

Avanços e Perspectivas do Projeto Integrate na UFMA

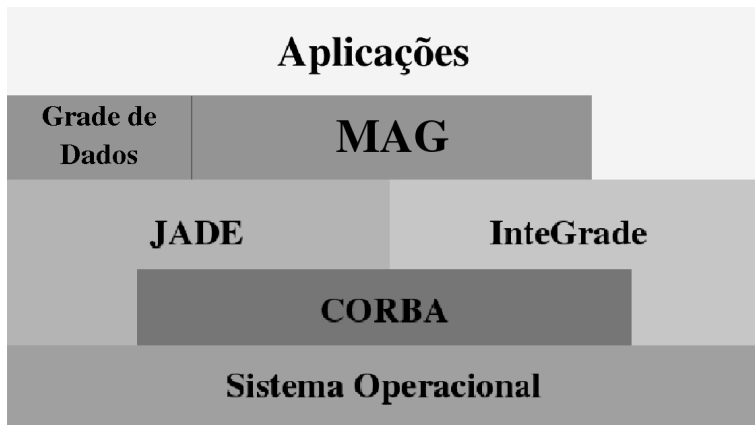
Francisco José da Silva e Silva

Universidade Federal do Maranhão - UFMA
Departamento de Informática
Laboratório de Sistemas Distribuídos - LSD

Agosto de 2006



Arquitetura em Camadas do MAG/Integrade

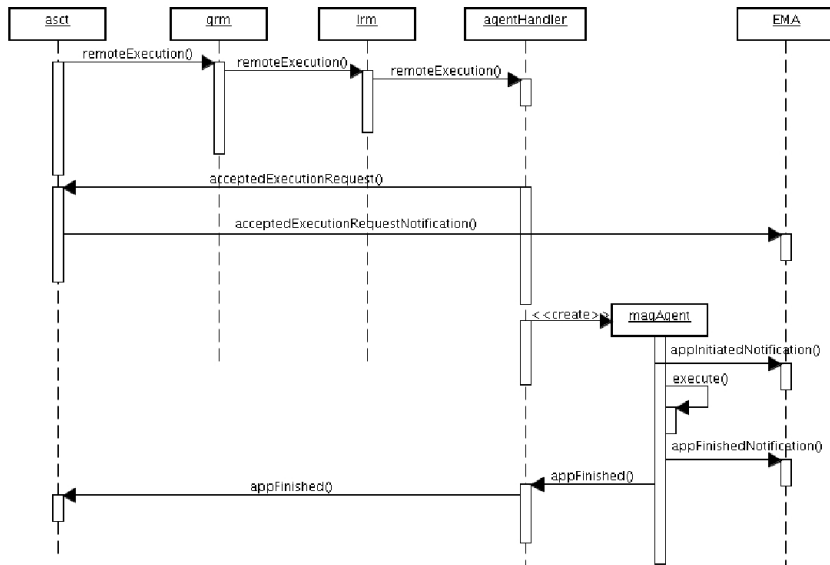


Funcionalidades do MAG

- Execução de aplicações através de agentes de software;
- Gerenciamento da execução de aplicações na grade;
- Migração forte de agentes de software;
- Serviço de metadados e dados;
- Mecanismos de tolerância a falhas de aplicações;
- Ferramentas de auxílio à mobilidade do usuário.



Execução de Aplicação



Atividades em Andamento

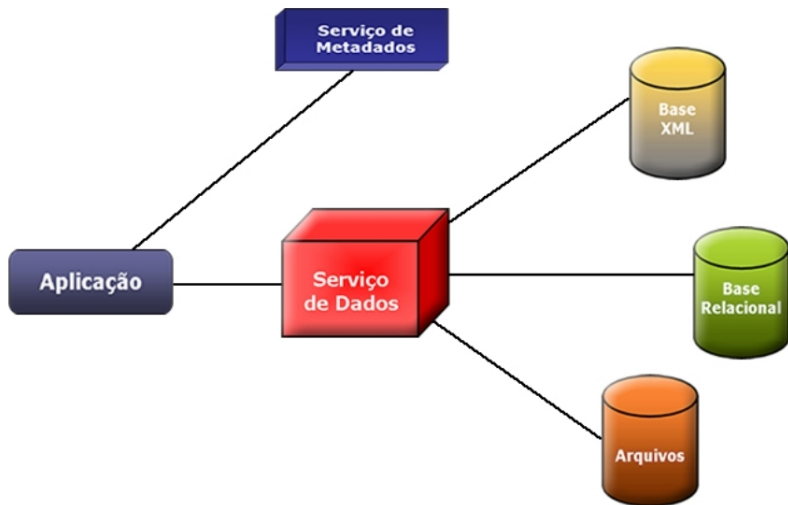
- Defesa do Bysmarck (julho de 2006);
- Mecanismo de autenticação de usuários;
- Nova versão do MAG compatível com o Integrade 0.3;
- Mecanismo de checkpointing compatível com Java 1.5;
- Melhorias no serviço de dados/metadados;
- Mecanismo de tolerância a falhas flexível;
- Mecanismos de suporte à mobilidade do usuário.



