

# Portal ICE - Uso de Web Services para Atingir Extensibilidade no Gerenciamento de Múltiplos Agregados\*

Rodrigo Righi, Lucas Schnorr, Clarissa Marquezan,  
Nicolas Maillard, Alexandre Carissimi, Philippe Navaux  
Grupo de Processamento Paralelo e Distribuído, PPGC/UFRGS  
{rrrighi, lmschnorr, clarissa, nicolas, asc, navaux}@inf.ufrgs.br

## Resumo

Arquiteturas para alto desempenho são principalmente representadas por agregados (clusters) e grades. Agregados representam um ambiente controlado e empregam ferramentas específicas para viabilizar as suas funcionalidades. Sistemas para grades apresentam máquinas paralelas como uma única imagem e tratam parâmetros que comumente não são necessários em agregados. Analisando esse contexto, pode-se apontar uma situação entre grades e agregados que é caracterizada pelo acesso isolado de usuários a vários agregados. Ela é marcada pela necessidade desses usuários de aprender detalhes de cada ferramenta dos agregados que tem acesso para poderem utilizá-los. O gerenciamento desse cenário é o alvo do Portal Web ICE - Integrated Cluster Environment. Ele é baseado numa arquitetura orientada a serviço, construída com Web Services, que esconde a complexidade de lidar com diversos agregados. O arcabouço do ICE oferece transparência de acesso e extensibilidade para fácil tratamento de funcionalidades e ferramentas de agregados. Esse artigo descreve o ICE, seu ambiente de testes e trabalhos relacionados.

## 1 Introdução

Mesmo com os avanços tecnológicos e de ferramentas de *software*, pode-se observar que muitas aplicações ainda necessitam de mais desempenho. Para o tratamento dessas aplicações intensivas, são formadas máquinas cada vez mais potentes, tanto em termos de processamento como na parte de rede de interconexão. Nestes ambientes utiliza-se a programação paralela e distribuída para se obter os resultados com uma precisão e em um intervalo de tempo satisfatórios. Em termos de máquinas, as arquiteturas mais comumente usadas para computação de alto desempenho são: agregados (clusters)[4] e grades (grids)[8, 9].

Agregados são formados por computadores conectados com uma rede de interconexão dedicada. Normalmente,

eles se caracterizam por ter recursos homogêneos e operam com ferramentas e sobre políticas bem definidas. Hoje em dia, existem diversos projetos de pesquisa encarregados de cobrir funcionalidades para os agregados. Dentro da funcionalidade de gerenciamento de trabalhos, por exemplo, pode-se citar as ferramentas OpenPBS[2] e SGE[10]. Apesar de almejam o mesmo propósito, cada qual possui suas próprias peculiaridades. Elas podem compreender diferentes parâmetros e configurações, o que dificulta a adaptação dos usuários na utilização de um novo agregado.

Grades são formadas pelo uso coordenado dos recursos, constituindo organizações virtuais (VO)[8]. Explorando essa idéia, uma VO pode englobar computadores distribuídos pela Internet como uma única plataforma. Esse tipo de arquitetura envolve desafios como o tratamento da heterogeneidade de processadores e de rede, segurança, tolerância a falhas e descoberta e alocação de recursos. No âmbito das grades, o usuário muitas vezes não conhece os recursos que está usando. O Kit Globus[8] oferece sistemas que cobrem questões para esse ambiente. Além de ambientes de larga cobertura, pode-se formar grades locais reunindo agregados como uma única imagem, formando multi-agregados. DECK[1] é exemplo de sistema que visa suportar essa visão de grade.

Analisando os ambientes descritos acima, pode-se ainda observar uma modificação de multi-agregados que é quando vários agregados são acessados de maneira isolada e não como uma única imagem. Esse cenário pode ser situado entre agregados e grades e aparece freqüentemente quando um usuário possui acesso a múltiplos agregados. O problema desse cenário é a necessidade por parte dos usuários de lidar com diferentes parâmetros das ferramentas de cada agregado. Nesse campo nota-se uma lacuna que pode ser preenchida com sistemas que padronizam operações sobre ferramentas de uma funcionalidade de agregado.

Pensando no contexto explorado acima, foi desenvolvido o ambiente ICE - *Integrated Clusters Environment* - que representa uma nova abordagem Web para tratar acesso a múltiplos agregados. Ele fornece transparência usando um

\*Projeto JavaWSPAD. Financiado parcialmente pela HP Brasil e Finep

Portal Web para acesso a agregados e extensibilidade para a inclusão de novas ou existentes ferramentas de agregados. Nesse sentido, o ICE visa lidar com ferramentas para vários tipos de funcionalidades, tais como monitoramento, depuração, gerenciamento de trabalhos e configuração de máquinas. A característica chave do ambiente ICE é a extensibilidade. Ela é disponibilizada através de uma Arquitetura Orientada a Serviço (SOA)[9]. Essa arquitetura é atingida através de um arcabouço e da uniformização de interfaces para lidar com diferentes funcionalidades de agregados e suas implementações. No contexto do ICE, SOA contribui para esconder a heterogeneidade das ferramentas de múltiplos agregados e permite a fácil inclusão deles ao ambiente. Esse artigo objetiva descrever o ICE, mostrando seus módulos e plataforma de execução, e faz uma comparação entre ele e trabalhos relacionados.

