

# Grades Computacionais: Uma Introdução Prática

***Raphael Y. de Camargo***  
***Ricardo Andrade***

*Departamento de Ciência da Computação  
Instituto de Matemática e Estatística  
Universidade de São Paulo, Brasil*

*São Paulo, 23 de novembro de 2007*

## I. Grades Computacionais

1. Conceito
2. Tipos de grades computacionais
3. Exemplos de sistemas de grade

## II. InteGrade

1. Motivação e características
2. Arquitetura
3. Execução de aplicações no InteGrade
4. Tolerância a falhas

# PARTE I

## *Grades Computationais*

- Aplicações paralelas
  - Seqüenciamento de genes (genoma)
  - Mineração de dados (mercado financeiro)
  - Enovelamento de proteínas (ind. farmacêutica)
  - Análise de sinais (SETI)
  - Previsão do tempo (INPE/CPTEC)
- Alto poder computacional
- Grandes quantidades de dados

- Computação de alto-desempenho
  - Abordagens Tradicionais:
    - Supercomputadores
      - Custam milhões de dólares
      - Alto custo de manutenção
    - *Clusters* dedicados (Beowulf)
      - Computadores pessoais e redes TCP/IP
      - Baixo preço de aquisição e manutenção
      - Alta disponibilidade

- Grades computacionais
  - Infra-estrutura de software que permitem a *interligação e compartilhamento* de recursos computacionais.
  - Estes recursos normalmente são:
    - Altamente heterogêneos
      - Servidores de dados
      - *Clusters* de computadores pessoais
      - Supercomputadores
    - Geograficamente dispersos
    - Pertencentes a diferentes instituições

# Grades Computacionais

- Infra-estrutura computacional sofisticada
- Principais sistemas de grades para *e-science*
  - Supercomputadores, aglomerados de computadores
    - Recursos dedicados à grade
  - Sistemas de armazenamentos de dados
    - Armazenamento de Terabytes e Petabytes
  - Redes de comunicação de alta-velocidade
    - Largura de banda de vários Gbps

## Computação de alto-desempenho

- Recursos dedicados
- Configuração estática
- Gerenciamento central
- Ambiente controlado
- Máquinas homogêneas

## Grades computacionais

- Recursos compartilhados
- Configuração dinâmica
- Gerenciamento distribuído
- Ambiente não controlado
- Máquinas heterogêneas

## Computação de alto-desempenho

- Dezenas ou centenas de máquinas
- Máquinas localizadas no mesmo espaço físico
- Falhas pouco freqüentes

## Grades computacionais

- Dezenas de milhares de máquinas
- Máquinas geograficamente distribuídas
- Falhas constantes

# Principais Desafios

- Segurança
  - Para os donos dos recursos
  - Para os usuários da grade
- Tolerância a Falhas
  - Para os componentes da grade
  - Para as aplicações em execução na grade
- Escalonamento
  - Determinar onde uma aplicação deve executar

- Gerenciamento de recursos
  - Busca de recursos computacionais
  - Entrada e saída de recursos computacionais
- Gerenciamento de dados
  - Determinar onde dados devem ser armazenados
  - Busca por dados armazenados
  - Disponibilidade e desempenho
- outros desafios

# Grades de Dados

- Grades onde a ênfase é no gerenciamento e distribuição de dados
  - Física de Altas Energia
    - Dados gerados em um acelerador de partículas são utilizados por pesquisadores do mundo todo
  - Precisam gerenciar Petabytes de dados
    - Armazenamento e transmissão dos dados
    - Tolerância a falhas (Replicação)
    - Serviços de busca e diretórios de dados

- O middleware de grade mais conhecido atualmente
  - Pode ser baixado gratuitamente (Versão 4.0.5)
  - Padrão OGSA (Open Grid Services Architecture)
- Utiliza o conceito de *Grid Services*
  - Comunicação baseada em Web Services
    - Baixo acoplamento entre os recursos
    - Desempenho ruim
  - Adiciona funcionalidades a Web Services
    - Descoberta de serviços
    - Serviços nomeados
    - Serviços com estado

- É composto por um conjunto de serviços
  - Segurança
    - Baseado em GSS + Kerberos
  - GridFTP
    - Transferência eficiente de dados
  - Armazenamento e gerenciamento de réplicas
    - Tolerância a falhas e maior desempenho
  - Monitoramento e descoberta de recursos
  - Escalonamento
    - Provê apenas a interface para o escalonador

- Worldwide LHC Grid (Europa)
  - Criado para permitir a análise de aproximadamente 15 Petabytes de dados que serão gerados no LHC
  - Muitos dados para serem armazenados e analisados por uma única instituição
  - Criar e manter uma infraestrutura de armazenamento e análise dos dados gerados pelo LHC
  - Motivação para uso de Grades:
    - Melhor aproveitamento de recursos
    - Menos pontos únicos de falhas
  - <http://lcg.web.cern.ch/LCG/>

- Open Science Grid (EUA)
  - Consórcio de provedores de software, serviços e recursos computacionais
  - Prover um sistema distribuído para processamento e armazenamento de dados
  - Interligado ao Worldwide LHC Grid
  - <http://www.opensciencegrid.org/>
- HEPGrid (Brasil)
  - Permitirá a instituições brasileiras ter acesso a dados do LHC e do Open Science Grid
  - <http://www.hepgrid.uerj.br/>