Serviço de Metadados



Serviços Básicos de Grades de Dados



Serviço de Metadados

- Permite descrever arquivos, coleções e visões;
- Esquema extensível;
- Linguagem de consulta própria;
- Modelo de dados independente do sistema de armazenamento;
- Implementado como uma sociedade de agentes;
- Possui interface CORBA.



Serviço de Dados

- Atualmente implementado através de um único agente que funciona como um repositório de arquivos;
- Transferência de arquivos através de Java NIO;
- Trabalho de mestrado na UFG.



Mobilidade do Usuário



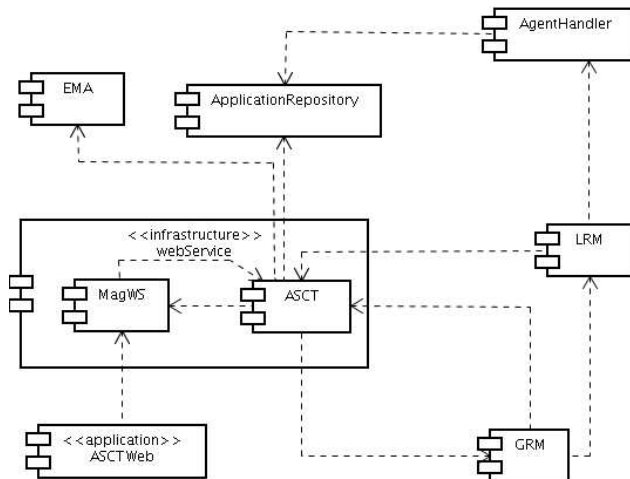
Mobilidade do Usuário no MAG/Integrade

A mobilidade do usuário é baseada em dois mecanismos:

- Acesso aos serviços da grade utilizando a Internet (Serviços Web): computação nômade;
- Acesso aos serviços da grade utilizando dispositivos móveis (plataforma PalmOS): computação móvel.