O presente artigo está organizado da seguinte maneira. A seção 2 descreve o ambiente ICE. A seção 3 apresenta os serviços desenvolvidos. A seção 4 mostra a plataforma de testes para esse ambiente. A seção 5 menciona trabalhos relacionados e o diferencial do ICE. Por fim, a seção 6 ressalta as suas contribuições e descreve trabalhos futuros.

## 2 Ambiente ICE

O ambiente ICE foi idealizado para esconder a complexidade de lidar com diferentes ferramentas para agregados de computadores. Para submeter trabalhos em agregados, além de realizar o *login*, os usuários deviam ter conhecimento sobre detalhes de atuação do agendador presente naquele agregado. Por exemplo, o usuário é obrigado a aprender a usar o PBS, SGE e OAR caso ele tenha acesso a três agregados, cada qual com um desses agendadores. Isso impulsionou a criação de um Portal Web para tratar funcionalidades comuns em agregados. Os usuários alvo do ICE são administradores de sistemas e principalmente aqueles de outras áreas de ciência (como hidrologia ou biologia) que não são especialistas no uso de agregados. O cenário de execução do ICE pode ser visto na figura 1. Nela pode-se observar a comunicação usuário-portal e portal-agregado. É escondido do usuário a interação com os agregados, bem como o modo de uso de suas ferramentas.

O projeto da arquitetura do ICE objetiva modularidade para que novas funcionalidades de agregados e suas implementações possam ser integradas ao ambiente. No ponto de vista do ICE, cada funcionalidade de agregado (como gerenciamento de trabalhos e monitoramento) pode ser mapeada para um módulo cliente de serviço, chamado SM, e possui uma interface USI. A arquitetura do ICE está apresentada na figura 2. Ela é baseada no modelo de componentes SOA e está dividida em três módulos principais: (i) Portal Web; (ii) Interface Unificada de Serviço, ou USI; (iii) Implementação do Serviço, ou SI. Essa separação da

parte de código que está no Portal da parte presente nos agregados visa distinguir o cliente de um serviço de sua implementação na arquitetura fracamente acoplada SOA. Para viabilizar esse tipo de arquitetura o projeto do ICE escolheu Web Services[13, 16].

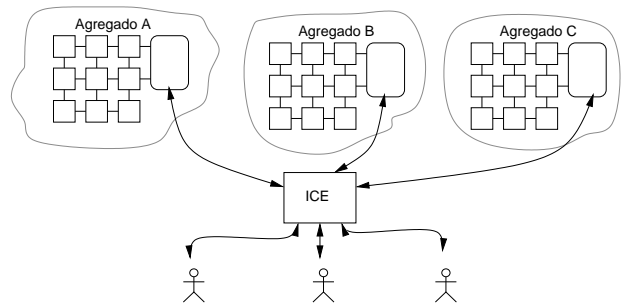


Figura 1. Ambiente de execução do ICE

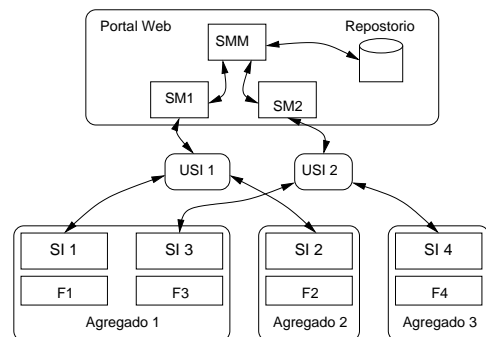


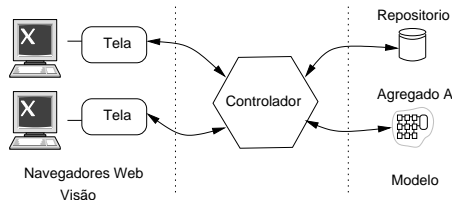
Figura 2. Arquitetura de módulos do ICE

A idéia do ICE difere de uma grade pois os agregados dela são acessados de maneira independente e não existe uma transparência de localização. Os agregados vinculados ao ICE não interagem entre si por esse ambiente. Além disso, no ICE os usuários têm conhecimento do agregado usado, mas não sobre os detalhes de uso nem de acesso à ferramenta empregada para realizar a operação desejada. As próximas subseções descrevem a divisão dos módulos do ICE desenvolvidos para disponibilizar a sua arquitetura.

### 2.1 Portal Web

O módulo Portal é uma aplicação Web que executa no servidor e encapsula o acesso a funcionalidades de agregados através de uma interface gráfica. Ele é subdividido em outros dois sub-módulos que são: Módulo de Gerenciamento do Sistema, ou SMM, e Módulo de Serviço, ou SM (ver figura 2). Uma das tarefas do primeiro é controlar a apresentação da interface Web. Para facilitar o desenvolvimento de aplicações desse tipo, o ICE foi construído segundo o modelo MVC (Modelo-Visão-Controlador)[11]. Essa abordagem permite separar a apresentação dos dados da lógica da aplicação (acesso a banco de dados e chamada

de Web Services) e está ilustrada na figura 3. O próprio servidor do Portal hospeda a parte da visão, do controlador e a lógica (modelo) referente ao acesso ao banco de dados.