# Grades Oportunistas

- Foco na utilização ciclos computacionais ociosos estações de trabalho compartilhadas
  - Máquinas de laboratórios, professores, etc.
  - Deve manter a Qualidade de Serviço dos donos das máquinas compartilhadas
  - Máquina passa do estado ocioso para utilizada
    - Aplicações da grade devem ser migradas para outras máquinas

# Grades Oportunistas

- **Vantagens**
  - ✓ Baixo custo: utiliza hardware já existente
  - ✓ Economia de recursos naturais
    - ✓ Eletricidade, refrigeração, espaço físico.
- **Desvantagens**
  - × Gerenciamento de recursos é mais complicado
  - × Menor desempenho: Utiliza apenas períodos ociosos das máquinas



# Grades Oportunistas: Desafios

- Segurança
  - Donos de recursos compartilhados
  - Usuários da grade
- Gerenciamento de recursos
  - Escalonamento de aplicações
- Qualidade de serviço
  - Máquinas compartilhadas

- Tolerância a falhas
  - Componentes da grade
  - *Execução robusta de aplicações paralelas*
- Dados de usuários e aplicações
  - Gerenciamento de repositórios de dados
  - Gerenciamento dos dados armazenados

# Exemplo: SETI@HOME

- Sistema de computação distribuída
  - Análise de sinais de rádio-telescópios
  - Objetivo é procurar por padrões não-naturais nos sinais obtidos
- Quebra o problema em blocos independentes
  - Não existe comunicação entre os nós
- Distribui grupos de blocos a cada computador
  - Computador realiza análise dos dados contidos nos blocos recebidos
  - Dados são enviados de volta a um servidor central

# Exemplo: SETI@HOME

- Ótimo exemplo do poder computacional
  - Desempenho total até Janeiro de 2006
    - 7.745.000.000.000 Gflops
    - Mais de 5 milhões de usuários
  - Core 2 Duo (2.16GHz): ~ 3 Gflop/s
    - 259.200 Gflops por dia
  - Computador mais rápido do mundo:
    - Blue Gene/L (212.992 processadores)
    - 478.200 Gflops/s
    - 187 dias no Blue Gene/L ↔ Seti@Home

- Folding@home
  - Sistema para realizar simulações computacionais do enovelamento de proteínas
  - Aplicações: cura de doenças, como o câncer, Alzheimer, Parkinson, etc.
  - Possui versão que pode ser executada em Playstation 3.

# Exemplo: Condor

- Grade oportunista
  - Desenvolvido na University of Wisconsin-Madison
  - Permite utilizar ciclos ociosos de estações de trabalho compartilhadas
- Condor pools: grupos de computadores
  - Condor pools podem ser conectados entre si
- Suporte a aplicações seqüências, bag-of-tasks e paralelas MPI

# Exemplo: Condor

- Computadores fazem anúncios dos recursos disponibilizados
  - Estes recursos são então monitorados pelo Condor
- Aplicação é submetida para execução
  - *Match-maker* repassa a execução a máquinas que possuam os recursos necessários para executar a aplicação
- Caso uma máquina seja requisitada pelo dono
  - Aplicação é migrada para outro nó
  - Aplicações paralelas rodam em nós dedicados

# Exemplo: OurGrid

- Grade oportunista do tipo peer-to-peer
  - Desenvolvida na UFCG
  - Parceria com a HP Labs
- Suporte a aplicações paralelas onde os processos não se comunicam
  - *Bag-of-tasks*
- Mecanismo para estimular a doação de recursos
  - Rede de favores

