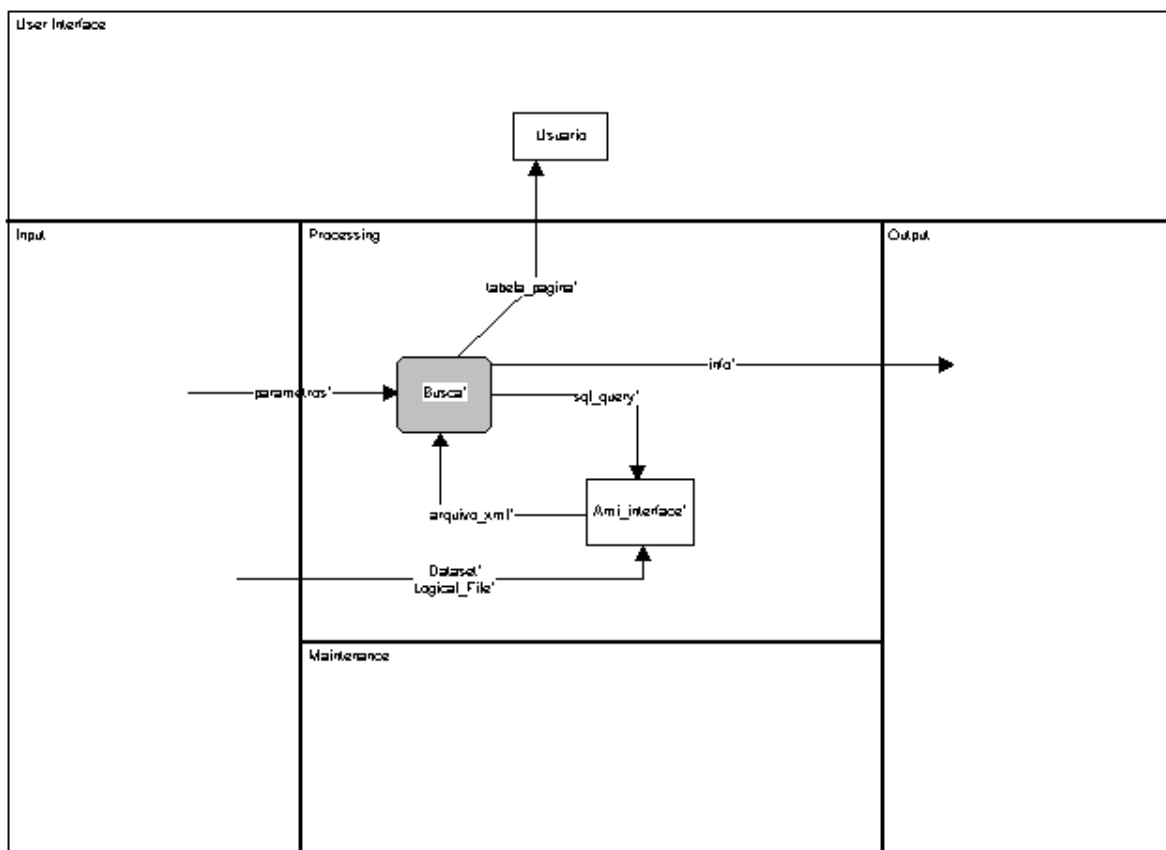


Figura 36 – Arquitetura do módulo Pesquisa

7.4.1.5 Módulo Acesso

O módulo Acesso recebe as escolhas do usuário, dentre elas o método de transferência, a forma de resultado (na página ou por e-mail) e os arquivos desejados. Além disso, recebe as informações dos experimentos vindas do módulo de pesquisa, das quais constam as lnfs ou localizações de arquivos no Castor. As escolhas recebidas passam por um processo de validação, onde são verificados se o usuário escolheu corretamente os arquivos, ou se inseriu o e-mail no formato correto. Dependendo da escolha do usuário, o sistema seleciona o módulo Resultado_pagina ou Resultado_email que serão detalhados mais adiante. Esses módulos se relacionam com Castor e Don Quijote para poder disponibilizar os arquivos desejados pelo usuário fornecendo, ao final do processo, links ou scripts, na página HTML ou no e-mail que será enviado. Os arquivos recebidos pelo sistema são armazenados no diretório local e são deletados periodicamente.

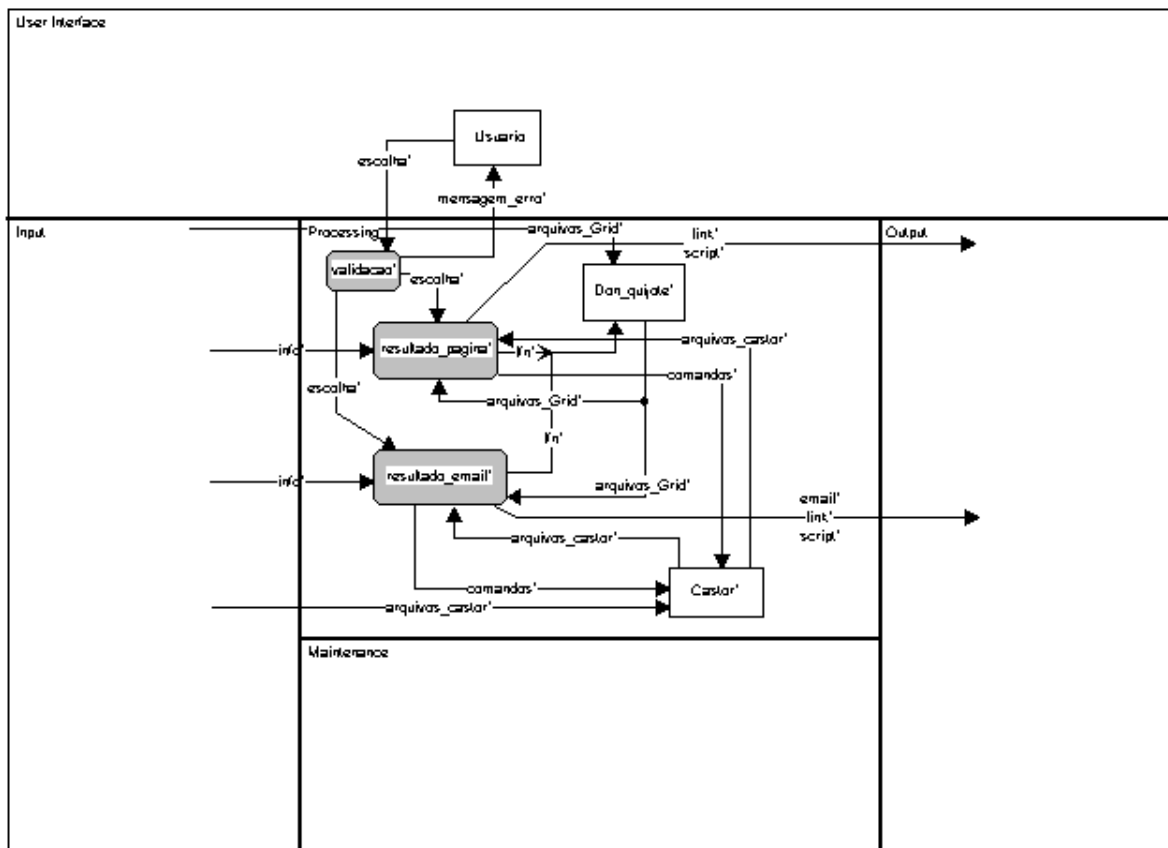


Figura 37 – Arquitetura do módulo Acesso

7.4.1.6 Módulo Resultado_pagina

O módulo resultado_pagina é chamado quando o usuário escolhe fazer o download naquele exato momento. Sendo assim, o sistema executa os processos necessários para a obtenção dos arquivos e vai informando ao usuário, em tempo real, escrevendo mensagens na página web. Ao receber a opção de método de transferência, o sistema chama um dos três módulos (HTTP, Wget ou GridFTP), que irão executar comandos no Castor se os arquivos estiverem no Castor, ou no Don_Quijote caso estejam na Grid. Como saída, temos o link (HTTP) ou script (Wget e GridFTP).

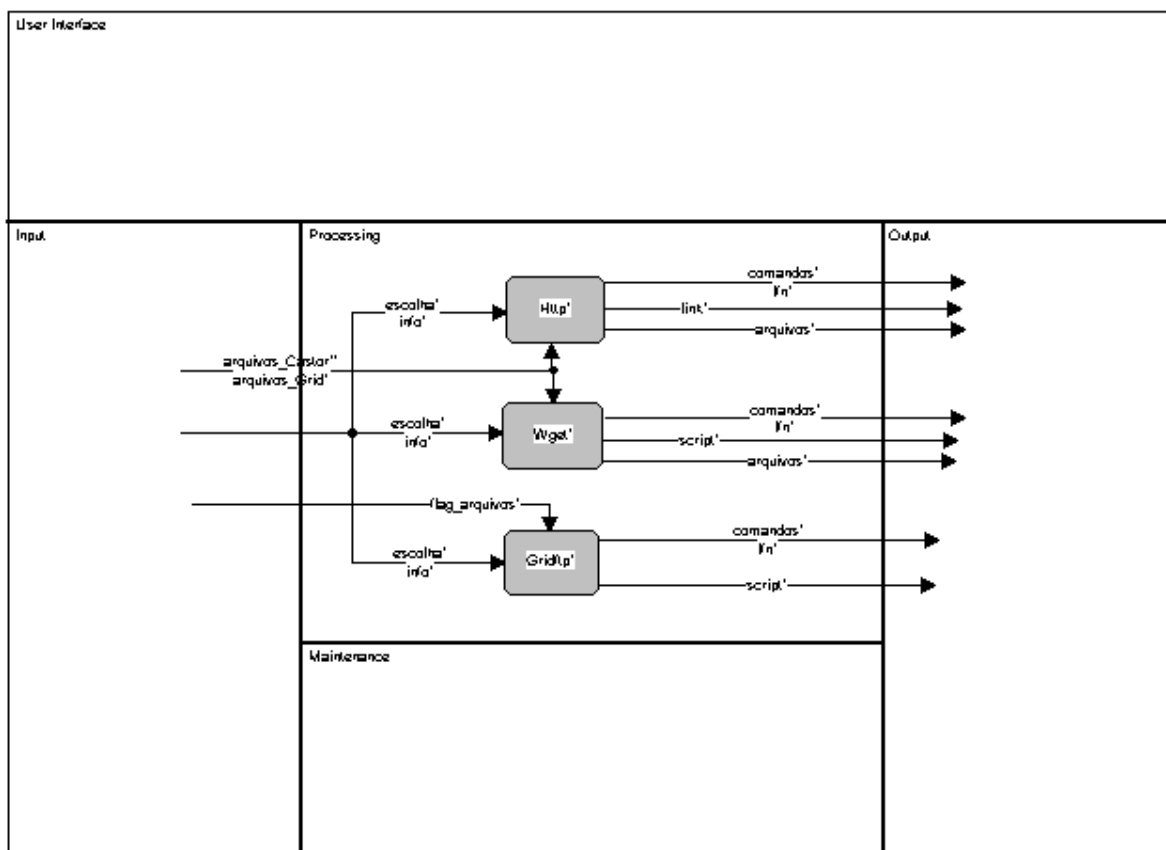


Figura 38 – Arquitetura do módulo Resultado_pagina

7.4.1.7 Módulo Resultado_email

O módulo Resultado_email é chamado quando o usuário escolhe receber os resultados por e-mail. Desta forma, o sistema cria um processo-filho e passa os parâmetros de escolha do usuário. O processo filho executará os mesmos processos relacionados no módulo Resulta_pagina, porém o usuário não acompanhará a execução na página, podendo até fechá-la. Após o processo, o sistema envia um e-mail ao usuário com o link ou script para que ele faça o download do arquivo desejado.