**Figura 3. Modelo-Visão-Controlador**

O SMM atua como um controlador entre a parte da saída para o navegador Web e a parte da lógica de negócio do Portal. Entre as suas tarefas pode-se citar: autenticação do usuário; manter informações de agregados, Web Services, usuários e funcionalidades no Portal; tratar as permissões de acesso a agregados e as funcionalidades para cada usuário. Para amparar nas decisões do SMM, foi projetado um repositório central no Portal como pode ser visto na figura 2. A interface Web é dinamicamente atualizada pelo controlador SMM de acordo com as permissões de cada usuário para acessar funções do Portal e funcionalidades dos agregados. Como consequência, a apresentação dos menus do Portal é personalizada para cada usuário. Nesse ponto é relevante mencionar que o ICE possui um administrador que controla o cadastro de usuários e o seu mapeamento para funcionalidades suportadas pelo ambiente.

SM é um módulo do Portal que trata de uma funcionalidade de agregado. A figura 2 ilustra dois módulos SM, que poderiam ser funcionalidades como monitoramento e configuração. Ele é chamado pelo controlador SMM e atua como um cliente de Web Services que segue uma interface de serviço descrita pelo módulo USI. Ele é responsável por contatar o provedor de serviço de uma funcionalidade, receber a resposta da requisição e repassá-la ao SMM.

## 2.2 Interface Unificada de Serviço (USI)

Um módulo USI é a interface de um serviço para agregados. A sua definição envolve a especificação de operações de um serviço, bem como os parâmetros de cada operação. Através dele é possível uniformizar o caminho de acesso a diferentes ferramentas de agregados que possuem o mesmo propósito, ou seja, que se referem a mesma funcionalidade. Essa abordagem possibilita ao ICE tratar tantas funcionalidades quanto o número de módulos USI. O número de módulos SM é sempre igual ao de USI. Além disso, módulos SI implementam uma USI e módulos SM a usam para criar clientes de módulos SI. O processo de criação de uma USI consiste em analisar ferramentas de agregados para a mesma funcionalidade e apontar parâmetros e operações em comum que possam existir entre elas. Conceitualmente, uma USI pode ser comparada com uma

descrição WSDL de um Web Service. Entretanto, o que o diferencia desse último é que não informa o endereço do serviço como o WSDL faz.

## 2.3 Implementação de Serviço (SI)

O módulo SI é um componente provedor de serviço segundo a arquitetura de Web Services e é construído segundo a interface USI. Pode-se implementar vários módulos SI baseados em uma única USI. Muitas vezes, o mapeamento entre a definição da USI de uma funcionalidade e os parâmetros de uma determinada ferramenta não é exato. Nestas situações, deve ser feita uma adaptação na construção do módulo SI. Ela pode consistir no uso de banco de dados e ferramentas auxiliares. Uma implementação SI pode ser escrita em qualquer linguagem de programação (desde que tenha um servidor de aplicações que a suporte), visto que Web Services tem interoperabilidade através de mensagens XML-SOAP.

A figura 2 apresenta a implementação de quatro módulos SI. Um agregado pode ter mais de um módulo SI. Por exemplo, o agregado 1 nessa figura possui o SI1 da ferramenta F1 e segue a interface USI1. Em adição, ele possui a SI3 que se refere a outra USI e a ferramenta F3. A SI1 poderia ser um serviço para trabalhar com PBS, segundo a USI de gerenciamento de trabalhos, e a SI3 o serviço para operar com o Ganglia, ferramenta de monitoramento.

A arquitetura do ICE, baseada em módulos SM, USI e SI, atinge extensibilidade em dois níveis. O primeiro nível é o SM, uma vez que a arquitetura permite a incorporação de várias funcionalidades de agregados dependendo do número de módulos USI definidos. O segundo nível é atingido com módulos SI, visto que é possível integrar ao ICE várias ferramentas para uma dada funcionalidade. Esse arcabouço torna possível o uso de agregados em diversos domínios administrativos, com máquinas potencialmente heterogêneas, bem como a gerência de agregados a partir de um ponto uniforme de acesso.

## 2.4 Segurança no Ambiente ICE

O ICE oferece autenticação, autorização e criptografia no transporte de dados. Primeiramente, cada usuário fornece um *login* e senha para acessar o Portal Web que são conferidos com os dados no repositório. Caso aconteça a confirmação deles, ambos são cifrados e gravados na sessão do servidor para serem usados em chamadas Web Services.

A autorização trabalha com permissões de usuários para acessar páginas do Portal. Ela acontece em duas situações: (i) sobre as ligações para acessar operações do próprio Portal (usuário-portal); (ii) sobre ligações para acessar funcionalidades de um agregado (portal-agregado). Como exemplo, um usuário pode ter acesso ao menu de gerenciamento de trabalhos e selecionar a opção de submissão. Após escolher um agregado e submeter o formulário, acontece a

verificação por parte do controlador (ver subseção 2.1) se o usuário é ou não autorizado a acessar aquele agregado. Todas as ligações do ICE estão no menu do Portal que trata das funcionalidades do ambiente. Na criação ou alteração dos dados de um usuário, pode-se atribuir ou retirar ambos tipos de permissões a ele.

Para disponibilizar autenticação na manipulação de Web Services, foram usados certificados para garantir identidade entre a máquina onde reside o Portal ICE e os pontos de acesso (*front-ends*) de cada agregado. Isso garante que somente o Portal possa chamar os serviços residentes num agregado. A autenticação é implementada configurando o servidor de aplicações (*container*) na máquina do Portal ICE e nos agregados. Cada qual trabalha com arquivos que representam os certificados de autenticação mútua.