# Exemplo: OurGrid

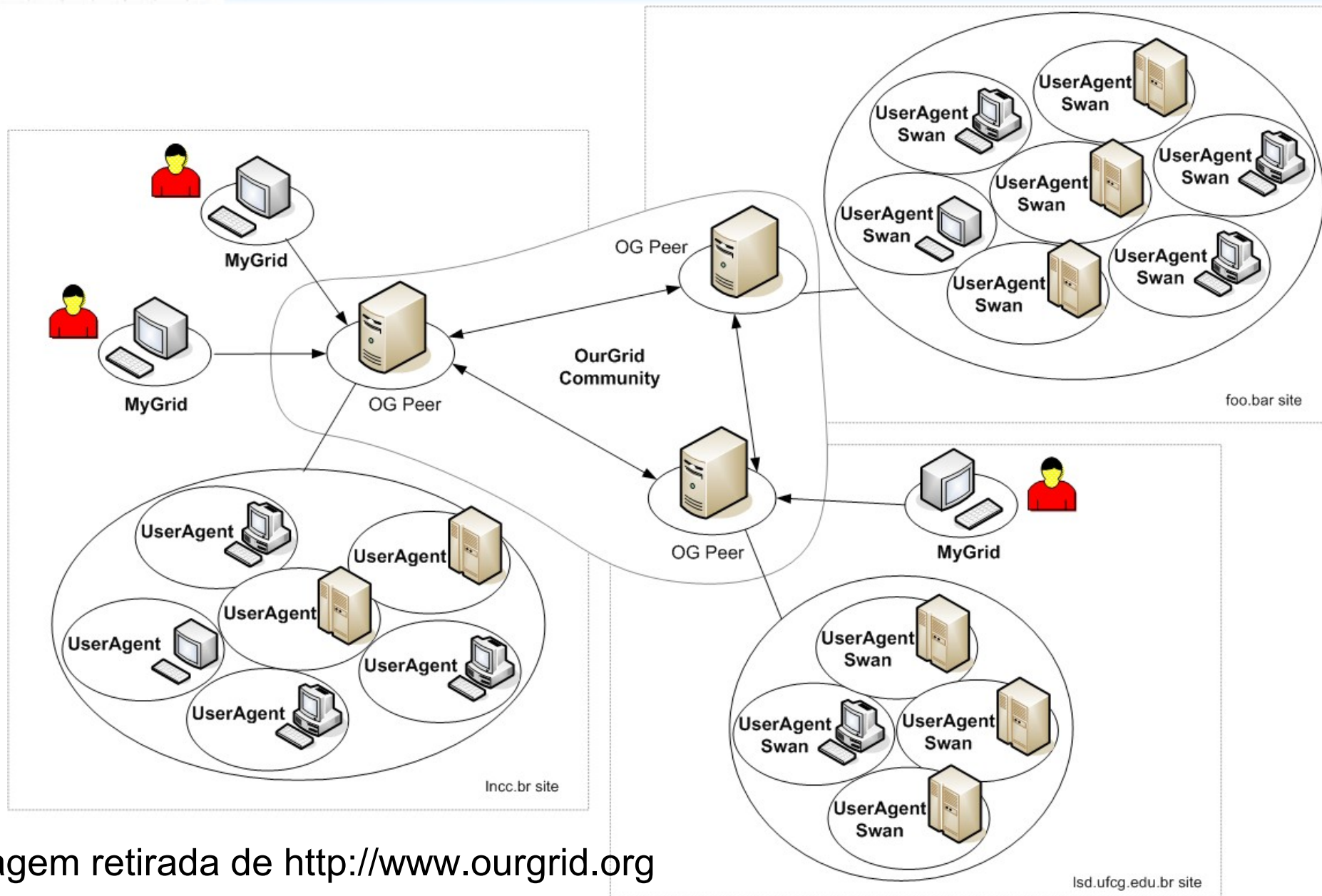


Imagem retirada de <http://www.ourgrid.org>

## PARTE II

# *InteGrade*

Infra-estrutura de software para  
Grades Oportunistas

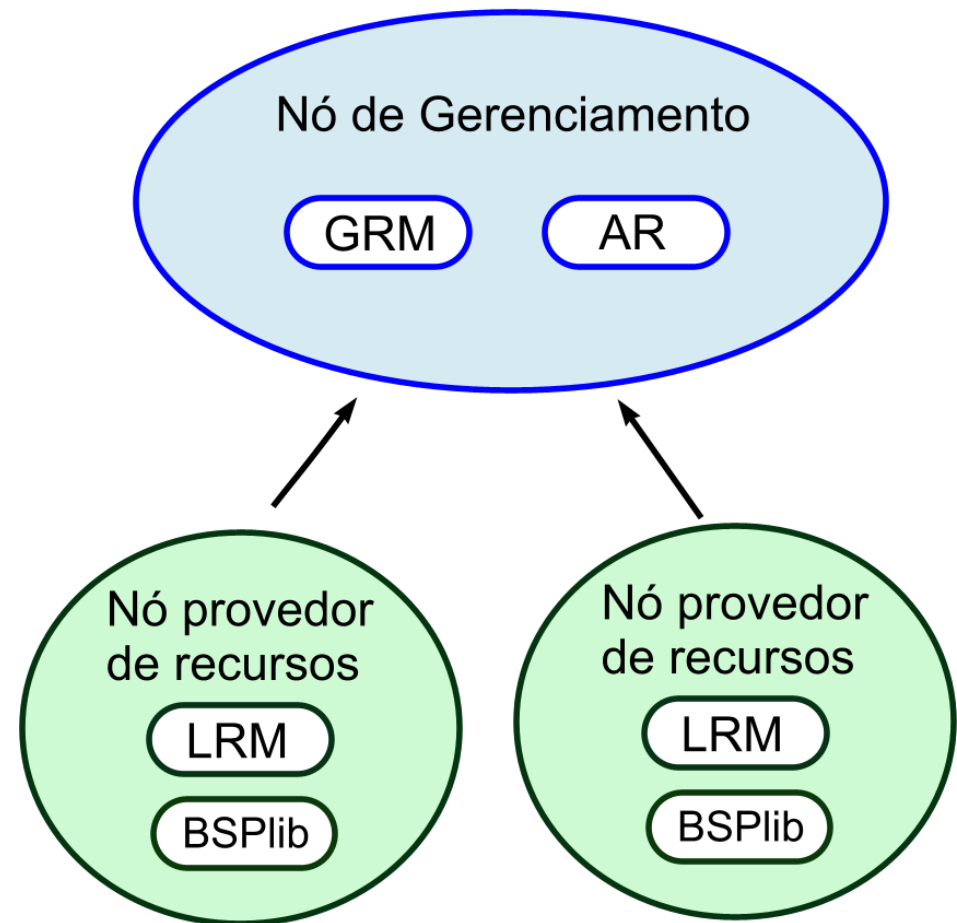
- **Motivação**
  - Desenvolver um middleware que permita utilizar recursos ociosos de máquinas compartilhadas
  - Objetivo é realizar a execução de aplicações paralelas computacionalmente intensivas
  - Armazenamento de dados de aplicações no espaço livre em disco das máquinas compartilhadas
- Importante para instituições com recursos financeiros escassos