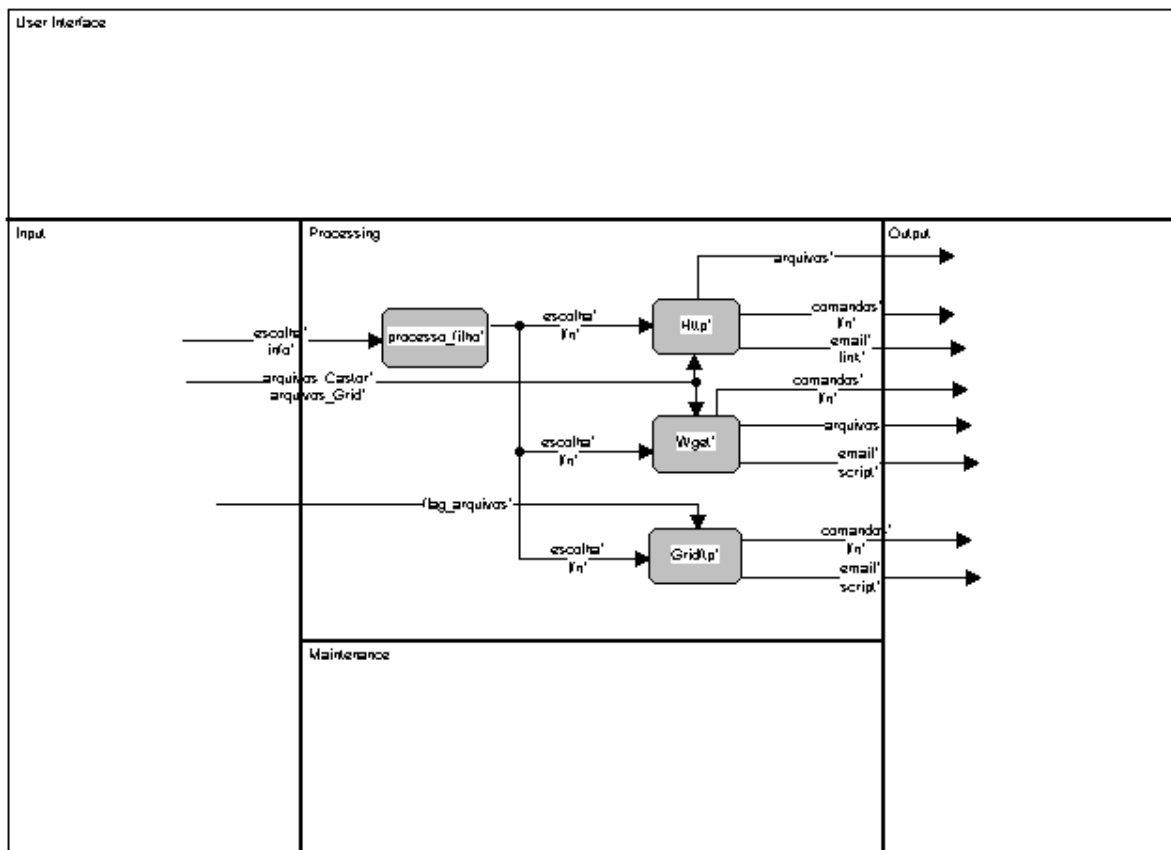


Figura 39 – Arquitetura do módulo Resultado_email

7.4.2 Projeto Detalhado e Implementação

Nesta seção será detalhado o projeto de cada módulo, descritos utilizando a linguagem de programação usada na implementação.

7.4.2.1 Módulo Início

O usuário, ao entrar na página [HTTP://tileinfo.web.CERN.ch/tileinfo/lps/CTB/VGRID/](http://tileinfo.web.CERN.ch/tileinfo/lps/CTB/VGRID/) tem que escolher entre quatro tipos de experimentos em um menu. Para apresentá-lo, o sistema utiliza um programa processador chamado Xalan 10, que transforma os dados recebidos XML em HTML para que o usuário possa visualizá-los na página. Isso é feito através da linguagem XSL.

Sendo assim, os dados organizados no arquivo index.xml, contendo os nomes de cada experimento, são trabalhados usando o index.xsl que monta a página inicial com o menu para o usuário escolher o tipo de experimento. O comando utilizado é o seguinte:

```
JAVA_BIN."/javac -classpath ".XALAN_BIN."; ".JAVA." org.apache.xalan.xslt.Process
-IN index.xml -XSL index.xsl";
```

Onde JAVA_BIN, XALAN_BIN e JAVA são constantes definidas em um arquivos php como o caminho onde encontra o javac, xalan e java respectivamente.

O arquivo XSL deve montar a página HTML com um formulário contendo o menu para que o usuário selecione o experimento.

Ao selecionar o experimento, o sistema executa o arquivo ctbrunsearch.php que irá chamar um arquivo XSL (detectors.xml) da mesma forma que foi feita na página inicial. Esse arquivo utiliza os dados do detectors.xml que contém os atributos do tipo de experimento selecionado e monta uma página com entradas (inputs) para que o usuário possa fazer uma pesquisa de experimentos.

O comando abaixo mostra a execução mencionada acima:

```
passthru(JAVA." ".XALAN." -IN detectors.xml -XSL detectors.xml -PARAM name $subsystemName", $retCode);
```

7.4.2.2 Módulo Pesquisa

Ao visualizar a página com inputs, o usuário deve colocar nos campos os valores que serão utilizados para fazer a busca na base de dados. O sistema, então, monta uma string que será a sql_query. Isso é feito no ctbrunselect.php usando funções de manipulação de strings.

A próxima etapa é se conectar ao banco de dados. Isso é feito, primeiramente, usando informações sobre as conexões de cada tipo de experimento, tais como nome, projetos, processing_steps, tabelas, contidas no arquivo dbconnection.xml. Para fazer a conexão é utilizado um cliente Java passando, como parâmetros, as informações obtidas anteriormente, bem como a sql_query. O comando abaixo mostra a execução do cliente Java:

```
snprintf(command, 10*SIZE, "export CLASSPATH=%s:$CLASSPATH; %s %s -command=SearchQuery -project=%s -processingStep=%s -sql=\"%s\" > tmp%s", AMI_CLASS, JAVA, AMI, argv[1], argv[2], argv[3], filename);
```

Onde, “AMI_CLASS” é a classe compilada que contém as funções para a conexão com a base de dados AMI e “AMI” é o endereço de um arquivo de configuração utilizado para conexão.

O cliente retorna dados em XML que são direcionados (> tmp%s) para um arquivo temporário. O sistema verifica se ocorreu algum erro nesse processo (ctbAmiError.xml). Se ocorrer, envia um alerta ao usuário. Se não, o próximo passo é ler esse arquivo de dados XML e montar uma tabela com checkboxes para que o usuário possa selecionar os arquivos que deseja baixar. Isso é feito da mesma forma apresentada anteriormente, usando o programa Xalan e a linguagem XSL:

```
$command = JAVA." ".XALAN." -IN $filename -XSL ".$connection["name"].".xml -PARAM start $start -PARAM end $end -PARAM subsystemName";
```

Existe um arquivo XSL para cada experimento ("\$connection["name']"). São passadas variáveis contendo a quantidade de arquivos que devem ser apresentados na tabela (start e end).

Além da tabela, o XSL apresenta um formulário para que o usuário possa inserir sua opção de método de transferência, se quer o arquivo comprimido ou não e se quer receber o download na página ou por e-mail.

7.4.2.3 Módulo Acesso

O módulo de acesso é a sequência do módulo de pesquisa, ou seja, após serem apresentados os arquivos para download na tabela, o usuário seleciona quais deseja, o método de transferência, se quer comprimido e o resultado na página ou por e-mail. Após submeter suas escolhas, o sistema passa por um módulo de validação (validate.js), onde é verificado se o usuário escolheu ou não um experimento (se não escolheu envia um alerta javascript) e se inseriu o e-mail na forma correta (palavra@palavra.palavra). É importante ressaltar que o módulo validação foi feito em Javascript, linguagem apropriada para executar código na máquina do cliente, evitando processamento no servidor.

O módulo de acesso foi desenvolvido em um arquivo php (ctbgetfiles.php) que chama diversos programas feitos em C que serão detalhados posteriormente.

Ele recebe em variáveis as escolhas citadas anteriormente. Se o usuário escolheu o resultado na página, o módulo resultado_pagina é chamado. Porém, se o mesmo escolheu receber por e-mail, o módulo de acesso chama o módulo resultado_email, que será executado em um programa C:

```
passthru ("./ctbSendMailDc2 $mailFile $compress $numOfEmails $email $transferMethod  
> /dev/null", $retCode);
```

Esses módulos são responsáveis por disponibilizar os arquivos escolhidos pelo usuário, fornecendo um link ou script para o mesmo. Isso é feito através da interface com o Castor ou Don Quijote. Isso tudo será detalhado no módulo resultado_pagina.

7.4.2.4 Módulo Resultado_pagina

O primeiro passo é verificar qual é o tipo de experimento, porque se for TileCalorimeter, LarCalorimeter e CombinedTestBeam, o sistema se comunicará com o Castor para baixar os arquivos. Se for DataChallenge2, o sistema utiliza a interface Don Quijote para obter os arquivos que estão armazenados na Grid.

Como a comunicação com o Castor já foi detalhada no projeto do sistema TileTransfer na seção Histórico, irei apenas detalhar o processo de comunicação com o Don Quijote.

Para poupar tempo de processamento, o sistema verifica primeiro se o arquivo desejado já consta no diretório local pela função `dc2_searchNtuple(char *lfn, char *compress)`. Se já estiver, cria o link ou o script, se não fará o download através do Don Quijote.

O sistema, por restrição de espaço em disco, apaga regularmente os arquivos, usando o programa `ctbDeleteFiles`.

O cliente Don Quijote pode ser acessado diretamente pelo sistema através do endereço `/afs/CERN.ch/atlas/offline/external/DQClient/dms3/dms3.py` declarado no `ctbConsts.h` como 'DQ'.

Se o usuário selecionar HTTP ou Wget, o sistema fará o download do arquivo armazenado em Grid para o diretório local chamando o programa `ctbDqget`, passando a `lfn` correspondente ao arquivo. Esse programa executa o don Quijote dessa forma:

```
snprintf (command, MAX_LENGTH_FILENAME, "%s get '%s' --copy", DQ, lfn);
retCode = system (command);
```

Neste momento é feito o tratamento de erros do Don Quijote. Os principais erros que ocorrem são quando o don quiote não encontra nenhuma réplica do arquivo solicitado, por falha na conexão com o Don Quijote e erro se não transferir para o diretório local do sistema. Todos eles são apresentados ao usuário.

Se o usuário escolheu comprimir o arquivo o programa `gzip` é executado:

```
snprintf (command, MAX_LENGTH_FILENAME, "gzip %s/%s", DC2_DIRECTORY,
lfn);
```

A diferenciação entre os módulos HTTP e Wget ocorre na hora de fornecer o link ou script, sendo que o HTTP fornece link e Wget script.