A criptografia dos dados é feita através do protocolo seguro de transporte HTTPS. A abordagem usada garante a criptografia sobre dados transferidos da máquina do usuário até o Portal, bem como desse último até os agregados. Nessa interação ocorre a criptografia sobre mensagens SOAP-XML (que por si só são montadas com texto limpo).

## 2.5 Tecnologias de Implementação

O servidor de aplicações utiliza o Apache Tomcat versão 5 e foi usada a linguagem Java para escrever os códigos no lado do Portal Web (controlador, código do SM) e no lado dos agregados (código do SI). O sistema Apache Axis versão 2 foi usado como *middleware* para a escrita de clientes e provedores de Web Services. Para repositório foi usado o banco de dados PostgreSQL com suporte a JDBC.

O modelo MVC é atingido usando Velocity e Tiles para a construção da interface Web e Struts para gerenciamento do controlador. Esse modelo permitiu o desenvolvimento paralelo da parte gráfica e da parte das lógicas do controlador e dos módulos SM. O controlador no Struts atua com a noção de ações e sub-ações que são mapeadas para uma funcionalidade de agregado e uma operação dela, respectivamente. Uma ação é vista como uma classe Java e uma sub-ação como um dos seus métodos. Qualquer ligação é intermediada pelo controlador e destinada a um dos métodos de uma ação. Antes disso, acontece a verificação das permissões.

## 3 Desenvolvimento de Serviços ICE

O Portal Web do ICE oferece serviços administrativos para configurar o ambiente e serviços para acessar funcionalidades de agregados. Os serviços administrativos tratam com a manutenção de Web Services, usuários, agregados, funcionalidades e configuração de banco de dados e *driver* JDBC. A figura 4 apresenta a lista de Web Services do ambiente ICE. Cada linha da tabela representa uma instância e é associada a um agregado, a um tipo de Web Services e a um endereço Web.

Count	Cluster	Type	Description	Delete	Update
1	labtec	Monitoring	Monitoring WS	Delete	Update
2	labtec	Upload	Uploader WS	Delete	Update
3	labtec	JobSubmission	Job Management WS	Delete	Update
4	LCP	Monitoring	Monitoring WS	Delete	Update
5	frontal-minuano	Upload	Uploader WS	Delete	Update
6	frontal-minuano	JobSubmission	Job Management WS	Delete	Update

Figura 4. Configuração de Web Services

Atualmente, o ICE suporta três tipos de Web Services: (i) carregamento de arquivos (*uploading*); (ii) gerenciamento de trabalhos; (iii) Monitoramento de recursos. Eles constituem os serviços para acessar funcionalidades de agregados e são descritos a seguir.

### 3.1 Serviço de Carregamento de Arquivos

Algumas aplicações necessitam de arquivos auxiliares para serem executadas. Nestes casos, o ICE permite a transferência de arquivos em dois passos. O primeiro entre a máquina do usuário e a do Portal Web e o segundo entre essa e a do ponto de acesso do agregado. Para viabilizar o primeiro caso, o Portal Web possui um menu para tratamento de arquivos que possui a operação de listar aqueles que já foram carregados até a máquina do Portal ICE. Na tela que se abre é possível carregar outros arquivos, assim como descarregar (*downloading*) ou apagar algum que foi enviado previamente. Cada arquivo na máquina Portal é gravado no banco de dados.

O segundo passo acontece no momento da submissão de trabalhos (ver subseção 3.2). Existe uma caixa no formulário que descreve um trabalho que possibilita selecionar um ou mais arquivos para serem transportados da máquina do Portal até o agregado alvo. Para isso, foi desenvolvido o serviço de carregamento de arquivos com Web Services. No momento da submissão dos dados, é verificada a permissão da operação sobre o agregado, capturado o endereço do Web Services do tipo *uploading* dele e é chamado o módulo SM. Esse por sua vez cria o envelope da mensagem (com o nome do arquivo, o arquivo em si e onde ele deve ser gravado) e chama o provedor de serviço SI. A mensagem é enviada usando o mecanismo otimizado MTOM (*Message Transmission Optimization Mechanism*). O SI implementa o seguinte método da USI de carregamento de arquivos:

- `upload()` - Cria um objeto arquivo informando o lugar na árvore de diretórios e o nome especificado, grava os dados do arquivo e retorna código da operação.



A abordagem construída, baseada em duas etapas, tem a vantagem de diminuir o tempo no transporte de arquivos na submissão de trabalhos, uma vez que eles já estão no Portal. Além disso, é possível que o arquivo enviado seja o próprio binário da aplicação a ser executada.

### 3.2 Serviço de Gerenciamento de Trabalhos

A funcionalidade de gerenciamento de trabalhos possui um espaço no menu do Portal com operações para submeter e listar trabalhos, bem como visualizar gráficos estatísticos. A tela de submissão de trabalhos está ilustrada na figura 5. Os campos do formulário dessa figura refletem os parâmetros da operação de submissão da USI dessa funcionalidade. O primeiro campo dele se refere ao agregado alvo. O ICE emprega o conceito que um agregado possui somente um agendador e, ao marcar o agregado seleciona-se transparentemente ao usuário o agendador. Atualmente, ICE suporta submissão nos agregados da UFRGS Labtec e Frontal-Minuano. Acompanhando o formulário, completa-se o nome da aplicação e seus parâmetros, os arquivos de saída padrão e de erro. É possível unir os dados de erro com o arquivo de saída selecionando a opção "Join Output". O ICE suporta somente aplicações já compiladas. O binário da aplicação já está no agregado ou pode ser levado até lá usando o carregamento de arquivos.