Infraestrutura de Suporte à Computação Nômade

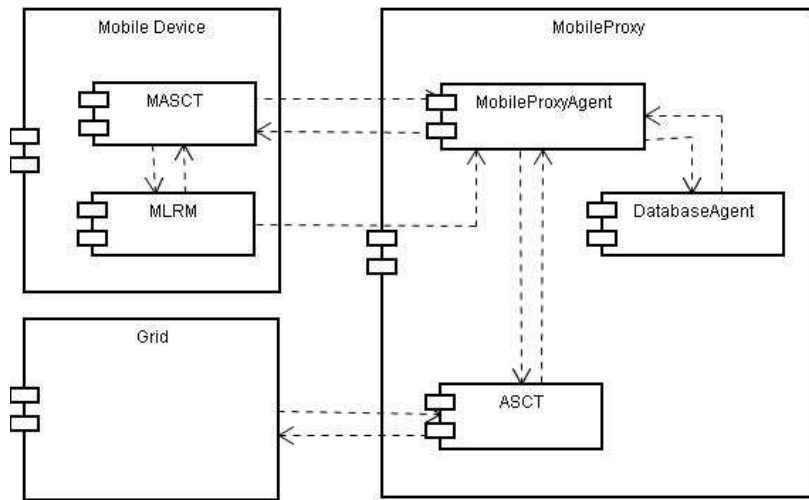


Suporte à Computação Móvel: Características da Solução

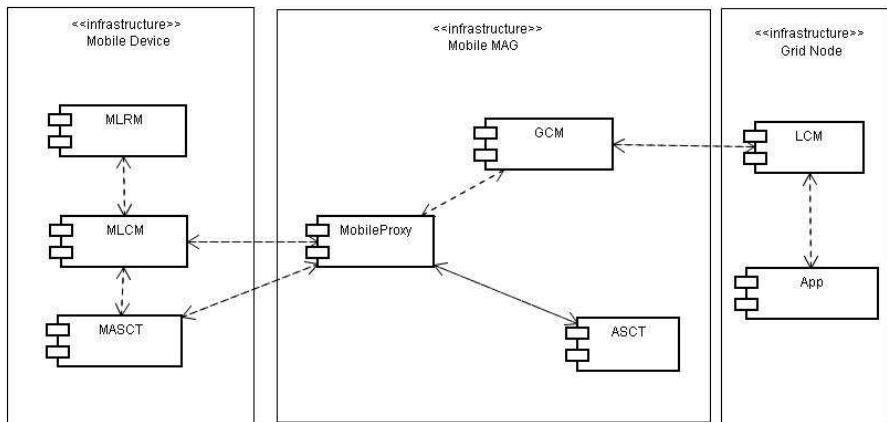
- Utilização de redes sem-fio e dispositivos móveis, tornando a grade uma extensão dos recursos dos dispositivos;
- Deve tratar questões como:
 - ▶ Baixa largura de banda;
 - ▶ Conectividade intermitente;
 - ▶ Heterogeneidade de dispositivos;
 - ▶ Carência de recursos computacionais de dispositivos móveis.



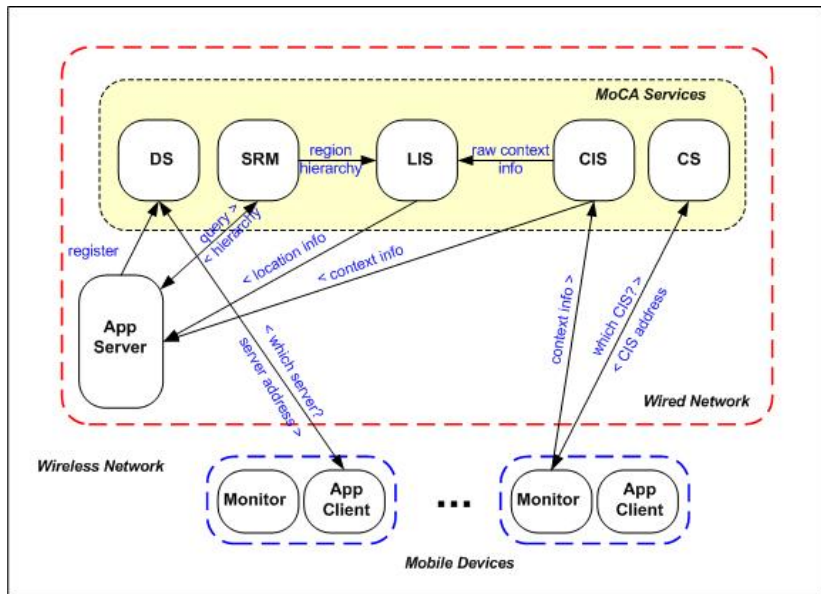
Infraestrutura de suporte à Computação Móvel



Infraestrutura para Armazenamento de Informações de Contexto



Arquitetura do MoCA (Mobile Collaboration Architecture)

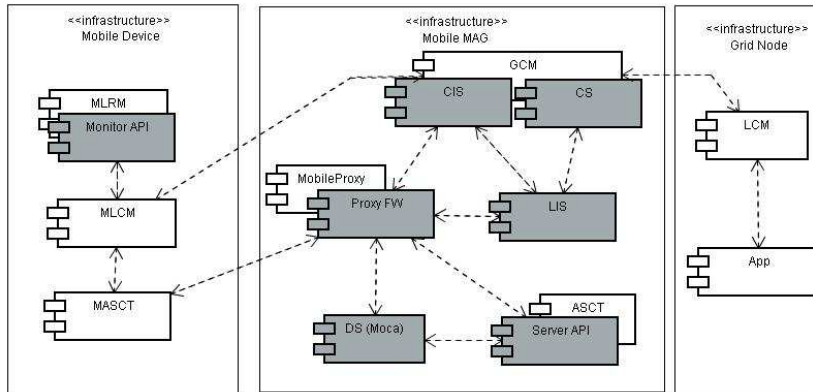


Componentes do MoCA (Mobile Collaboration Architecture)

- Client and server APIs;
- Monitor;
- Configuration Service (CS);
- Discovery Service (DS);
- Context Information Service (CIS);
- Location Inference Service (LIS);
- Symbolic Region Manager (SRM);
- Proxy Framework for Customized Adaptation.