Para HTTP, o programa PHP cria um link através da função `echo`:

```
Echo          "<a          href=\"download.php?file=\".DC2_DIRECTORY.          \"/"
$lfn[$index]\".\".gz\".\">\".\"$lfn[$index]\".\"</a> ";
```

Já para Wget, a função `dc2_Wgetscriptmaker` é chamada. Ela basicamente cria um arquivo `.sh` com o comando `Wget -c <endereço HTTP do arquivo>`:

```
snprintf (stringdq, MAX_LENGTH_FILENAME, "Wget -c %s", link);
```

Se o usuário selecionar GridFTP, o sistema faz uma busca usando o Don Quijote e apresenta as Surls (nomes físicos) na página para o usuário. Ele escolhe uma delas e o sistema monta o script com o comando `globus-url-copy` acrescido da seleção do usuário. Depois disso, é executada a função `ctbGridFTPScriptMaker` que vai criar, da mesma forma que no Wget, um arquivo `.sh` com os comandos que o usuário deverá executar em sua máquina para fazer a transferência GridFTP.

O usuário também pode selecionar transferir usando Don Quijote, se o tiver instalado em sua máquina. O sistema também monta o script com o comando de transferência do Don Quijote através da função `ctbDqScriptMaker`.

Ao final do processo, o sistema cria um link para o script na página:

```
echo          "<a          href=\"download.php?file=\".DC2_SCRIPTS.          \"/
$lfm[$index]". ".sh". "\">". "$lfm[$index].sh". "</a> ";
```

7.4.2.5 Módulo Resultado_email

O módulo `resultado_email` recebe os parâmetros do módulo `Acesso` e cria um processo filho, onde será executada a maioria dos processos chamados no módulo `resulta_pagina`. O processo filho é criado pela função `Fork` do Unix.

```
pid=fork();
if (pid < 0 )
{
    printf("error id=%d\n",pid);
    exit(PROCESS_ERROR);
}

else if (pid)
{
    //printf("parent\n");
    exit(OK);
}

else
{
    //CHILD PROCESS
...

```

Dentro do processo filho, os módulos `HTTP`, `Wget` e `GridFTP`, mostrados anteriormente, são chamados. Porém, o resultado (links e scripts) não é disponibilizado na página, mas enviado por e-mail utilizando o programa `Sendmail`. Isso é feito abrindo um novo processo em paralelo:

```
p = popen("/usr/sbin/sendmail -t","w");
...
```

Dependendo do programa de e-mail que o usuário utiliza, só um formato de mensagem (HTML ou texto) poderá ser visualizado. O programa envia, então, o e-mail nos dois formatos.

Alguns detalhes são importantes. Na mensagem HTML, o caracter espaço deve ser substituído por “=20” e o caracter ‘:’ por =3D. Já na mensagem texto isso não ocorre.

7.5 Execução

O sistema GridTransfer se encontra na página “<http://tileinfo.web.cern.ch/tileinfo/lps/CTB/VGRID/>”. Conforme mostrado na figura a seguir, na página inicial, o usuário pode selecionar o tipo de teste realizado no detector ATLAS.

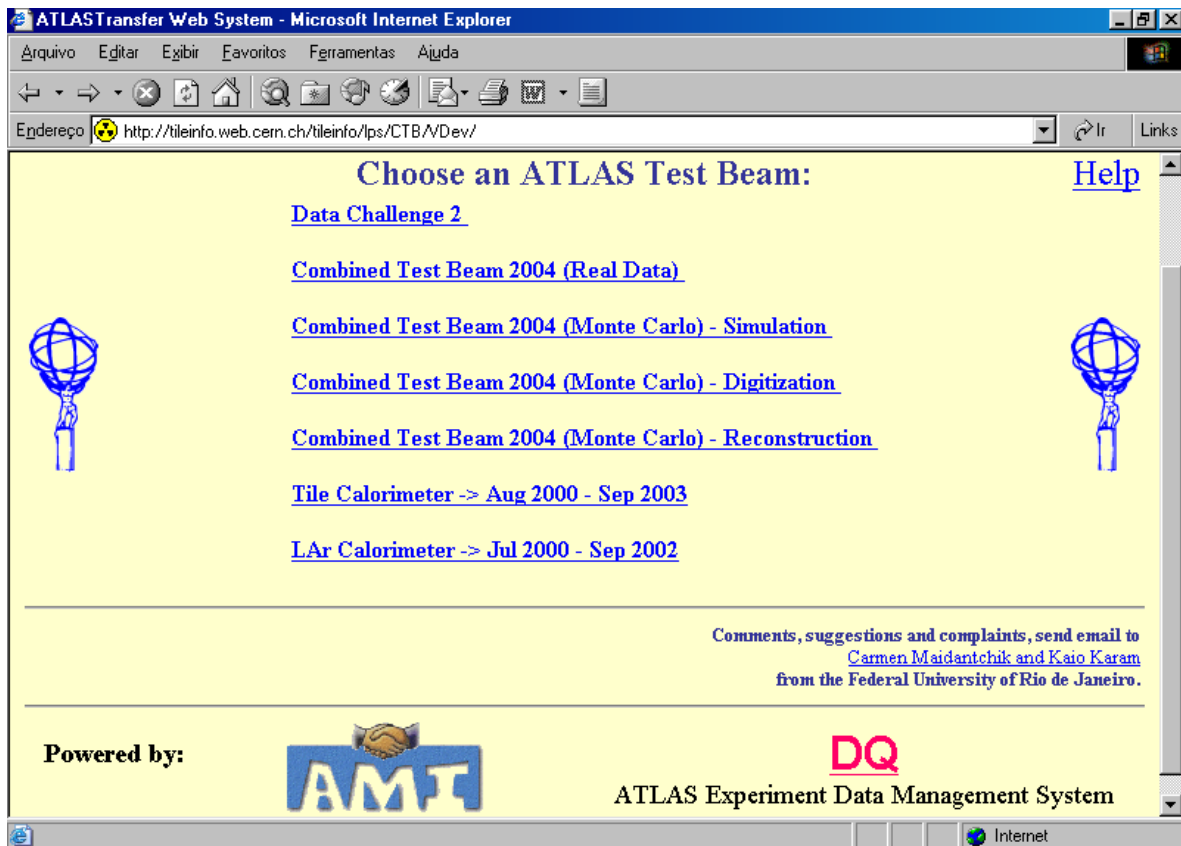


Figura 40 – Página Inicial do sistema GridTransfer

Em seguida, na figura 41 é apresentada uma página de busca com os atributos do experimento selecionado, onde o usuário insere os dados com as características dos testes que deseja analisar.

ATLASTransfer - ATLAS Data Challenge 2 Information Database - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço <http://tileinfo.web.cern.ch/tileinfo/lps/CTB/VDDev/ctbRunSearch.php?processName=dc2%20dc2T%20.xml> Ir Links

Name	Expression	Display	Order	Name	Expression	Display	Order
datasetNumber		☒	☐	logicalDatasetName		☒	☐
projectName		☒	☐	principalPhysicsGroup		☒	☐
datasetGUIDHigh		☐	☐	datasetGUIDLow		☐	☐
dataType		☒	☐	detectorLayout		☒	☐
lastModified		☐	☐	logicalDatasetName		☐	☐
luminosity		☐	☐	physicsShort		☐	☐
principalPhysicsGroup		☒	☐	prodsysCreated		☒	☐
prodsysModification		☒	☐	LFN		☐	☐
productionSite		☒	☐	logicalCollection		☒	☐

arrange in order, show rows/page

Concluído Internet

Figura 41 – Página de Pesquisa do sistema GridTransfer

Ao submeter a busca, o sistema consulta o banco de dados correspondente ao módulo do teste escolhido na tela inicial e disponibiliza no navegador os resultados da pesquisa, conforme apresenta a figura 42.

ATLAS Data Challenge 2 Run Information Database - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço <http://tileinfo.web.cern.ch/tileinfo/lps/CTB/VDev/ctbRunSelect.php> Ir Links

	root file	datasetNumber	logicalDatasetName	projectName	principalPhysicsGroup	dataType	detectorLayout
1	[☐]	003002	dc2.003002.evgen.A2_z_mumu	dc2	test	EVGEN	dc2
2	[☐]	003002	dc2.003002.evgen.A2_z_mumu	dc2	test	EVGEN	dc2
3	[☐]	003002	dc2.003002.evgen.A2_z_mumu	dc2	test	EVGEN	dc2
4	[☐]	003002	dc2.003002.evgen.A2_z_mumu	dc2	test	EVGEN	dc2
5	[☐]	003002	dc2.003002.evgen.A2_z_mumu	dc2	test	EVGEN	dc2
6	[☐]	003002	dc2.003002.evgen.A2_z_mumu	dc2	test	EVGEN	dc2
7	[☐]	003002	dc2.003002.evgen.A2_z_mumu	dc2	test	EVGEN	dc2
8	[☐]	003002	dc2.003002.evgen.A2_z_mumu	dc2	test	EVGEN	dc2
9	[☐]	003002	dc2.003002.evgen.A2_z_mumu	dc2	test	EVGEN	dc2
10	[☐]	003002	dc2.003002.evgen.A2_z_mumu	dc2	test	EVGEN	dc2
11	[☐]	003002	dc2.003002.evgen.A2_z_mumu	dc2	test	EVGEN	dc2
12	[☐]	003002	dc2.003002.evgen.A2_z_mumu	dc2	test	EVGEN	dc2
13	[☐]	003002	dc2.003002.evgen.A2_z_mumu	dc2	test	EVGEN	dc2
14	[☐]	003002	dc2.003002.evgen.A2_z_mumu	dc2	test	EVGEN	dc2
15	[☐]	003002	dc2.003002.evgen.A2_z_mumu	dc2	test	EVGEN	dc2
16	[☐]	003002	dc2.003002.evgen.A2_z_mumu	dc2	test	EVGEN	dc2

Concluído Internet

Figura 42 – Página com os resultados da pesquisa

Nesta mesma página, o usuário deve selecionar o método de transferência (HTTP, Wget, GridFTP ou don quijote), se quer o arquivo comprimido ou não e se deseja fazer o download no momento ou mais tarde. Isto é apresentado na figura 43.