- Foco na utilização de computadores pessoais
- Suporte a aplicações seqüenciais e paralelas
  - Bag-of-tasks, BSP e MPI
- Portal web para submissão de aplicações
- Em desenvolvimento:
  - Armazenamento distribuído de dados
  - Preservação da Qualidade de Serviço dos donos de máquinas compartilhadas
  - Segurança baseada em redes de confiança

# InteGrade: Arquitetura

- Arquitetura orientada-a-objetos baseada em CORBA
  - Java, C++ e Lua
- Aglomerados compostos por grupos de máquinas

## Aglomerado do InteGrade



# Principais componentes

- Global Resource Manager (GRM)
  - Gerencia os recursos de um aglomerado
  - Seleciona em quais máquinas cada aplicação submetida será executada
- Local Resource Manager (LRM)
  - Gerencia uma máquina provedora de recursos
  - Inicia a execução de aplicações na máquina
- Application Repository (AR)
  - Armazena os binários das aplicações dos usuários

# Principais componentes

- Application Submission and Control Tool (ASCT)
  - Permite a submissão de aplicações e a obtenção e visualização de resultados
- Portal Web
  - Versão web da ferramenta ASCT
- Local Usage Pattern Analyser (LUPA)
  - Analisa o padrão de uso das máquinas da grade
  - Permite ao escalonador realizar melhores escolhas no momento de definir máquinas que executarão uma aplicação



# Portal InteGrade

GridSphere Portal - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://agua1.ime.usp.br:8080/gridsphere/gridsphere?cid=83&gs\_action=setAppType&STR\_appType=bsj

Google

**InteGrade**

[Logout](#)  
Welcome, Raphael Y. Camargo

Welcome Administration **The Grid**

Repository **Job Submission** Grid Monitoring

### Job Submission

Sequential BSP Parametric

Preferences:

Constraints:

Arguments:

Number of tasks:

Output Files  + Input Files  
No available input files

Application Request Id	Execution state
Matriz 1	FINISHED <a href="#">(results)</a>
Matriz 0	FINISHED <a href="#">(results)</a>

Uploaded files will be available on the input files list

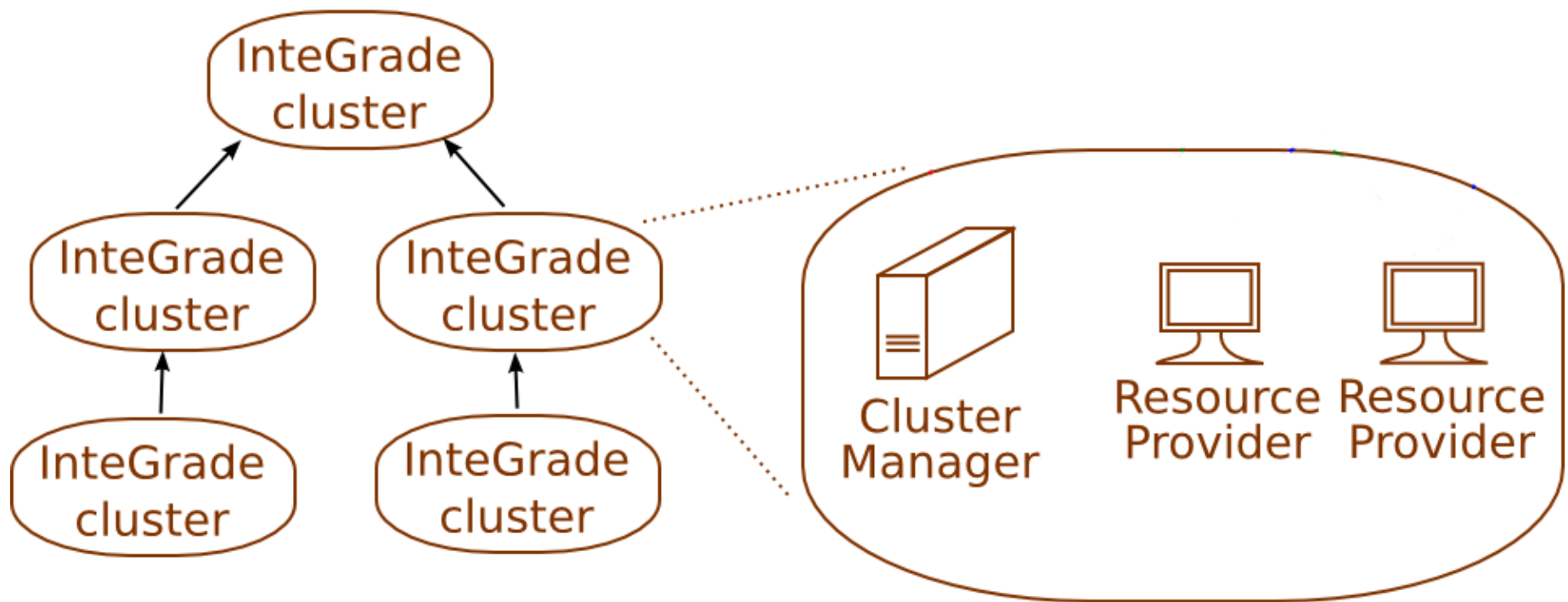
File:

