

Gerenciamento de Recursos para Grades Computacionais *Node Control Center*

Carlos Eduardo Moreira dos Santos
Orientador: Fabio Kon

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

Trabalho de Formatura Supervisionado, 2009

Visão Geral

- 1 **Introdução**
 - InteGrade
 - Objetivo
- 2 **Programas Utilizados**
 - CPUReserve
 - NCC
 - InteGrade LRM
- 3 **Integração**
- 4 **Resultados e Contribuições**
 - Limitação de CPU
 - InteGrade

Tópicos

- 1 **Introdução**
 - **InteGrade**
 - Objetivo
- 2 **Programas Utilizados**
 - CPUReserve
 - NCC
 - InteGrade LRM
- 3 **Integração**
- 4 **Resultados e Contribuições**
 - Limitação de CPU
 - InteGrade

InteGrade

- Um dos projetos oficiais do CCSL
- Financiado pelo CNPq
- *Middleware* para grades computacionais oportunistas
- Soma recursos
- Comunicação entre processos
- Consome a ociosidade de computadores
- Máquinas dedicadas

InteGrade

- Um dos projetos oficiais do CCSL
- Financiado pelo CNPq
- *Middleware* para grades computacionais oportunistas
- Soma recursos
- Comunicação entre processos
- Consome a ociosidade de computadores
- Máquinas dedicadas

InteGrade

- Um dos projetos oficiais do CCSL
- Financiado pelo CNPq
- *Middleware* para grades computacionais oportunistas
- Soma recursos
- Comunicação entre processos
- Consome a ociosidade de computadores
- Máquinas dedicadas

InteGrade

- Um dos projetos oficiais do CCSL
- Financiado pelo CNPq
- *Middleware* para grades computacionais oportunistas
- Soma recursos
- Comunicação entre processos
- Consome a ociosidade de computadores
- Máquinas dedicadas

InteGrade

- Um dos projetos oficiais do CCSL
- Financiado pelo CNPq
- *Middleware* para grades computacionais oportunistas
- Soma recursos
- Comunicação entre processos
- Consome a ociosidade de computadores
- Máquinas dedicadas

InteGrade

- Um dos projetos oficiais do CCSL
- Financiado pelo CNPq
- *Middleware* para grades computacionais oportunistas
- Soma recursos
- Comunicação entre processos
- Consome a ociosidade de computadores
- Máquinas dedicadas

InteGrade

- Um dos projetos oficiais do CCSL
- Financiado pelo CNPq
- *Middleware* para grades computacionais oportunistas
- Soma recursos
- Comunicação entre processos
- Consome a ociosidade de computadores
- Máquinas dedicadas

Tópicos

- 1 **Introdução**
 - InteGrade
 - **Objetivo**
- 2 **Programas Utilizados**
 - CPUReserve
 - NCC
 - InteGrade LRM
- 3 **Integração**
- 4 **Resultados e Contribuições**
 - Limitação de CPU
 - InteGrade

Objetivo

- Minimizar impacto de aplicações da grade nos nós
- Possibilidade de o usuário limitar recursos compartilhados
- Interface descomplicada para leigos

Objetivo

- Minimizar impacto de aplicações da grade nos nós
- Possibilidade de o usuário limitar recursos compartilhados
- Interface descomplicada para leigos

Objetivo

- Minimizar impacto de aplicações da grade nos nós
- Possibilidade de o usuário limitar recursos compartilhados
- Interface descomplicada para leigos

Tópicos

- 1 Introdução
 - InteGrade
 - Objetivo
- 2 **Programas Utilizados**
 - **CPUReserve**
 - NCC
 - InteGrade LRM
- 3 Integração
- 4 Resultados e Contribuições
 - Limitação de CPU
 - InteGrade

CPUReserve - Introdução

- Integrantes do Projeto InteGrade na PUC-Rio
- Limita o uso de CPU em nível de usuário
 - Sem a sobrecarga das máquinas virtuais
 - Não é necessário alterar o *kernel*
- Escrito em C, é leve e eficiente

CPUReserve - Introdução

- Integrantes do Projeto InteGrade na PUC-Rio
- Limita o uso de CPU em nível de usuário
 - Sem a sobrecarga das máquinas virtuais
 - Não é necessário alterar o *kernel*
- Escrito em C, é leve e eficiente

CPUReserve - Introdução

- Integrantes do Projeto InteGrade na PUC-Rio
- Limita o uso de CPU em nível de usuário
 - Sem a sobrecarga das máquinas virtuais
 - Não é necessário alterar o *kernel*
- Escrito em C, é leve e eficiente

CPUReserve - Introdução

- Integrantes do Projeto InteGrade na PUC-Rio
- Limita o uso de CPU em nível de usuário
 - Sem a sobrecarga das máquinas virtuais
 - Não é necessário alterar o *kernel*
- Escrito em C, é leve e eficiente

CPUReserve - Introdução

- Integrantes do Projeto InteGrade na PUC-Rio
- Limita o uso de CPU em nível de usuário
 - Sem a sobrecarga das máquinas virtuais
 - Não é necessário alterar o *kernel*
- Escrito em C, é leve e eficiente

CPUReserve - Funcionamento

- Tempo de execução obtido em */proc*
- `kill(pid_t pid, SIGSTOP)`
- `kill(pid_t pid, SIGCONT)`
- `sched_getaffinity(pid_t pid, size_t cpusetsize, cpu_set_t *mask)`