Figura 5. Submissão de trabalhos no ICE

Ainda no formulário, deve-se completar o número de iterações de execução da aplicação. Ela é importante para a avaliação de aplicações paralelas onde são necessárias várias execuções. Preenche-se também o número de nós requeridos, o número de processadores por nó e o tempo a ser alocado para o trabalho. O próximo campo diz respeito ao tipo de aplicação. A implementação atual do ICE interage com quatro tipos de aplicações: seqüencial, MPI, DECK[1] e CJava[5]. Por fim, é possível escolher um ou mais arquivos a serem carregados até o agregado destino.

Na submissão, além da verificação da necessidade do transporte de arquivos, é chamado o módulo SM de gerenciamento de trabalhos usando o endereço do Web Service do SI do agregado alvo. O agregado Labtec possui um módulo

SI para trabalhar com o agendador OpenPBS, enquanto o Frontal-Minuano tem um SI para o SGE. Os dados de cada submissão são gravados no banco de dados do Portal.

```
1. #!/bin/sh
2. #PBS -l nodes=10,ppn=2,walltime=1:00:00
3. #PBS -q labtec
4. #PBS -o /home/gppd/righi/saida
5. #PBS -j eo
6. /usr/local/mpich/bin/mpirun -machinefile $PBS_NODEFILE
   -np 10 /home/gppd/righi/app
```

Figura 6. Script PBS para o agregado Labtec

A operação de listagem apresenta uma tabela com todos os trabalhos já submetidos pelo usuário. Cada linha dela informa características da aplicação e o seu estado. Esse último é capturado usando uma chamada Web Service ao módulo SI e pode ter o valor enfileirado, executando ou finalizado. Ao lado das informações do trabalho, tem-se um botão para emitir o *script* elaborado para o lançamento do trabalho. Um exemplo de *script* para o agregado Labtec e gerado pelo SI (para a ferramenta OpenPBS) pode ser visto na figura 6. A sua criação é totalmente abstraída do usuário. Existe também o botão para capturar a saída do trabalho, outro para cancelá-lo (caso estiver em execução) e o botão de apagar, que retira o seu registro da base de dados. Caso o usuário quiser apagar um trabalho em execução, esse é automaticamente cancelado.

Um SI que segue a USI de gerenciamento de trabalhos deve implementar os seguintes métodos:

- `finalizeJob()` - Recebe o identificador e exclui o trabalho em execução referente do agregado alvo;
- `verifyStatusJob()` - Recebe o identificador do trabalho, realiza a requisição e retorna o estado;
- `getScriptJob()` - Recebe os dados de um trabalho e monta o *script* segundo o seu tipo e a ferramenta de agendamento no agregado e retorna o *script*;
- `submitJob()` - Recebe os dados que descrevem um trabalho, monta o *script*, submete o trabalho e retorna código da operação;
- `getOutputJob()` - Recupera e retorna o arquivo de saída da aplicação, a partir do identificador.

Além das operações listadas acima, o usuário pode obter gráficos em pizza a respeito de seus trabalhos e da utilização de recursos. Eles são montados com informações do banco de dados e são mostrados em duas classificações gerais: (i) usuário e (ii) global. A classificação tipo usuário mostra dois gráficos que podem ser vistos na figura 7. O primeiro deles mostra uma relação entre a quantidade e o tipo de trabalho submetido pelo usuário. O segundo gráfico informa a relação de trabalhos enviados a cada agregado do ICE. A

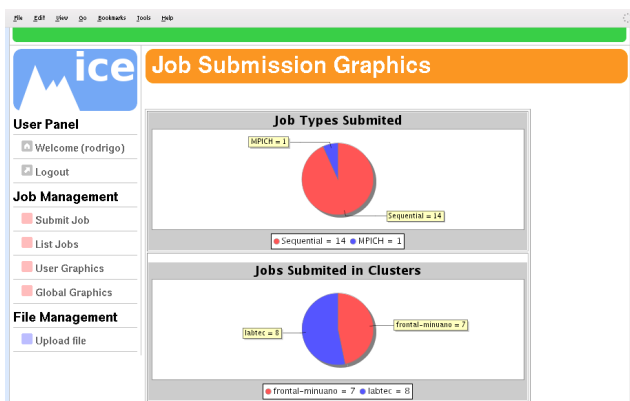


Figura 7. Gráficos relevantes ao usuário

segunda classificação, global, mostra um gráfico para cada agregado associado ao ICE. Cada um mostra a quantidade de trabalhos de cada usuário naquele agregado. Os gráficos dessa classificação são úteis para administradores e o acesso a usuários normais pode ser controlado nas suas permissões.