31 de Agosto de 2007

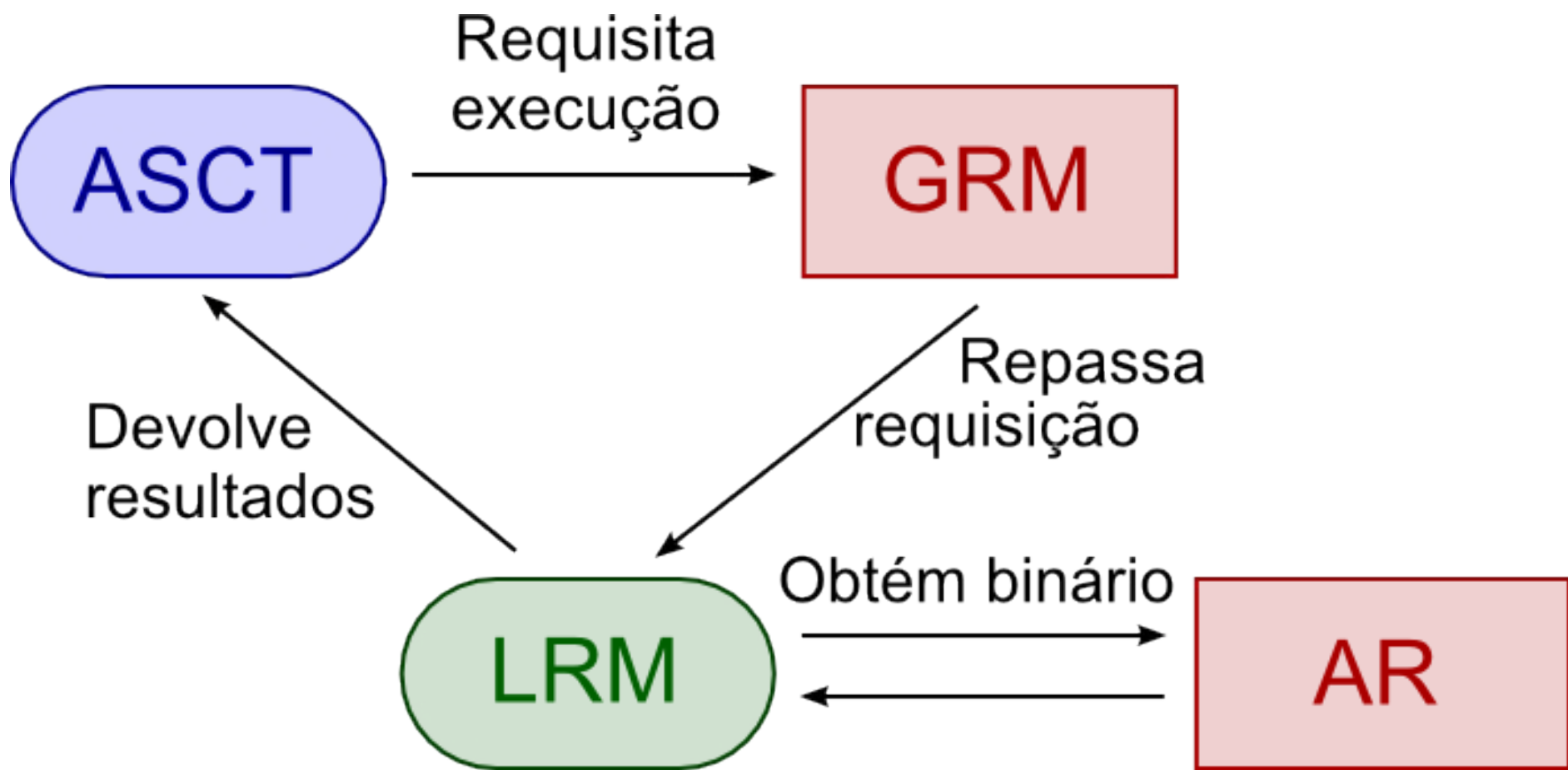
Done

# Federação de Aglomerados

- Federação de aglomerados
  - Aglomerados conectados em estrutura hierárquica
- Aglomerados possuem informações aproximadas sobre recursos disponíveis em aglomerados filhos