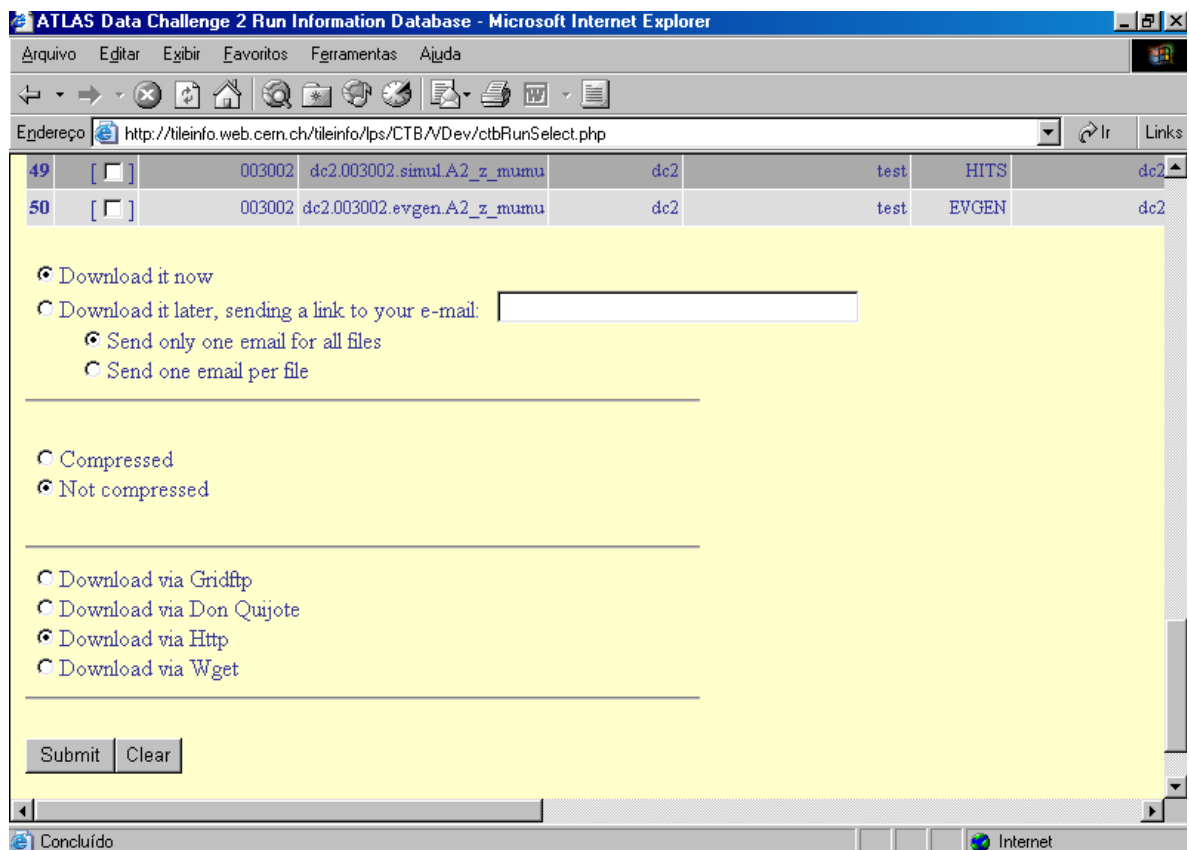


Figura 43 – Na página de resultados, o usuário deve selecionar suas opções

O sistema executa o processamento com as informações obtidas e fornece um link, no caso de transferência HTTP, ou script nos demais casos. A figura 44 mostra o resultado da opção HTTP e a figura 45 apresenta o resultado da opção Wget.

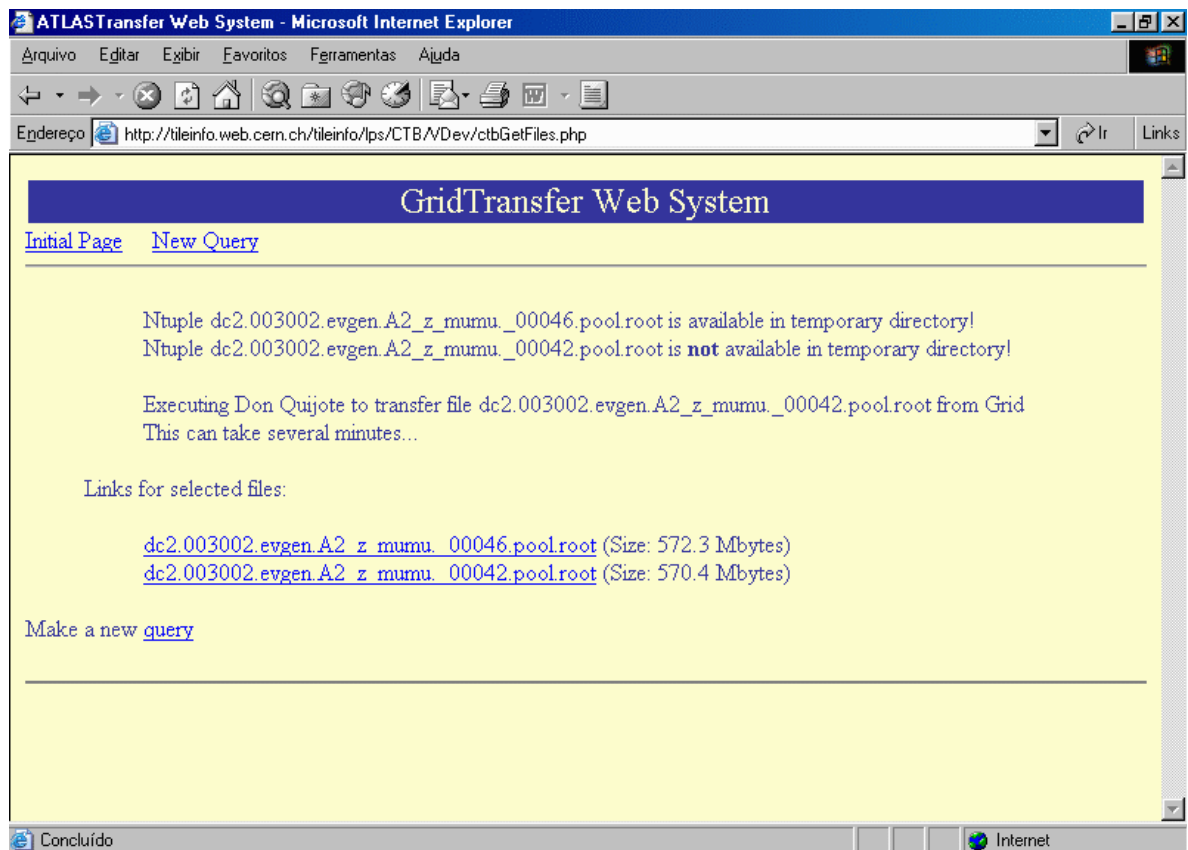


Figura 44 – Transferência HTTP, o sistema informa ao usuário as etapas do processamento e no final apresenta o link

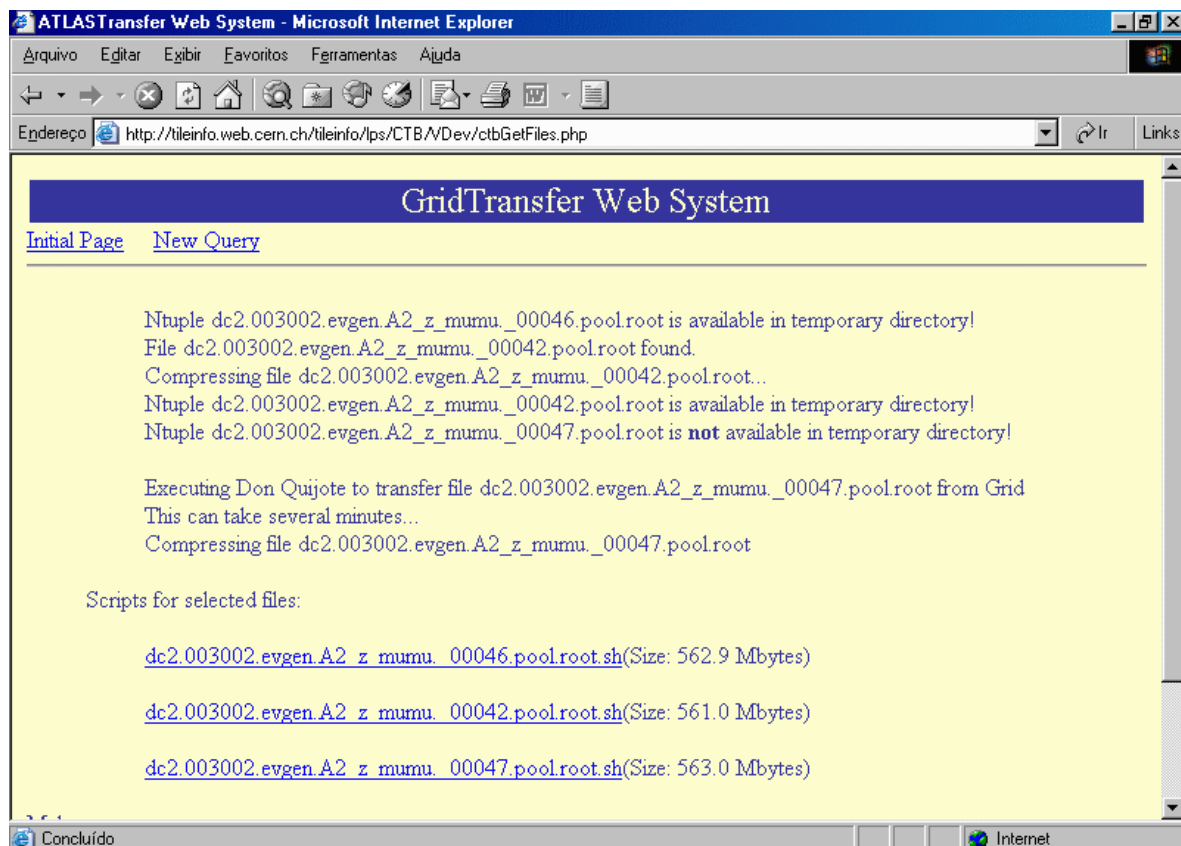


Figura 45 – Transferência Wget, o sistema informa ao usuário as etapas do processamento e no final apresenta o script

O sistema também envia os resultados por e-mail, como apresentado na figura 46, no exato momento em que o usuário recebe o e-mail do sistema. O usuário poderia fechar a janela do navegador e continuar seu trabalho, aguardando o recebimento dos links e scripts por e-mail.

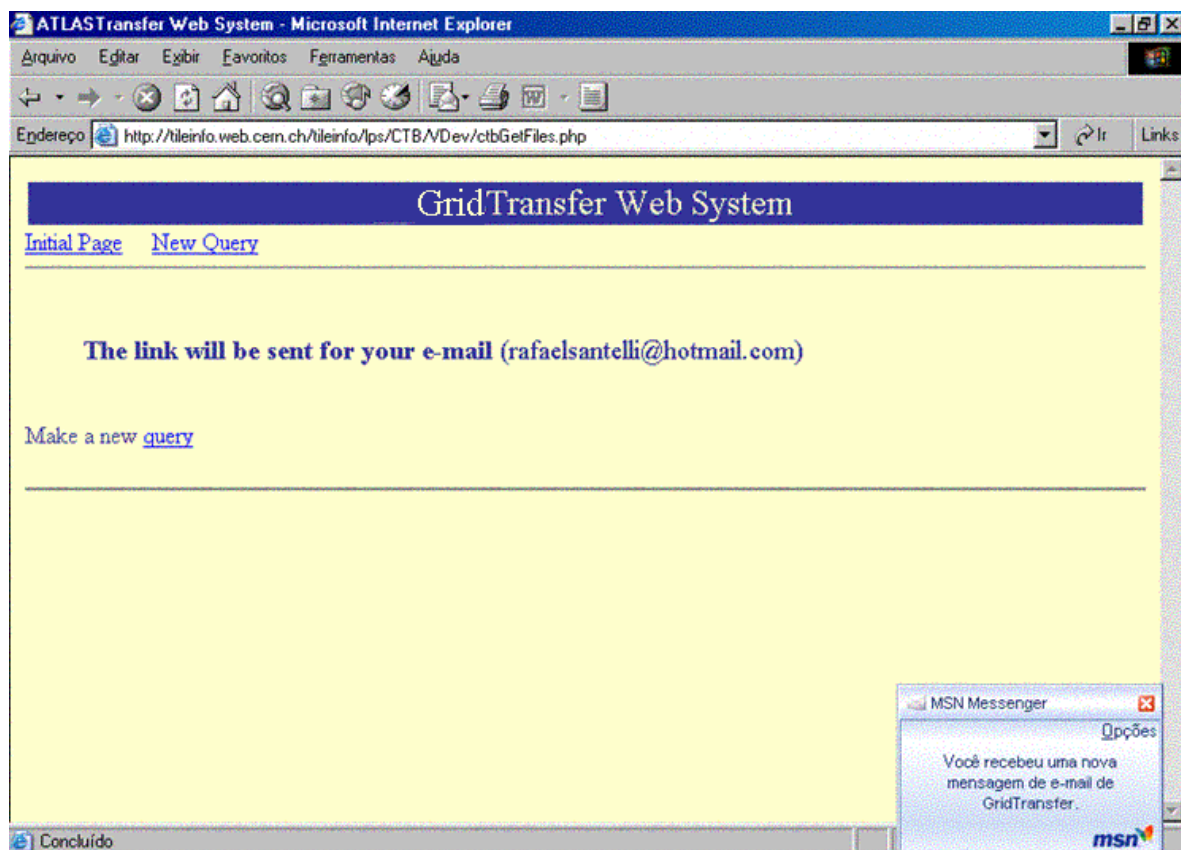


Figura 46 – Na página de resultados, o usuário deve selecionar suas opções

Se o usuário selecionar GridFTP, o sistema executa uma busca usando o Don Quijote e apresenta as Surls (nomes físicos) na página (figura 47) para o usuário escolher uma delas. O sistema, então, monta o script com o comando globus-url-copy acrescido da seleção do usuário.