### 3.3 Serviço de Monitoramento de Recursos

A funcionalidade de monitoramento de recursos oferece aos usuários a possibilidade de obter informações a respeito das máquinas que podem ser utilizadas para a execução de aplicações. A sua principal característica é permitir que o usuário não se atenha a detalhes de como as informações são coletadas. O serviço de monitoramento agrega operações para configuração dos dados que devem ser monitorados, a visualização instantânea e textual dessas informações, a geração de gráficos e a configuração de um conjunto de parâmetros para que o Portal fique responsável por coletar informações dos recursos.

A configuração dos dados que devem ser monitorados se dá, primeiramente, através da especificação de quais máquinas o usuário quer monitorar. Para isso, o Portal Web fornece ao usuário o conjunto de máquinas que ele pode obter informação de monitoramento. Na segunda parte dessa configuração ocorre a especificação de quais métricas de cada máquina devem ser monitoradas. Da mesma forma que nas máquinas, o Portal oferece as métricas disponíveis.

Depois de ter o ambiente de monitoramento configurado, o usuário pode obter as informações que são coletadas pelo Portal de duas formas: representação textual ou gráfica. Na representação textual, o usuário recebe um relatório de valores instantâneos das métricas de monitoramento configuradas. Esse relatório pode incluir métricas que não são associadas a um valor, como por exemplo a versão do sistema operacional de determinada máquina. Para a representação gráfica, o usuário deve configurar a parte do serviço de monitoramento dinâmico. Essa parte inclui a especificação do intervalo de coleta de dados de monitoramento. Esse in-

tervalo é utilizado pelo Portal para configurar um fluxo de execução (*thread*) que, de acordo com esse valor, coleta informações a respeito dos recursos. Dessa forma, o usuário não precisa ficar gerando relatórios textuais, uma vez que o Portal se encarrega de coletar as informações periodicamente e, posteriormente, de apresentá-las em gráficos.

A parte dinâmica de coleta de dados pode ser inicializada e finalizada diretamente no menu de monitoramento do Portal. A figura 8 apresenta a tela principal de configuração, onde o usuário interage com o Portal no sentido de configurar as questões apresentadas nos parágrafos anteriores, como quais máquinas e métricas devem ser monitoradas.

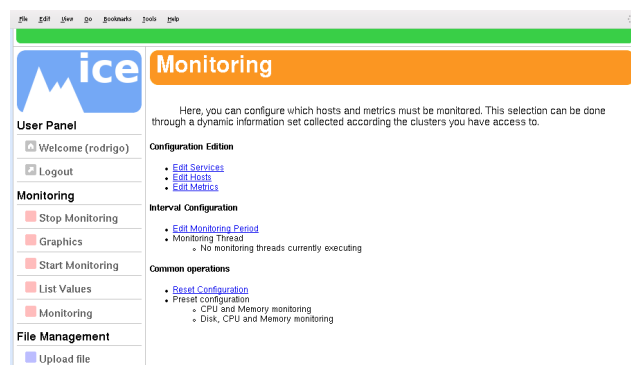


Figura 8. Serviço de Monitoramento do ICE

Para viabilizar o funcionamento do serviço de monitoramento, o SM dessa funcionalidade no Portal funciona como um cliente de Web Services. Esse cliente sabe acessar a seguinte USI de monitoramento:

- `getHosts()` - Operação sem parâmetros deve retornar as máquinas monitoradas por um determinado SI;
- `getMetrics ()` - Recebe um conjunto de máquinas e retorna as métricas disponíveis para cada uma delas;
- `getValues ()` - Recebe um conjunto de máquinas e métricas e retorna os respectivos valores instantâneos.

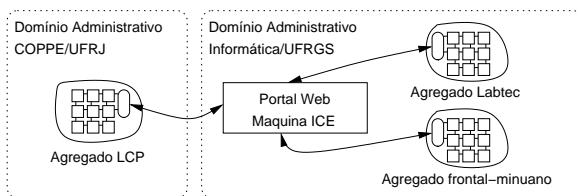
A USI de monitoramento está implementada para dois SI: Ganglia e Performance Co-Pilot. Dessa forma, agregados que tenham algum desses dois monitores de recursos podem ter suas informações publicadas no Portal de forma transparente. Atualmente, os agregados Labtec, na UFRGS, e LCP, na UFRJ, estão executando o módulo SI para o Ganglia. As informações do serviço de monitoramento apresentadas no Portal podem ser úteis para detectar eventuais problemas em recursos. Em adição, o monitoramento permite observar o comportamento dos recursos quando uma aplicação é colocada em execução.

## 4 Plataforma de Testes do ICE

O ICE foi projetado, inicialmente, para englobar agregados do Instituto de Informática da UFRGS e o Laboratório

de Computação Paralela (LCP) da COPPE/UFRJ. A figura 9 mostra a plataforma de testes do ICE. Ela reúne os agregados Labtec e Frontal-Minuano por parte da UFRGS e LCP na COPPE. Essa organização possibilita testar programas CJava[5] (ferramenta que trabalha com memória compartilhada e distribuída implementada na COPPE) em agregados da UFRGS. Além do mais, ela aparece como uma plataforma de máquinas paralelas que, junto ao Portal Web, pode ser usada para a execução de aplicações científicas de outros grupos de pesquisa de ambas as universidades.