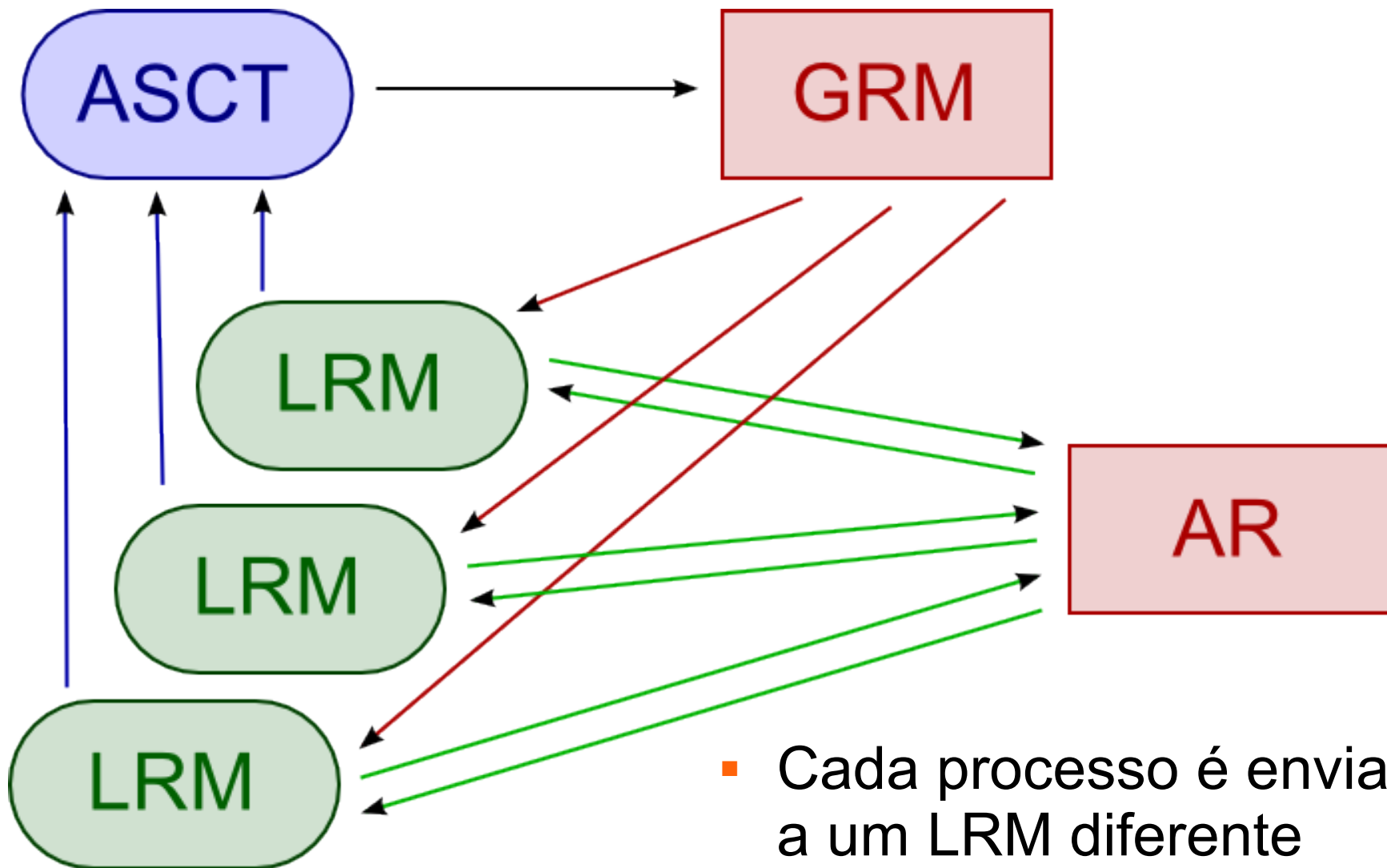
# Execução de Aplicações



# Execução de Aplicações

1. Usuário submete aplicações através do Portal
  - Pode ser executado a partir de qualquer máquina
2. GRM determina em quais máquinas a aplicação será executada (escalonamento)
3. Requisição é repassada para as máquinas selecionadas
4. Máquinas obtêm dados de entrada e os binários da aplicação e realizam sua execução
5. Após o término da execução, arquivos de saída são enviados à máquina executando o portal

# Execução de Aplicações Paralelas



- Aplicações executadas durante ciclos ociosos de máquinas não-dedicadas
  - Podem ficar indisponíveis ou mudar de ociosa para ocupada inesperadamente
  - Execução da aplicação é comprometida
    - É preciso reiniciar a execução do início
  
- Mecanismo de tolerância a falhas
  - Permite reiniciar a aplicação de um ponto intermediário de sua execução

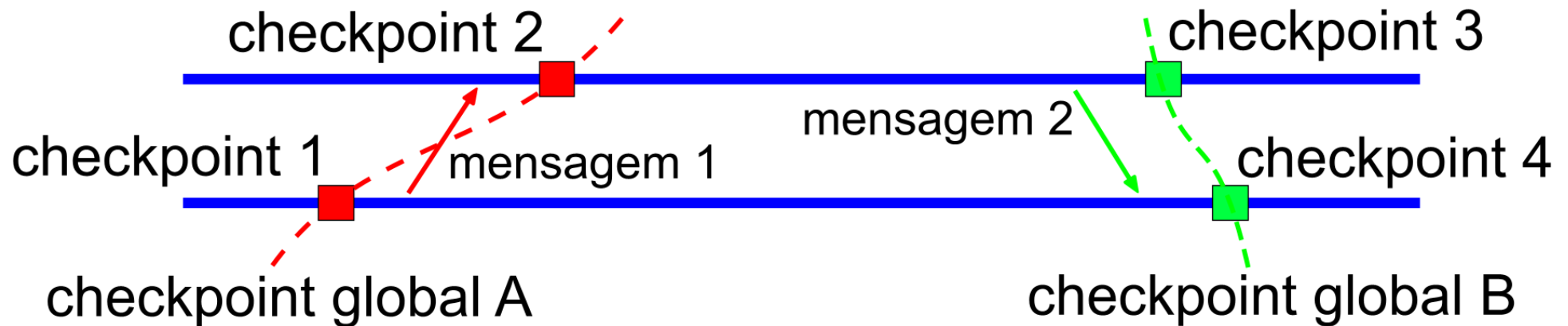
- Recuperação por retrocesso baseada em *checkpointing*
  - *Checkpoints* contendo o estado da aplicações são gerados periodicamente
  - *Checkpoints* são armazenados nas máquinas do aglomerado
  - Em caso de falha durante execução, aplicação é reiniciada a partir do último checkpoint gerado