Algumas Possibilidades de Integração com o MOCA



Tolerância a Falhas de Aplicações



Motivação

- Grades são mais propensas a falhas que plataformas de computação tradicionais já que agregam potencialmente milhares de recursos, serviços e aplicações que necessitam interagir através de diferentes domínios administrativos;
- Os nós da grande são instáveis, já que em geral não compreendem um ambiente controlado como ocorre com máquinas paralelas e aglomerados de computadores. O problema é amplificado em grades oportunistas;
- Longas aplicações podem requerer dias para serem computadas.



Técnicas Básicas para Tratamento de Falhas de Aplicação

- *Retrying*
 - ▶ Em caso de falha, a tarefa deve ser iniciada do princípio de sua execução.
- *Checkpointing*
 - ▶ A tarefa armazena periodicamente seu *checkpoint*. Em caso de falha, ela pode ser iniciada a partir do seu *checkpoint* mais recente. Técnica utilizada originalmente no MAG/Integrade.
- *Replication*
 - ▶ Executa diversas cópias de uma aplicação em recursos distintos, com os mesmos parâmetros. Se uma cópia falhar, a técnica de recuperação a ser adotada é *retrying*.
- *Replication with checkpointing*
 - ▶ Se uma cópia falhar, a técnica de recuperação a ser adotada é *checkpointing*.



Tratamento de Falhas

- Hwang e Kesselman, no artigo *“Grid Workflow: A Flexible Failure Handling Framework for the Grid”*, realizaram simulações com o objetivo de investigar as vantagens de se utilizar diversas técnicas de tratamento de falhas no ambiente de Grades;
- Parâmetros utilizados na simulação:
 - ▶ **Tempo de execução livre de falhas (F)**: Tempo de execução de uma tarefa ausente de falhas;
 - ▶ **Taxa de falhas (λ)**: Frequência de ocorrência de falhas;
 - ▶ **Mean Time to Failure (MTTF) = $\frac{1}{\lambda}$** : Intervalo de tempo médio entre a ocorrência de falhas adjacentes;
 - ▶ **Downtime**: Intervalo de tempo desde a falha de uma tarefa até o seu correto restabelecimento.



Conclusões de Hwang e Kesselman

- Quando a taxa de falhas (λ) é alta (baixo valor do MTTF), *checkpointing* e *replicação com checkpointing* são as duas melhores técnicas;
- Quando a taxa de falhas (λ) é baixa (alto valor do MTTF), as técnicas *checkpointing* e *replicação com checkpointing* são inapropriadas devido ao alto custo do *checkpoint* periódico;
- Se o ambiente de execução é relativamente confiável ($\frac{MTTF}{F} > 0.6$), então a *replicação* é melhor que as outras técnicas;
- Quando o ambiente de execução tem um elevado *downtime*, *replicação* e *replicação com checkpointing* são as duas melhores técnicas.



Conclusão Geral

- Devido a natureza genérica, dinâmica e heterogênea das Grades de computadores, é vantajoso oferecer flexibilidade no mecanismo de recuperação de falhas.



Recuperação de Falhas Flexível no Integrate

Nosso objetivo: tornar flexível o mecanismo de tolerância a falhas do projeto Integrate2 em duas etapas:

- 1 Permitir ao usuário selecionar o mecanismo de recuperação de falhas a ser empregado ao submeter uma solicitação para execução de aplicação;
- 2 Em segundo momento, a Grade faria a escolha do mecanismo de recuperação de falhas a ser adotado na execução de uma aplicação baseada nos dados armazenados pelo mecanismo de gerenciamento da execução de aplicações.