A figura 9 mostra o ambiente de testes do ICE. A idéia é que a máquina do Portal seja acessível pela Internet. Além disso, os agregados fora do domínio da UFRGS também devem ser atingíveis por esse meio. Atualmente, o Portal ICE está instalado em uma máquina da UFRGS não acessível pela Internet. Ele será disponibilizado para pesquisadores tão logo seja adquirido um servidor para hospedá-lo.



**Figura 9. Plataforma de testes para o ICE**

Os três serviços de funcionalidades do ICE mais os serviços administrativos do ambiente estão em funcionamento e foram testados. Alguns resultados das telas do ICE estão apresentados na seção 3. O próximo passo é disponibilizá-lo e verificar o retorno da sua utilização principalmente das pessoas não especialistas em agregados. Como resultado dos testes, observou-se que o tempo de execução dos serviços do ICE depende diretamente da carga do processador no ponto de acesso do agregado alvo. Em geral, quando a carga desses pontos de acesso é baixa, os serviços do ICE têm um bom desempenho.

## 5 Trabalhos Relacionados

A literatura a respeito de Portais Web para trabalhar com máquinas paralelas pode ser dividida em grupos, da seguinte forma: (i) Portais Web para aplicações específicas; (ii) Kits para desenvolvimento de Portais; (iii) Portais de uso genérico voltados para o usuário. No primeiro grupo pode-se citar o Bio Portal[21], que constitui uma grade com o Globus. Os recursos são automaticamente selecionados de acordo com a entrada dos dados usando uma ferramenta proprietária. Beeson et al.[3] desenvolveram um Portal para simulações de astrofísica. Ele também usufrui do Globus, agora integrado ao gerente de recursos Gridbus. Esse Portal usa o Struts e sua versão corrente impõe ao usuário preencher os parâmetros de sua aplicação em um arquivo.

Quanto aos Kits para construção de Portais, é possível citar o GridSpeed[18], Ninf[19], GridPort[6], GPDK[14]. Todos eles usam o Globus para gerir serviços de grade. O GridSpeed suporta aplicações baseadas em tarefas simples, tarefas instanciadas com parâmetros diferentes e a combinação dos dois casos com relações de dependência entre tarefas. Ninf viabiliza aplicações segundo o modelo RPC, usando o GridRPC. Ela necessita de um arquivo XML que descreve a aplicação. O GridPort fornece uma API para a construção de Portais (como o Hotpage) e trabalha com CGI. Diferente do ICE, ele suporta a compilação de programas. O GPDK (*The Grid Portal Development Kit*) usa o sistema CoG (*Java Commodity Grid*) para a sua interface e atua com aplicações modeladas em tarefas.

Dentro do grupo de Portais genéricos, o Pegasus[17] é usado para submeter grafos de tarefas para grades. Para isso, ele faz uso do Condor-G. Tourino et al.[20] apresentam um Portal baseado no Kit GPDK para ensino de processamento paralelo. Assim como o ICE, ele também possibilita a seleção de um agregado e a submissão de trabalhos MPI. WebCom[12] é voltado para grades e é implementado com CGI. Para a submissão de um programa são necessários os fontes, um arquivo Makefile (para compilar e ligar) e um arquivo que descreve o ambiente de execução. Outro trabalho relevante é o GridSphere[15] que apresenta a sua parte gráfica usando componentes reutilizáveis Web chamados Portlets. Cada Portlet pode ser um pedaço da tela e referenciar uma operação de agregado. O seu emprego é estudado para as próximas versões do ICE. Finalizando, tem-se o Portal HPC2N[7] para tratar múltiplos agregados. Ele trabalha somente com os agendadores PBS e Maui e não apresenta um arcabouço genérico para trabalhar com funcionalidades de agregados. Além disso, ele obriga o usuário a escrever um *script* para a submissão de trabalhos.

O Globus é bastante usado em Portais. Entretanto, optou-se por não usá-lo porque o ICE não opera em grade e sim com múltiplos agregados acessados de forma independente. Em adição, agregados por si só normalmente não possuem o Globus instalado. Na submissão de trabalhos, o ICE exige que o usuário tenha conta no agregado alvo e o seu foco são aplicações paralelas. Em contra-partida, nota-se que a maioria dos sistemas acima tratam aplicações como sacola de tarefas. A maior contribuição do ICE é a sua idéia de extensibilidade, que não é o alvo dos sistemas estudados.

## 6 Conclusão

Este artigo apresentou o Portal ICE e o seus módulos. Com a arquitetura SOA (viabilizada com Web Services), ele engloba clientes de serviço SM, uma interface para cada funcionalidade de agregado USI e provedores de serviço SI. Essa combinação permite atingir o diferencial do ICE que é extensibilidade em dois níveis: (i) através da inclusão de funcionalidades de agregados especificando uma interface

USI e um módulo SM; (ii) construção de vários serviços (implementação SI) para a mesma funcionalidade. O ICE pode se adaptar a diferentes ferramentas. Assim, ele pode englobar novos agregados através do desenvolvimento de módulos SI para as ferramentas específicas deles.

O Portal Web é o ponto de entrada do ICE e oferece uma interface gráfica simplificada para os usuários. O ICE é direcionado para usuários das áreas de ciência que não são especialistas no uso de agregados (como área da biologia). Eles somente escolhem a funcionalidade desejada, a operação e o agregado e não precisam se preocupar com qual ferramenta será utilizada, nem como. Como contribuição para esses usuários, o ICE fornece uma fácil interação com máquinas para a computação de alto desempenho. O ICE foi desenvolvido com o modelo MVC (Modelo-Visão-Controlador). Por causa disso, a manutenção e a evolução do Portal são facilitadas pela divisão da apresentação e da lógica de negócio.