- O estado de uma aplicação pode ser obtido de diferentes modos
- Checkpointing no nível do sistema
  - Dados obtidos diretamente do espaço de memória
    - Transparente à aplicação
    - Requer uso de máquinas homogêneas
- Checkpointing no nível da aplicação
  - Aplicação fornece dados a serem salvos
    - Checkpoints portáteis
    - Precisa modificar código-fonte da aplicação

# Checkpointing de Aplicações Paralelas

- Checkpoints locais → checkpoint global
- Aplicações paralelas desacopladas
  - Aplicações bag-of-tasks
  - Basta gerar um checkpoint para cada processo
- Aplicações paralelas acopladas
  - Aplicações BSP e MPI
  - Dependências entre processos
  - É preciso coordenar a criação de checkpoints locais

# Checkpointing de Aplicações Paralelas

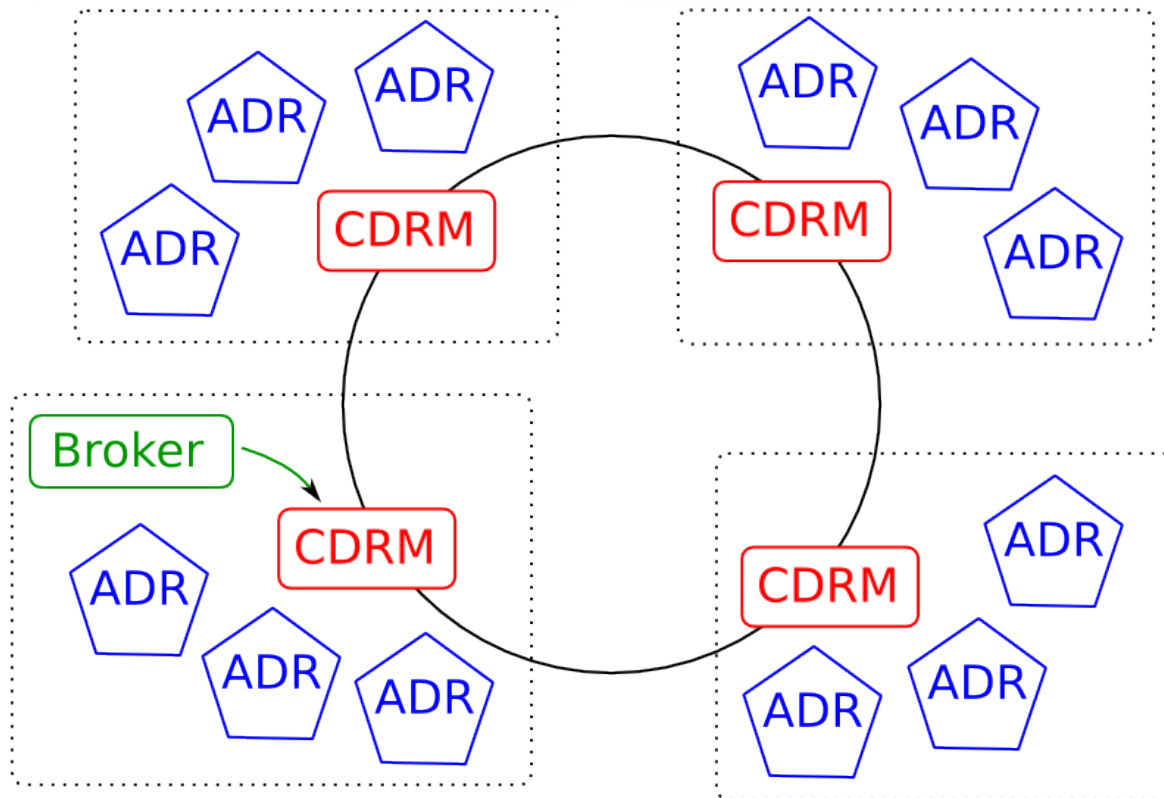


- Checkpoint 1 é *inconsistente* e 2 é consistente
- Protocolo coordenado de checkpointing
  - Sincroniza os processos antes de gerar checkpoint
  - Deste modo os checkpoints gerados são sempre consistentes

- Aplicações da grade podem utilizar ou produzir grandes quantidades de dados
  - Dados de aplicações podem ser compartilhados
- Abordagem tradicional: servidores dedicados
- Grade oportunista
  - Máquinas com grandes quantidade de espaço livre
  - Ambiente altamente dinâmico
    - Composto por dezenas de milhares de máquinas
    - Utilizar somente períodos ociosos
    - Máquina entram e saem do sistema continuamente