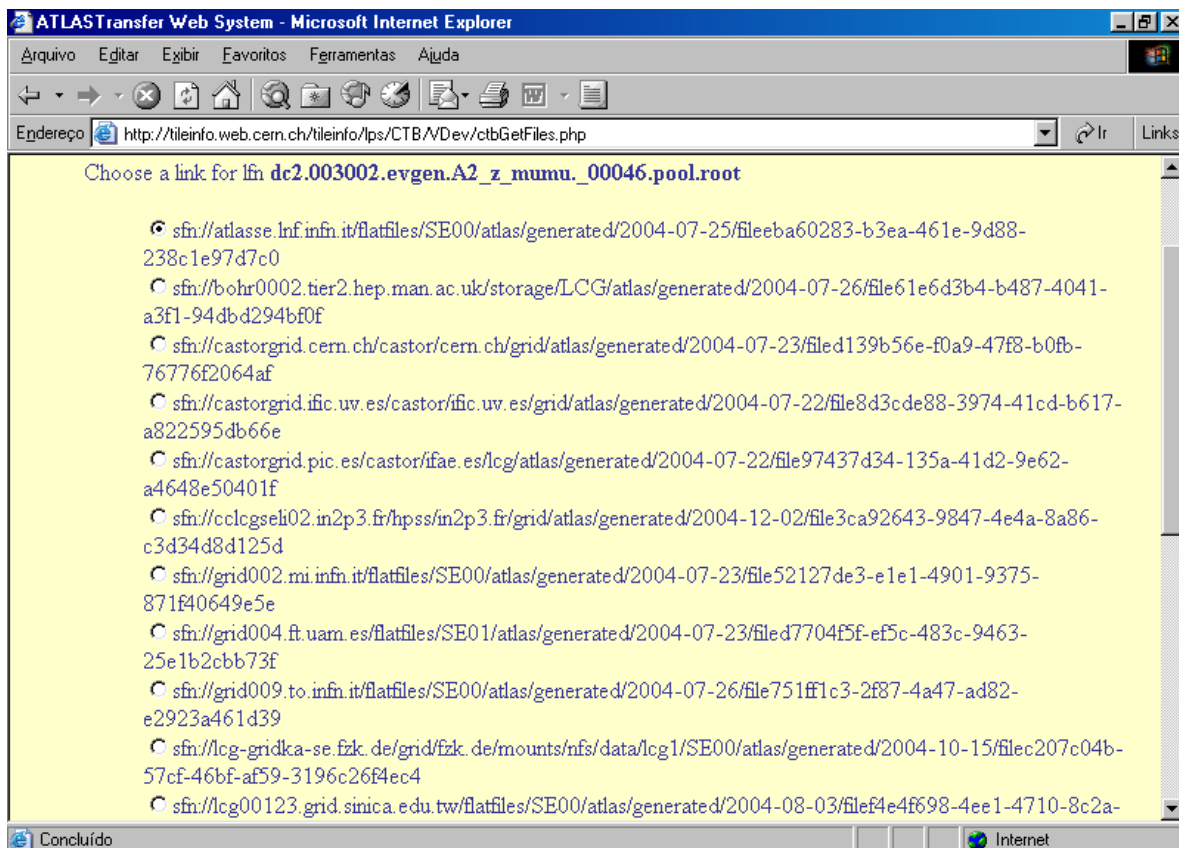


Figura 47 – O sistema mostra os endereços físicos para que o usuário selecione

O usuário executa o script no Shell localmente para fazer a transferência. A figura 48 apresenta a execução do script “dc2.003002.evgen.A2_z_mumu._00046.pool.root” para transferência GridFTP na máquina sion.lps.ufrj.br, localizada no laboratório LPS da UFRJ. O script verifica o espaço em disco, executa o comando de autenticação na Grid (Grid-proxy-init). O usuário insere a senha da chave privada e a transferência GridFTP é feita através do cliente globus-url-copy.

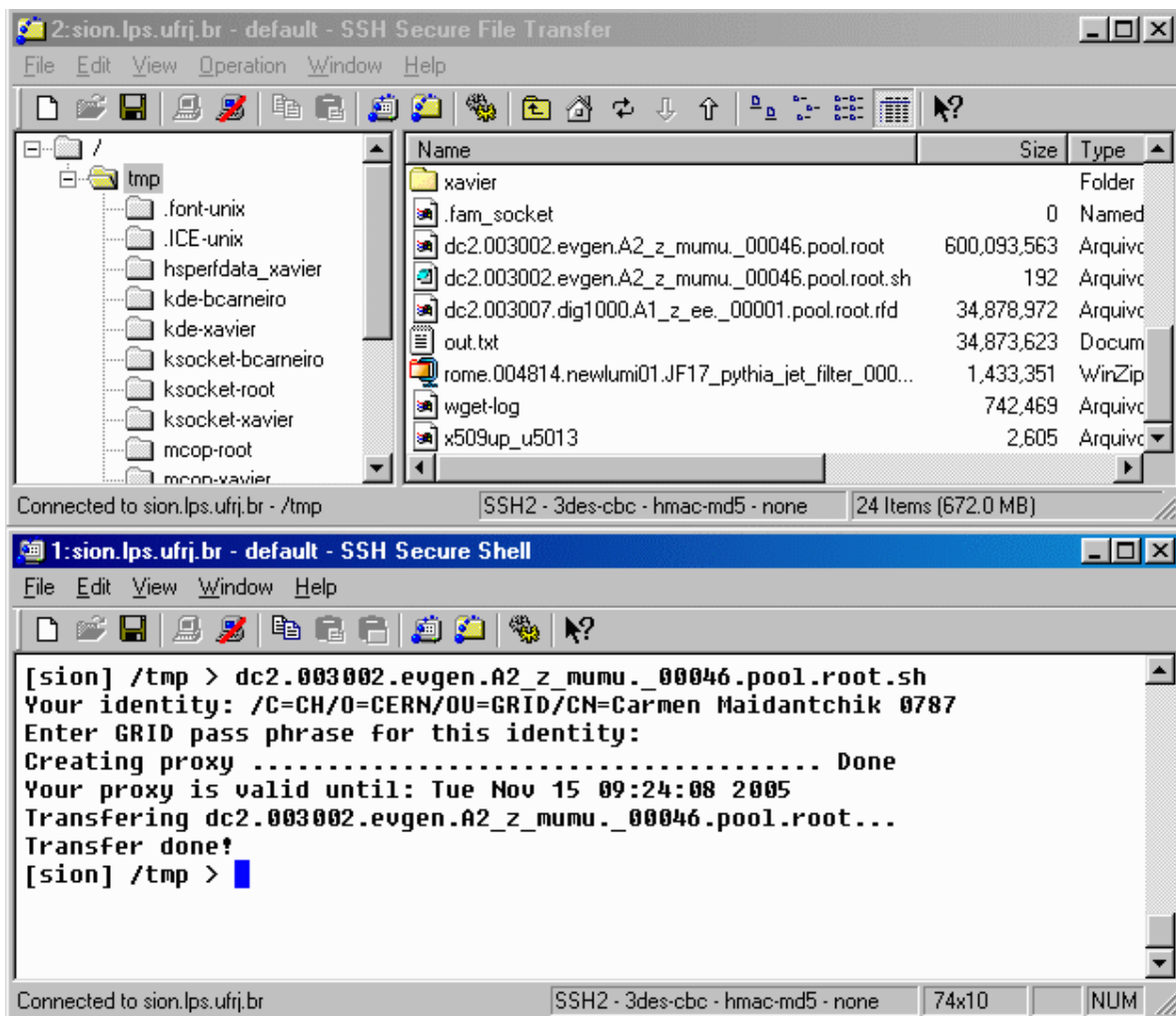


Figura 48 – Execução do script GridFTP na sion.lps.ufrj.br

O usuário também pode transferir executando o script Wget, apresentado na figura 49. Nota-se que a transferência foi interrompida propositalmente e depois retomada para verificar a funcionalidade de recuperar arquivos parciais do Wget.

```
[sion] /tmp > dc2\[1\].003002.evgen.A2_z_mumu._00046.pool.root.sh
--22:00:21-- http://tileinfo.web.cern.ch/tileinfo/tiletransfer/dc2/dc2.003002.e
vgen.A2_z_mumu._00046.pool.root
=> `dc2.003002.evgen.A2_z_mumu._00046.pool.root'
Resolving tileinfo.web.cern.ch... 137.138.77.184
Connecting to tileinfo.web.cern.ch[137.138.77.184]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 600,093,563 [text/plain]

26% [=====>] 160,430,352 11.12K/s ETA 39:12
[sion] /tmp > dc2\[1\].003002.evgen.A2_z_mumu._00046.pool.root.sh
--22:14:46-- http://tileinfo.web.cern.ch/tileinfo/tiletransfer/dc2/dc2.003002.e
vgen.A2_z_mumu._00046.pool.root
=> `dc2.003002.evgen.A2_z_mumu._00046.pool.root'
Resolving tileinfo.web.cern.ch... 137.138.77.184
Connecting to tileinfo.web.cern.ch[137.138.77.184]:80... connected.
HTTP request sent, awaiting response... 206 Partial Content
Length: 600,093,563 (439,663,211 to go) [text/plain]

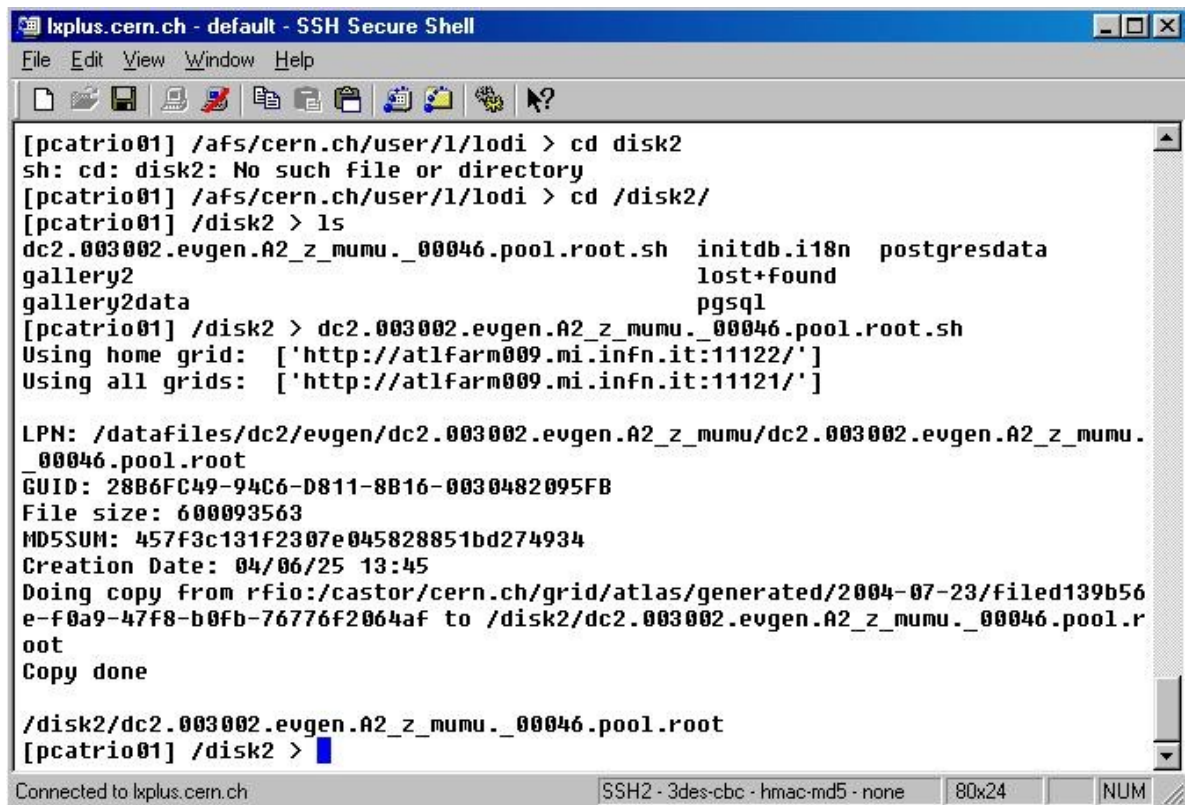
100%[++++++=>] 600,093,563 114.22K/s ETA 00:00

22:51:12 (196.48 KB/s) - `dc2.003002.evgen.A2_z_mumu._00046.pool.root' saved [60
0,093,563/600,093,563]

[sion] /tmp >
```