Arquitetura Proposta para o Mecanismo de Replicação

- Atualmente, o Integrade possui um mecanismo de *checkpointing*, porém, o de *replicação* ainda não é oferecido.
- Dois componentes seriam adicionados no Integrade:
 - ▶ CRM (*Cluster Replication Manager*)
 - ▶ ERM (*Execution Replication Manager*)



CRM - *Cluster Replication Manager*

Funcionalidades:

- Trata as requisições de execução com réplicas;
- Torna transparente a existência das réplicas ao EM (*ExecutionManager*);
- Instancia um ERM para cada requisição de execução com réplicas.



ERM - *Execution Replication Manager*

Funcionalidades:

- Mapeamento dos LRMs responsáveis pelas execuções das réplicas;
- Quando um réplica termina, requisita o término da execução das demais;
- Informa ao ASCT se a requisição com réplicas foi aceita, recusada ou finalizada;
- Obtêm os arquivos de entrada através do ASCT;
- Obtêm os arquivos de saída armazenados no LRM;
- Do ponto de vista do ASCT e do EM, o ERM se apresenta como o LRM responsável pela execução da aplicação.



Diagrama de Sequência - Submissão de Aplicação com Réplicas

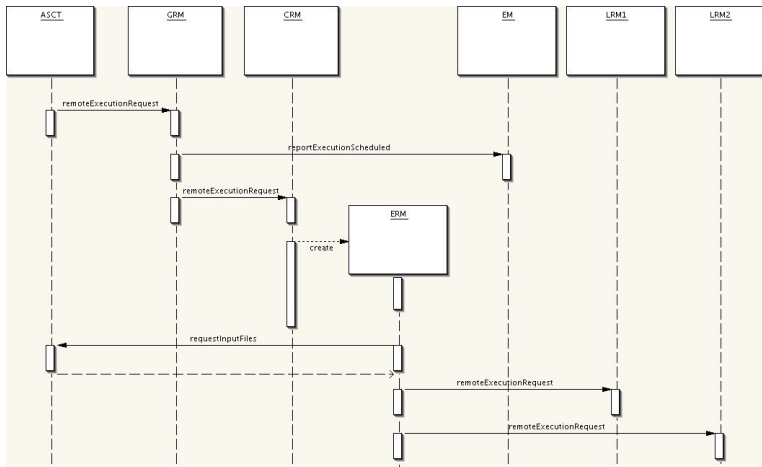
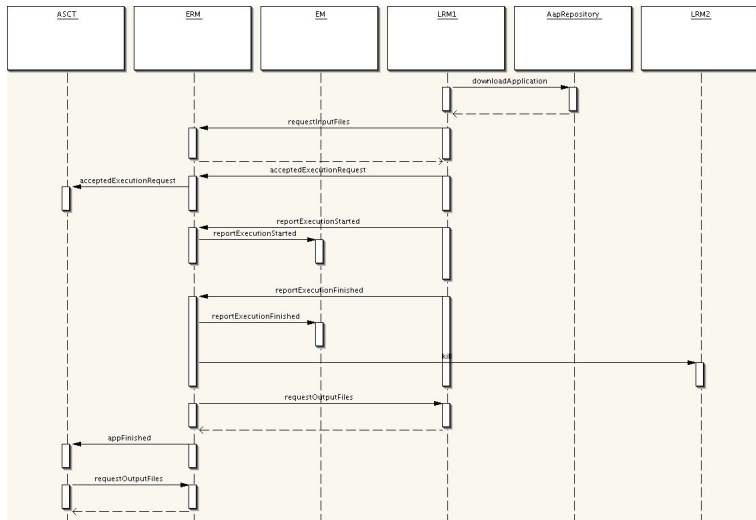


Diagrama de Sequência - Submissão de Aplicação com Réplicas



O Que Temos Implementado

- O ASCT já permite a submissão de aplicação utilizando-se replicação;
- Alterações no GRM para tratar as requisições de execução com réplicas;
- Alterações no LRM: mensagens enviadas para o EM são agora enviadas para o ERM;
- CRM e ERM implementados.



Atividades em Andamento

- Realização de experimentos:
 - ▶ Objetivo: avaliar, no Integrate, os benefícios do uso do mecanismo de tolerância a falhas flexível observados por Hwang e Kesselman em suas simulações.
- Desenvolvimento do mecanismo de replicação para aplicações BSP.