- Middleware que permite o armazenamento distribuído de dados da grade
  - Provê armazenamento confiável e eficiente
  - Utiliza o espaço livre de máquinas compartilhadas
- Organizado como federação de aglomerados
  - Similar à maioria das grades oportunistas
    - Aglomerados do OppStore são mapeados em aglomerados da grade oportunista
  - Facilita o gerenciamento do dinamismo do sistema
  - Aglomerados conectados por uma rede *peer-to-peer*

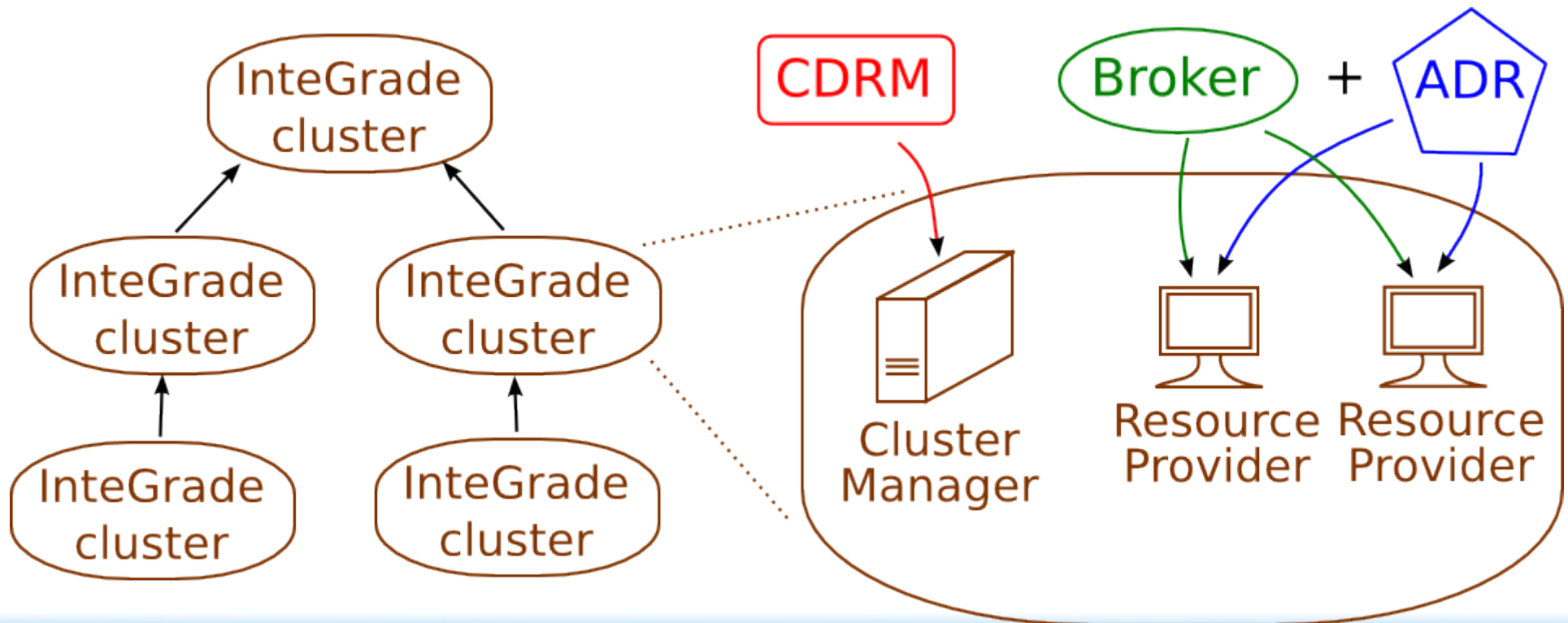
# Arquitetura do OppStore



- Repositório Autônomo de Dados (ADR)
- Gerenciador de Repositórios de Dados (CDRM)
- Intermediador de Acesso (Access Broker)

- Arquivos codificados em fragmentos redundantes
  - Fragmentos armazenados em aglomerados distintos
- Vantagens
  - Maior tolerância a falhas
  - Bom desempenho, pois fragmentos são transferidos em paralelo

- CDRM → Máquina gerenciadora do aglomerado
- ADRs e Brokers → Provedores de recursos
- Interface com o InteGrade em desenvolvimento



- Estamos realizando a integração do InteGrade com o OppStore
  - Dados de entrada e saída de aplicações passarão a ser armazenados no sistema de armazenamento distribuído
  - Máquina de onde aplicação foi submetida deixa de ser um ponto único de falhas

- Atualmente na versão 0.4
- Execução em dois aglomerados do IME-USP e em aglomerados de outras universidades
- Portal permite realizar remotamente requisições de execução
- Permite execução de aplicações paralelas
  - MPI, BSP e *Bag-of-tasks*
- Precisam ser realizados mais testes
  - Para tal, precisamos de mais usuários

- Grades Computacionais
- Grades de Dados
- Grades Oportunistas
- InteGrade
- Protocolo de execução
- Armazenamento distribuído de dados

- Sítio do InteGrade
  - <http://www.integrade.org.br>
- Suporte do InteGrade
  - [integrade-support@integrade.org.br](mailto:integrade-support@integrade.org.br)