Figura 49 – Execução do script Wget na sion.lps.ufrj.br

Da mesma forma, é possível transferir arquivos executando o script com o Don Quijote. A figura 50 apresenta a execução do script “dc2.003002.evgen.A2_z_mumu._00046.pool.root” na máquina pcatrio01.cern.ch, onde o arquivo é transferido utilizando o Don Quijote.



```
[pcatrio01] /afs/cern.ch/user/l/lodi > cd disk2
sh: cd: disk2: No such file or directory
[pcatrio01] /afs/cern.ch/user/l/lodi > cd /disk2/
[pcatrio01] /disk2 > ls
dc2.003002.evgen.A2_z_mumu._00046.pool.root.sh  initdb.i18n  postgresdata
gallery2                                         lost+found
gallery2data                                    pgsql
[pcatrio01] /disk2 > dc2.003002.evgen.A2_z_mumu._00046.pool.root.sh
Using home grid: ['http://atlfarm009.mi.infn.it:11122/']
Using all grids: ['http://atlfarm009.mi.infn.it:11121/']

LPN: /datafiles/dc2/evgen/dc2.003002.evgen.A2_z_mumu/dc2.003002.evgen.A2_z_mumu._
_00046.pool.root
GUID: 28B6FC49-94C6-D811-8B16-0030482095FB
File size: 600093563
MD5SUM: 457F3c131f2307e045828851bd274934
Creation Date: 04/06/25 13:45
Doing copy from rfio:/castor/cern.ch/grid/atlas/generated/2004-07-23/Filed139b56
e-f0a9-47f8-b0fb-76776f2064af to /disk2/dc2.003002.evgen.A2_z_mumu._00046.pool.r
oot
Copy done

/disk2/dc2.003002.evgen.A2_z_mumu._00046.pool.root
[pcatrio01] /disk2 > 
```

Figura 50 – Execução do script Don Quijote na pcatrio01.cern.ch

8 Conclusões

Através de uma interface amigável, de fácil acesso e independente de plataforma o sistema GridTransfer proporciona a transferência dos arquivos de dados adquiridos nos arranjos experimentais realizados no detector ATLAS, armazenados em Grid, oferecendo mecanismos de busca, acesso e seleção, independente da localização do arquivo, além de opções alternativas de métodos de transferência mais eficientes que o HTTP, como o Wget, GridFTP e Don Quijote. O usuário pode também escolher entre receber os resultados na página ou por e-mail evitando o tempo de processamento no navegador.

Foram utilizados os experimentos do Data Challenge, os primeiros armazenados em Grid, para validar o sistema. Porém, quando os detectores do LHC entrarem em operação o volume de dados estipulado será de aproximadamente 8 PetaBytes por ano, o que torna inviável o armazenamento no CASTOR. Sendo assim, o modelo computacional testado será colocado em prática e o sistema GridTransfer estará apto a apoiar tais experimentos

O desenvolvimento do sistema integrou, de forma eficaz, as tecnologias e ferramentas já implementadas nas versões anteriores com as estudadas durante o período de pesquisa do ambiente de Grids computacionais a fim de proporcionar o acesso aos arquivos armazenados nas Grids do ambiente CERN. Além disso, foram aplicados conhecimentos adquiridos durante o curso de graduação tais como engenharia de software, computação, sistemas operacionais, banco de dados, tcp/ip, linguagens de programação entre outros.

O sistema contribui para a melhoria na transferência de dados, além de permitir que os pesquisadores se concentrem apenas em suas análises em vez de estudar formas de obter os arquivos de dados, que podem ser transferidos utilizando o sistema.

Como projeto final, trabalhar no desenvolvimento do sistema GridTransfer foi uma experiência enriquecedora, não só pelo conhecimento técnico adquirido, mas também por exercitar a capacidade de identificar um problema, estudar o ambiente e apresentar soluções. Além disso, houve a oportunidade de adquirir conhecimento em outras áreas, já que o projeto está ligado à área de física de altas energias. Deve-se destacar que o fato de o projeto estar relacionado à uma colaboração internacional e o contato com profissionais altamente qualificados valorizaram o projeto. Além disso, tive a oportunidade de participar de jornadas de iniciação científica, apresentando a versão TileTransfer e da elaboração de artigos.

8.1 Trabalhos Futuros

O sistema GridTransfer transfere os dados resultantes dos testes realizados no detector ATLAS. O sistema, inicialmente, transferia dados resultantes dos testes de colisão de partículas no calorímetro de telhas, sendo denominado TileTransfer. Depois, os testes passaram a ser realizados em outros sub-detectores como o calorímetro de Argônio Líquido (“Lar Calorimeter”). Posteriormente, foram realizados testes em conjunto reunindo os sub-detectores. O sistema passou então a ser denominado CombinedTransfer. Ao propiciar a transferência de dados

armazenados em uma Grid computacional, o sistema passou a se chamar GridTransfer. A próxima etapa será integrar as últimas versões do CombinedTransfer e do GridTransfer em um único sistema, que será denominado AtlasTransfer. Outro trabalho a ser desempenhado é o apoio ao comissionamento, a etapa onde serão realizados testes com os sub-detectores montados no detector ATLAS, já na caverna onde o mesmo será posto para a operação.

Algumas novas funcionalidades poderiam ser agregadas futuramente. Uma vez inserido na Grid, o sistema GridTransfer poderia proporcionar, da mesma forma que o portal Genius 10, que os usuários submetessem jobs para o processamento dos dados adquiridos durante a execução do sistema. O sistema pode oferecer a opção de transferência do arquivo, como é realizado atualmente, ou processá-lo em algum recurso da Grid apresentando o resultado final para análise que pode ser apresentado no navegador ou enviado para algum recurso de armazenamento na Grid, ou mesmo para a máquina do usuário.

Ao verificar a complexidade dos ambientes Grid, uma melhoria seria a proposta do projeto Grenade 10 de tornar mais amigável o uso da Grid através de aplicativos instalados no ambiente de trabalho, como um plug-in para que o navegador possa transferir os arquivos usando GridFTP.

Um fato importante é apoiar o processo de análise do pesquisador, conforme mostrado no fluxo inserido na proposta do sistema GridTransfer (Seção 7.1). Proporcionar ferramentas de análise remota dos dados processados dos experimentos do detector ATLAS também é importante. Outra funcionalidade que poderia ser implementada seria um sistema amigável para replicar arquivos para outros institutos de acordo com a necessidade de usuários.

A questão da segurança também será observada nos trabalhos futuros. Um tópico importante é encriptar o conteúdo dos e-mails para que não seja possível que algum intruso leia o conteúdo com os links e scripts.

9 Glossário

AMI	-	ATLAS Metadata Interface
ATLAS	-	A Toroidal LHC ApparatuS
CA	-	Certification Authority (Grid)
CASTOR	-	CERN Advanced STORAge Manager
CERN	-	European Laboratory for Particle Physics
CTB	-	Combined Test Beam
DC	-	Data Challenge
DIRAC	-	Distributed Infrastructure with Remote Agent Control
DQ	-	Don Quixote, distributed data manager (DataManagement/Grid)
EDG	-	European Data Grid project
GENIUS	-	Grid Enabled web eNvironment for site Independent User Job
Submission		
GLOBUS	-	Grid Middleware
Grid	-	Grade computacional
Grid3	-	US Grid System
GridFTP	-	Protocol Extensions to FTP for the Grid
GSI	-	Grid Security Infrastructure
GUID	-	Globally Unique Identifier (Grid)
GILDA	-	Grid Infn Laboratory Dissemination Activities
HBOOK	-	Legacy histogramming package
JDL	-	Job Description Language
Lar	-	Liquid Argon
LCG	-	LHC Computing Grid
LCN	-	Logical Collection Name
LFN	-	Logical File Name
LHC	-	Large Hadron Collider
LHCB	-	Large Hadron Collider Beauty
LPN	-	Logical Path Name
Ntuples	-	Arquivos de dados adquiridos nos testes de partículas
RA	-	Registration Authority
ROOT	-	Biblioteca para análise de dados
SE	-	Storage Element (Grid)
SURL	-	Storage URL
TCP	-	Transmission Control Protocol
TILECAL	-	Calorímetro de telhas
UI	-	User Interface
VO	-	Virtual Organization (Grid)
XML	-	Extensible Markup Language
XSL	-	Extensible Stylesheet Language

Parâmetros da tabela Tilecal:

TYPE - Tipo da run (de fisica, de pulsos de calibracao, de pedestal)

ENERGY : Energia do feixe de partícula
BEAM : Tipo de partícula (electron, píons, muons)
DATE: Data da experiência
SPILLS : Número de “spills”, emissões do feixe.
EVENTS: Número de eventos depois da filtragem
LAS : Parâmetros dos eventos de laser
MOD: Módulo que foi utilizado na experiência
THETA : Ângulo com relação a transversal do feixe do ATLAS.
PHY : Número de eventos do tipo física
CELL: Tipo de célula que o feixe está apontando
Z: Posição z
CH1/CH2: Pressão cherenkov
TB: Período do testbeam (teste de feixe)

10 Referências Bibliográficas

Esta seção contém as referências de livros e sites de Internet utilizados durante a pesquisa para elaboração do relatório. Os sites foram acessados no mês de Novembro de 2005.