O ICE possui Web Services para as funcionalidades de carregamento de arquivos, monitoramento de recursos e gerenciamento de trabalhos. Referente a essa última, ele pode operar com aplicações sequenciais, além de paralelas do tipo MPI, CJava e DECK. Cada aplicação é direcionada a um agregado e eles são independentes. Atualmente, as três funcionalidades citadas acima estão em funcionamento no ICE. A sua plataforma inicial de testes inclui dois agregados na UFRGS e outro na UFRJ.

A cargo de trabalho futuro pode-se citar a seguinte tarefa. Construção junto aos serviços de administração do ICE de um módulo de mapeamento de usuário-agregados. Nele seriam completados o nome de entrada do usuário em cada agregado e o seu diretório local. Hoje esses dados são preenchidos no momento da submissão de trabalhos.

## Referências

- [1] M. Barreto, R. Ávila, and P. Navaux. The MultiCluster model to the integrated multiple workstation clusters. In *Workshop on Personal Computer based Networks of Workstations*, volume 1800, pages 71–80, Cancun, 2000. Springer.
- [2] B. Barrett, J. M. Squyres, and A. Lumsdaine. Integration of the LAM/MPI environment and the pbs scheduling system. In *17th International Symposium on High Performance Computing Systems and Applications and OSCAR Symposium*, pages 277–283, 2003.
- [3] B. Beeson, S. Melnikoff, and D. Barnes. A portal for grid-enabled physics. In *Workshop on Grid computing and e-research*, pages 13–20, Darlinghurst, Australia, 2005.
- [4] R. Buyya. *High Performance Cluster Computing: Architectures and Systems*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.
- [5] A. F. da Silva, M. Lobosco, and C. L. de Amorim. An evaluation of cjava system architecture. In *Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, page 91, Washington, USA, 2003. IEEE.
- [6] M. Dahan and J. R. Boisseau. The gridport toolkit: A system for building grid portals. In *International Symposium on High Performance Distributed Computing*, page 216, Washington, USA, 2001. IEEE Computer Society.
- [7] E. Elmorot, M. Nylen, and R. Oscarson. A user-centric cluster and grid computing portal. In *International Conference Workshop on Parallel Processing*, pages 103–110, 2005.
- [8] I. Foster. Globus toolkit 4: Software for service oriented systems. In *Conference on Network and Parallel Computing*, volume 3779, pages 2–13. Springer-Verlag, 2005.
- [9] I. Foster. Service-oriented science. *Science*, 308(5723):814–817, 2005.
- [10] W. Gentsch. Sun grid engine: towards creating a compute power grid. In *Symposium on Cluster Computing and the Grid*, pages 35–36, Brisbane, Australia, 2001. IEEE.
- [11] S. Hansen and T. V. Fossum. Refactoring model-view-controller. *Journal of Computing Sciences in Colleges*, 21(1):120–129, 2005.
- [12] G. He and Z. Xu. Design and implementation of a web-based computational grid portal. In *International Conference on Web Intelligence*, pages 478–482. IEEE, 2003.
- [13] P. Kumar. Web services in it management. *ACM Queue*, 3(6), August 2005.
- [14] J. Novotny. The grid portal development kit. *Concurrency and Computation: Practice and Experience*, 14(13-15):1129–1144, 2002.
- [15] J. Novotny, M. Russell, and O. Wehrens. Gridsphere: An advanced portal framework. In *30th EUROMICRO*, pages 412–419, Washington, USA, 2004. IEEE.
- [16] M. Pierce, G. Fox, C. Youn, S. Mock, K. Mueller, and O. Balsoy. Interoperable web services for computational portals. In *ACM/IEEE conference on Supercomputing*, pages 1–12, Los Alamitos, USA, 2002. IEEE Press.
- [17] G. Singh, E. Deelman, G. Mehta, K. Vahi, M.-H. Su, G. B. Berriman, J. Good, J. Jacob, D. Katz, A. Lazzarini, and S. Koranda. The pegasus portal: web based grid computing. In *ACM Symposium on Applied computing*, pages 680–686, New York, USA, 2005. ACM Press.
- [18] T. Suzumura, S. Matsuoka, H. Nakada, and H. Casanova. Gridspeed: A web-based grid portal generation server. In *High Performance Computing and Grid in Asia Pacific Region*, pages 26–33, Washington, USA, 2004. IEEE.
- [19] T. Suzumura, H. Nakada, M. Saito, S. Matsuoka, Y. Tanaka, and S. Sekiguchi. The ninf portal: an automatic generation tool for grid portals. In *Conference on Java Grande*, pages 1–7, New York, USA, 2002. ACM Press.
- [20] J. Tourino, M. Matin, J. Tarrio, and M. Arenaz. A grid portal for an undergraduate parallel programming course. *IEEE Transactions on Education*, 48(3):391–399, 2005.
- [21] H. Yuasa, T. Iwasaki, M. Kurokawa, T. Shigetani, and R. Himeno. Development of user-friendly supercomputing portal in bio research field. In *Conference on e-Science and Grid Technologies*, pages 508–515. IEEE, 2005.