- [1] Física de partículas – CERN
<http://public.web.cern.ch/Public/>
- [2] Site do ATLAS
<http://www.cern.ch/atlas>
- [3] Site do calorímetro de telhas (Tilecal)
http://atlas.web.cern.ch/Atlas/SUB_DETECTORS/TILE/tilecal.HTML
- [4] Visão do calorímetro
http://atlas.web.cern.ch/Atlas/SUB_DETECTORS/TILE/TDR/HTML/TDR-4.HTML#HEADING4-0
- [5] Princípio de funcionamento do Tilecal
http://atlas.web.cern.ch/Atlas/SUB_DETECTORS/TILE/TDR/HTML/TDR-8.HTML#HEADING8-0
- [6] Site do CERN com informações sobre como acessar as ntuples
HTTP://atlas.web.CERN.ch/Atlas/SUB_DETECTORS/TILE/testbeam/tb2002/get_ntuples.HTML
- [7] Site com informações sobre como utilizar o servidor CASTOR
<HTTP://it-div-ds.web.CERN.ch/it-div-ds/HSM/CASTOR/>
- [8] Manual do PHP
<HTTP://www.php.net>
- [9] Documentação do BBFTP
<HTTP://doc.in2p3.fr/BBFTP/doc.3.0.0.HTML#changes>
- [10] Software Wget
HTTP://www.gnu.org/manual/Wget/HTML_mono/Wget.HTML#SEC1
- [11] CASTOR
<HTTP://castor.web.CERN.ch/castor/>
- [12] Site do AMI
<HTTP://isnpx1158.in2p3.fr:8180/AMI/>

- [13] General Introduction to AMI:
[HTTP://isnpx1158.in2p3.fr:8180/AMI/servlet/net.hep.atlas.Database.Bookkeeping.AMI.Servlet.Help?Project=no&Process=no&Entity=no](http://isnpx1158.in2p3.fr:8180/AMI/servlet/net.hep.atlas.Database.Bookkeeping.AMI.Servlet.Help?Project=no&Process=no&Entity=no)
- [14] ATLAS Computing:
[HTTP://atlas-proj-computing-tdr.web.CERN.ch/atlas-proj-computing-tdr/PDF/Computing-TDR-draft-2.1.pdf](http://atlas-proj-computing-tdr.web.CERN.ch/atlas-proj-computing-tdr/PDF/Computing-TDR-draft-2.1.pdf)
- [15] Como obter um certificado:
[HTTP://service-Grid-ca.web.CERN.ch/service-Grid-ca/help/user_req.HTML](http://service-Grid-ca.web.CERN.ch/service-Grid-ca/help/user_req.HTML)
- [16] Projeto Grenade:
[HTTP://www.sve.man.ac.uk/Research/AtoZ/GRENADE](http://www.sve.man.ac.uk/Research/AtoZ/GRENADE)
- [17] Projeto Globus
[HTTP://www.globus.org/](http://www.globus.org/)
- [18] Delegação de proxy:
[HTTP://www-unix.globus.org/toolkit/docs/development/3.9.4/security/myproxy/user/#introduction](http://www-unix.globus.org/toolkit/docs/development/3.9.4/security/myproxy/user/#introduction)
- [19] Portal Gilda:
[HTTPs://gilda.ct.infn.it/](http://gilda.ct.infn.it/)
- [20] Portal Genius:
[HTTPs://genius.ct.infn.it/](http://genius.ct.infn.it/)
- [21] LCG User Guide:
[HTTP://Grid-deployment.web.CERN.ch/Grid-deployment/eis/docs/LCG-1-UserGuide.htm](http://Grid-deployment.web.CERN.ch/Grid-deployment/eis/docs/LCG-1-UserGuide.htm)
- [22] LHC Computing Grid Project
[HTTP://lcg.web.CERN.ch/LCG/](http://lcg.web.CERN.ch/LCG/)
- [23] Requirements for LCG User Registration and VO Membership Management
[HTTPs://edms.CERN.ch/file/428034//LCG User Registration.pdf](https://edms.CERN.ch/file/428034//LCG%20User%20Registration.pdf)
- [24] LCG User Developer Guide
[HTTP://Grid-deployment.web.CERN.ch/Grid-deployment/cgi-bin/index.cgi?var=eis/docs](http://Grid-deployment.web.CERN.ch/Grid-deployment/cgi-bin/index.cgi?var=eis/docs)
- [25] European DataGrid Project
[HTTP://eu-dataGrid.web.CERN.ch/eu-dataGrid/](http://eu-dataGrid.web.CERN.ch/eu-dataGrid/)
- [26] Overview of the Grid Security Infrastructure
[HTTP://www-unix.globus.org/security/overview.HTML](http://www-unix.globus.org/security/overview.HTML)
- [27] EDG Tutorial – Handout for Participants for EDG Release 2.x

[HTTP://edms.CERN.ch/document/393671](http://edms.CERN.ch/document/393671)

[28] Instalação do GridFTP

[HTTP://it-dep-fio-ds.web.CERN.ch/it-dep-fio-ds/Documentation/GridFTP-client.asp](http://it-dep-fio-ds.web.CERN.ch/it-dep-fio-ds/Documentation/GridFTP-client.asp)

[29] Instalação e uso do Don Quijote

[HTTPs://uimon.CERN.ch/twiki/bin/view/Atlas/DonQuijoteDms3](http://uimon.CERN.ch/twiki/bin/view/Atlas/DonQuijoteDms3)

[30] Site para registro na Grid LCG

[HTTP://lcg-registrar.CERN.ch/](http://lcg-registrar.CERN.ch/)

[31] Página do Xalan

[HTTP://xml.apache.org/xalan-j](http://xml.apache.org/xalan-j)

[32] ATLAS Data Challenge

<http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/DC/>

[33] Artigo: The anatomy of the Grid

<http://esc.dl.ac.uk/StarterKit/PDF/anatomy.pdf>

[34] Publicação Kaio Karam

<http://www.lps.ufrj.br/~kaiokaram/>

[35] GridFTP

<http://www.globus.org/toolkit/data/gridftp/>

[36] Site XSL

<http://www.w3.org/Style/XSL/>

[37] C Completo e Total, Herbert Schildt, Terceira Edição, ed. Makron Books

[38] Site do Dirac

<http://dirac.cern.ch>

[39] Publicação Lanza

<http://www.del.ufrj.br/~lanza>

11 Anexo A

Esta seção contém o script referido na seção 7.2.

```
VirtualOrganisation="lhcb";
Executable = "vamosdspi101.sh";
StdOutput = "std101.out";
StdError = "std101.out";
InputSandbox = {"vamosdspi101.sh","comum.tar","dspi.tar"};
InputData = {"lfn:kernlib", "lfn:packlib", "lfn:mathlib",
"lfn:/CERN/pro/lib/libpythia6205.a", "lfn:/CERN/pro/lib/libstdhep.a",
"lfn:/CERN/pro/lib/libpdflib804.a", "lfn:qqlib"};
DataAccessProtocol = {"gsiftp"};
OutputSandbox = {"std101.out","pythia.101.log","pythia_nt.101.hbook"};
Requirements = (other.GlueCEPolicyMaxWallClockTime > 2000 &&
other.GlueCEUniqueID != "Gridba2.ba.infn.it:2119/jobmanager-lcgpbs-long" &&
other.GlueCEUniqueID != "ce101.Grid.ucy.ac.cy:2119/jobmanager-torque-lhcb"
&& other.GlueCEUniqueID !=
"Gridba2.ba.infn.it:2119/jobmanager-lcgpbs-infinite");

#!/bin/csh -f
set MYID = $1
set MYDEC = $2
set NEVT = $3
set SEED = $4
set WD = 'pwd'
/bin/rm vamosdspi$MYID.sh
echo 'tar -xzf comum.tar' > vamosdspi$MYID.sh
echo 'mkdir job'$MYID >> vamosdspi$MYID.sh
echo 'mv DECAY.DEC job'$MYID >> vamosdspi$MYID.sh
echo 'tar -xzf libraries.tar' >> vamosdspi$MYID.sh
#echo 'rm libraries.tar' >> vamosdspi$MYID.sh
echo 'tar -xzf dspi.tar' >> vamosdspi$MYID.sh
echo 'mv BS02DS-_K-K+PI-_PI+.dec job'$MYID >> vamosdspi$MYID.sh
echo 'lcg-cp --vo lhcb lfn:kernlib file:$PWD/lib/libkernlib.a' >> vamosdspi$MYID.sh
echo 'lcg-cp --vo lhcb lfn:mathlib file:$PWD/lib/libmathlib.a' >> vamosdspi$MYID.sh
echo 'lcg-cp --vo lhcb lfn:packlib file:$PWD/lib/libpacklib.a' >> vamosdspi$MYID.sh
echo 'lcg-cp --vo lhcb lfn:/CERN/pro/lib/libpythia6205.a file:$PWD/lib/libpythia6205.a' >>
vamosdspi$MYID.sh
echo 'lcg-cp --vo lhcb lfn:/CERN/pro/lib/libpdflib804.a file:$PWD/lib/libpdflib804.a' >>
vamosdspi$MYID.sh
echo 'lcg-cp --vo lhcb lfn:/CERN/pro/lib/libstdhep.a file:$PWD/lib/libstdhep.a' >>
vamosdspi$MYID.sh
echo 'ls lib' >> vamosdspi$MYID.sh
echo 'csh make.sh '$MYID >> vamosdspi$MYID.sh
```



```

echo '/bin/rm $PWD/lib/*.a' >> vamosdspis$MYID.sh
echo '/bin/rm *.f' >> vamosdspis$MYID.sh
echo '/bin/rm *.F' >> vamosdspis$MYID.sh
echo '/bin/rm ntup.inc' >> vamosdspis$MYID.sh
echo '/bin/rm make.sh' >> vamosdspis$MYID.sh
echo '/bin/rm *.tar' >> vamosdspis$MYID.sh
#echo 'cp pythia.exe pythia'$MYID'.exe' >> vamosdspis$MYID.sh
echo 'mv subdspis.sh subdspis'$MYID'.sh' >> vamosdspis$MYID.sh
echo 'csh subdspis'$MYID'.sh' $MYID $MYDEC $NEVT $SEED >> vamosdspis$MYID.sh
echo '/bin/rm subdspis'$MYID'.sh' >> vamosdspis$MYID.sh
/bin/rm dspis$MYID.jdl
set LOGFILE = pythia.$MYID.log
set EXEFILE = vamosdspis$MYID.sh
set NTFILE = pythia_nt.$MYID.hbook
echo 'VirtualOrganisation="lhcb";' > dspis$MYID.jdl
echo 'Executable = "$EXEFILE";' >> dspis$MYID.jdl
echo 'StdOutput = "std'$MYID'.out";' >> dspis$MYID.jdl
echo 'StdError = "std'$MYID'.out";' >> dspis$MYID.jdl
echo 'InputSandbox = {"$EXEFILE","comum.tar","dspis.tar"};' >> dspis$MYID.jdl
echo      'InputData      =      {"lfn:kernlib",      "lfn:packlib",      "lfn:mathlib",
"lfn:/CERN/pro/lib/libpythia6205.a",      "lfn:/CERN/pro/lib/libstdhep.a",
"lfn:/CERN/pro/lib/libpdfplib804.a", "lfn:qqlib"};' >> dspis$MYID.jdl
echo 'DataAccessProtocol = {"gsiftp"};' >> dspis$MYID.jdl
echo      'OutputSandbox      =      {"std'$MYID'.out',"$LOGFILE","$NTFILE"};' >>
dspis$MYID.jdl
echo      'Requirements      =      (other.GlueCEPolicyMaxWallClockTime      >      2000      &&
other.GlueCEUniqueID      !=      "Gridba2.ba.infn.it:2119/jobmanager-lcgpbs-long"      &&
other.GlueCEUniqueID      !=      "ce101.Grid.ucy.ac.cy:2119/jobmanager-torque-lhcb"      &&
other.GlueCEUniqueID      !=      "Gridba2.ba.infn.it:2119/jobmanager-lcgpbs-infinite");' >>
dspis$MYID.jdl
set JDLFILE = dspis$MYID.jdl
# csh vamosdspis$MYID.sh
# edg-job-submit -r lcg06.sinp.msu.ru:2119/jobmanager-lcgpbs-long -o dspis1.txt $JDLFILE
edg-job-submit -o dspis1.txt $JDLFILE
rm vamosdspis$MYID.sh
%EOF

```

12 Anexo B

Esta seção contém o script referido na seção 7.2.

```
#!/bin/csh -f
#####
# Initialize LSF queue parameters
#
# "-q" gives the queue (1nh by default)
#
# -q 8nm    express
# -q 1nh    short
# -q 8nh    medium
# -q 1nd    long
# -q 1nw    verylong
#
#
# define the job name
#BSUB -J myjob
# define the queue
#BSUB -q 1nw
#
#####
#
# in BATCH output files are written in the BATCH machine
# and copy at the end to your $MYWORKDIR output directory
# in INTERACTIVE output files are written onto $WORKDIR
# you have to set $WORKDIR to what you want
#
if ($#argv >= 1 ) then
    echo " Usage: DaVinci.job [<mode>]"
    echo " where:"
    echo " <mode>      (optional) is execution mode (e.g. $CMTDEB)"
    if (" $1 " != " $CMTDEB ") then
        exit 0
    endif
endif
#
set MODE = $CMTCONFIG
if ($?LS_SUBCWD) then
# in BATCH go to the BATCH local directory
    cd $WORKDIR
else
# in interactive set WORKDIR to your output directory
    setenv WORKDIR $HOME/w1
    if ($#argv == 1) set MODE = " $1 "
    cd `dirname $0`
endif
set current = $PWD
```

```

#
# set location of DaVinci executable
set Dversion = "v11r11"
setenv MYJOBROOT $HOME/newmycmt/Phys/DaVinci/$Dversion

# == set the program environment
set Dversion = "v11r11"
source $DaVinci_release_area/DaVinciEnv/$Dversion/cmt/setup.csh
source $MYJOBROOT/cmt/setup.csh -tag=$MODE

# == set executable name
setenv EXEC DaVinci

# == in batch copy the
executable,=====
#   in interactive create a link to the executable you want to run
=====
#
if ($?LS_SUBCWD) then
  cp $MYJOBROOT/$MODE/$EXEC.exe $current/tmpjob.exe
else
  ln -fs $MYJOBROOT/$MODE/$EXEC.exe $current/tmpjob.exe
endif
#
#== create temporary joboptions file =====
#cp $MYJOBROOT/options/DaVinci.opts $current/tmpjob.opts
cp $LOKIEXAMPLEOPTS/DV_LoKi_Ks2PiPi.opts $current/tmpjob.opts

# == run $EXEC.exe =====
#
if ($MODE == "$CMTDEB") then
  setenv JOBOPTPATH $current/tmpjob.opts
  gvd $current/tmpjob.exe
else
  time $current/tmpjob.exe $current/tmpjob.opts
endif

# == job is finished: if in batch copy output files on user directories
if ($?LS_SUBCWD) then
  ls core* > /dev/null
  if ($status == 0) then
    gdb $current/tmpjob.exe core* << EOF
  where
  quit
  EOF
  rm core*
endif
ls *.* >! RETURN
endif

```

```
# == job is finished: remove temporary files =====  
rm -f tmp*
```

```
exit 0
```

13 Anexo C

Este anexo contém o e-mail referido na seção 7.2

De: Solveig Albrand
Enviada: qua 2/3/2005 15:56
Para: Carmen Maidantchik; Paulo Da Silva
Cc: Miguel De Oliveira Branco
Assunto: A new tool similar to tile transfer?

Dear Carmen and Paulo,

Yesterday was the start of a new series of meetings on ATLAS data management, coordinated by Miguel Branco. You may have seen the slides that Torre Wenaus presented in the SW week plenary session last Friday. We discussed among other things how we could improve the service to users with a better link between AMI, which contains as you know physics and "logical" attributes of data, to Don Quixote(DQ), which is the ATLAS

data management tool, and provides access to the various Grid file replica catalogs.

Currently AMI accesses a Don Quixote web service which tells users where physical replicas of the files can be found, and they must then use a DQ command from their own account to stage the files. This is not really very useful.

One small improvement would be to produce a well formed DQ command from AMI, but during the meeting we decided it would be better to actually perform the command for them, a bit like you did for the CASTOR access in Tile Transfer.

So, we were wondering if you have thought about implementing something similar to your Tile Transfer tool, but which would work over DQ, and be capable of getting files for users and putting them into some kind of cache area. I think it would be quite a similar interface to your current tool, and would not be too complicated below, although DQ is a bit more complex than rfdir etc. You probably would have to implement a security level, using the users certificate proxy.

Anyway if you are interested in this, and if Paulo is still at CERN, probably the best plan is for him to go and have a chat with Miguel.

Best wishes,
Solveig.

14 Anexo D

Esta seção apresenta as partes principais do código desenvolvido na implementação dos módulos que compõem o sistema TileTransfer.

Módulo Busca

O módulo de busca foi desenvolvido na linguagem PHP. Primeiramente, ele recebe a escolha feita pelo usuário:

```
while ( list($key, $val) = each($HTTP_POST_VARS)
```

Depois, se conecta ao banco de dados mysql e faz uma pesquisa na tabela runinfo. O **select** da consulta é montado de acordo com os parâmetros passados pelo usuário:

```
// construct select from where statement
if ( strlen($select)<6 ) {
    $select = "select ";
    if ( count($dsp)>0 ) $select .= implode(",", $dsp);
    else $select .= "*";

    $select .= " from runinfo ";
    if ( count($exp)>0 ) $select .= "where " . implode(" and ", $exp);

    if ( strlen($orderby)>0 ) $select .= " order by " . $orderby;
    if ( strlen($orderdir)>0 && strlen($orderby)>0 ) $select .= " " . $orderdir;

mysql_connect("localhost","tilecal","")
or die("cannot connect to database server :(");

$res = mysql_db_query("tile",$select)
or die("query failed :(");
```

O resultado da consulta é disponibilizado na página através de uma tabela contendo checkboxes para que o usuário possa selecionar experimentos e submeter a requisição:

```
for( $i=$start; $i<=$end; $i++) {
    ...
    echo "<TR ALIGN=right BGCOLOR=$tcol><TD><INPUT TYPE='checkbox' "
NAME='\"ntuple[\" VALUE='\"$field[0] $rdate $field[2]\"></TD>";
    ...
}
```

Módulo Download

O modulo download recebe os parâmetros de escolha do usuário, faz a validação desses dados e faz a chamada do módulo link_página ou link_email passando os parâmetros de escolha do usuário.

```
if ($download == "later")          //link_email
{

    passthru ("./tiletransfer_sendForMail $ip $run[$index] $monthYear[$index] $type[$index]
$email $online $compress > /dev/null", $retCode);

    ...

else          //link_página
{
    print "<br>Searching for ntuples in scratch directory ... <br>\n";
}
```

O módulo de download será explicado nas próximas seções já que ele utiliza os módulos validação, link_página, link_email e executa_comandos.

Módulo Validação

O módulo de validação verifica se o usuário inseriu corretamente os parâmetros para que o sistema possa buscar e transferir os arquivos. Ele verifica se o usuário escolheu ou não um experimento (se não escolheu envia um alerta ao usuário). Verifica também, quando o usuário escolhe receber o link por e-mail, se o e-mail foi preenchido corretamente já que o e-mail tem que conter “@” e “.”. É importante ressaltar que o módulo validação foi feito em Javascript, linguagem apropriada para executar código na máquina do cliente, evitando processamento no servidor.

```
<SCRIPT type="text/javascript">
function testejava()
{
    var flagjava=false;
    var indice;
    var i;
    if(document.Myform.download[0].checked)
    {
        validity =
/^\\w+((\\-\\w+)|(\\.\\w+))*\\@[A-Za-z0-9]+((\\.|\\-)[A-Za-z0-9]+)*\\.\\[A-Za-z0-9\\]+$;/
        if (document.Myform.email.value.search(validity) == -1)
        {
            report = 'The e-mail address you entered is invalid. ';
            report += 'Please use the format someone@somewhere.domain.';
            alert(report);
            document.Myform.email.value = "";
            document.Myform.email.focus();
            return false
        }
    }
}
```

```

    }

    indice=document.Myform.length;
    indice=indice-5;

    for(i=0; i<indice; ++i)
    {
        if(document.Myform.elements[i].checked)
            flagjava=true;
    }
    if(!flagjava)
    {
        alert("Please choose one ntuple.");
        return false;
    }
    else
        //document.Myform.submit();
        return true;
    }
    else
    {
        indice=document.Myform.length;
        indice=indice-5;

        for(i=0; i<indice; ++i)
        {
            if(document.Myform.elements[i].checked)
                flagjava=true;
        }
        if(!flagjava)
        {
            alert("Please choose one ntuple.");
            return false;
        }
        else
            //document.Myform.submit();
            return true;
    }
}
</SCRIPT>

```

Módulo Link_email

O modulo link_email recebe os parâmetros do módulo download e cria um processo filho, onde será executado o processo de execução de comandos:


```

pid=fork();
if (pid < 0 )
{
    printf("error id=%d\n",pid);
    exit(PROCESS_ERROR);
}

else if (pid)
{
    //printf("parent\n");
    exit(OK);
}

else
{
    //CHILD PROCESS
...

```

O resultado da execução dos comandos é enviado por e-mail ao usuário através da abertura de um novo processo que utiliza o programa sendmail para enviar a mensagem eletrônica.

```
p = popen("/usr/sbin/sendmail -t","w");
```

15 Apêndice

Esta seção apresenta os trabalhos publicados e as participações em apresentações do trabalho realizado.

- Participação na publicação do artigo:

Computação tipo Grid para o calorímetro de telhas do experimento ATLAS do CERN
Local: XXIII Encontro Nacional de Física de Partículas e Campos, realizado em
Águas de Lindóia, SP - 15 a 19 de outubro de 2002

Resumo:

A internet e, principalmente, a Web oferecem um compartilhamento de informações através da rede de computadores. As Grids computacionais permitem a utilização de recursos de forma distribuída. Neste trabalho apresentamos uma solução Grid para acessar as ntuplas resultantes do teste de feixe no Calorímetro de Telhas no experimento ATLAS do CERN. O objetivo do sistema TileTransfer é transferir as ntuplas de forma segura, rápida e eficiente. A equipe desenvolveu uma interface com o sistema de fitas CASTOR que gerencia a base de dados de arquivos e a solicitação do cliente. A transferência do(s) arquivo(s) é realizada através de um sistema Web. O módulo de gerência dos arquivos transferidos permite o acesso de um servidor mais próximo ou daquele de mais rápida transmissão. Esta abordagem permite tanto o compartilhamento de recursos quanto de conhecimentos uma vez que as análises resultantes podem ser publicadas no mesmo ambiente e associadas aos arquivos de dados.

- Participação nas XXIV e XXV Jornadas de iniciação científica
Elaboração de trabalho: "Tiletransfer: Um sistema web para transferência de dados do detector ATLAS do CERN"