CPUReserve - Funcionamento

- Tempo de execução obtido em */proc*
- `kill(pid_t pid, SIGSTOP)`
- `kill(pid_t pid, SIGCONT)`
- `sched_getaffinity(pid_t pid, size_t cpusetsize, cpu_set_t *mask)`

CPUReserve - Funcionamento

- Tempo de execução obtido em */proc*
- `kill(pid_t pid, SIGSTOP)`
- `kill(pid_t pid, SIGCONT)`
- `sched_getaffinity(pid_t pid, size_t cpusetsize, cpu_set_t *mask)`

CPUReserve - Alterações

- Não há mais necessidade de ser *root*
 - Sem garantia no tempo de execução mínimo
 - Facilidade de instalação
 - Segurança
- Mudança do limite enquanto em execução
- Tempo de execução calculado totalmente no servidor, com testes
- *Status* informando programas e limites
- Suporte a múltiplos núcleos

CPUReserve - Alterações

- Não há mais necessidade de ser *root*
 - Sem garantia no tempo de execução mínimo
 - Facilidade de instalação
 - Segurança
- Mudança do limite enquanto em execução
- Tempo de execução calculado totalmente no servidor, com testes
- *Status* informando programas e limites
- Suporte a múltiplos núcleos

CPUReserve - Alterações

- Não há mais necessidade de ser *root*
 - Sem garantia no tempo de execução mínimo
 - Facilidade de instalação
 - Segurança
- Mudança do limite enquanto em execução
- Tempo de execução calculado totalmente no servidor, com testes
- *Status* informando programas e limites
- Suporte a múltiplos núcleos

CPUReserve - Alterações

- Não há mais necessidade de ser *root*
 - Sem garantia no tempo de execução mínimo
 - Facilidade de instalação
 - Segurança
- Mudança do limite enquanto em execução
- Tempo de execução calculado totalmente no servidor, com testes
- *Status* informando programas e limites
- Suporte a múltiplos núcleos

CPUReserve - Alterações

- Não há mais necessidade de ser *root*
 - Sem garantia no tempo de execução mínimo
 - Facilidade de instalação
 - Segurança
- Mudança do limite enquanto em execução
- Tempo de execução calculado totalmente no servidor, com testes
- *Status* informando programas e limites
- Suporte a múltiplos núcleos

CPUReserve - Alterações

- Não há mais necessidade de ser *root*
 - Sem garantia no tempo de execução mínimo
 - Facilidade de instalação
 - Segurança
- Mudança do limite enquanto em execução
- Tempo de execução calculado totalmente no servidor, com testes
- *Status* informando programas e limites
- Suporte a múltiplos núcleos

CPUReserve - Alterações

- Não há mais necessidade de ser *root*
 - Sem garantia no tempo de execução mínimo
 - Facilidade de instalação
 - Segurança
- Mudança do limite enquanto em execução
- Tempo de execução calculado totalmente no servidor, com testes
- *Status* informando programas e limites
- Suporte a múltiplos núcleos

CPUReserve - Alterações

- Não há mais necessidade de ser *root*
 - Sem garantia no tempo de execução mínimo
 - Facilidade de instalação
 - Segurança
- Mudança do limite enquanto em execução
- Tempo de execução calculado totalmente no servidor, com testes
- *Status* informando programas e limites
- Suporte a múltiplos núcleos

CPUReserve - Distribuição do Tempo de CPU

Exemplo

Dado que o computador possui 4 núcleos e que o usuário limitou a CPU em 60%, como dividir 1000 ms entre as aplicações abaixo?

Aplicação	Tasks
a	2
b	1
c	3

CPUReserve - Distribuição do Tempo de CPU

Definições (tempos)

- **Ideal** = $(1000 \times \text{limite}) \times \min(\text{Tasks}, \text{núcleos})$
- **Disponível** = $(1000 \times \text{limite}) \times \text{núcleos} - \text{utilizados}$
- **PorAplicação** = $\frac{\text{Disponível}}{\text{aplicações restantes}}$
- **Fornecido** = $\min(\text{Ideal}, \text{PorAplicação})$

CPUReserve - Distribuição do Tempo de CPU

Definições (tempos)

- **Ideal** = $(1000 \times \text{limite}) \times \min(\text{Tasks}, \text{núcleos})$
- **Disponível** = $(1000 \times \text{limite}) \times \text{núcleos} - \text{utilizados}$
- **PorAplicação** = $\frac{\text{Disponível}}{\text{aplicações restantes}}$
- **Fornecido** = $\min(\text{Ideal}, \text{PorAplicação})$

CPUReserve - Distribuição do Tempo de CPU

Definições (tempos)

- **Ideal** = $(1000 \times \text{limite}) \times \min(\text{Tasks}, \text{núcleos})$
- **Disponível** = $(1000 \times \text{limite}) \times \text{núcleos} - \text{utilizados}$
- **PorAplicação** = $\frac{\text{Disponível}}{\text{aplicações restantes}}$
- **Fornecido** = $\min(\text{Ideal}, \text{PorAplicação})$

CPUReserve - Distribuição do Tempo de CPU

Definições (tempos)

- **Ideal** = $(1000 \times \text{limite}) \times \min(\text{Tasks}, \text{núcleos})$
- **Disponível** = $(1000 \times \text{limite}) \times \text{núcleos} - \text{utilizados}$
- **PorAplicação** = $\frac{\text{Disponível}}{\text{aplicações restantes}}$
- **Fornecido** = $\min(\text{Ideal}, \text{PorAplicação})$

CPUReserve - Distribuição do Tempo de CPU

Definições (tempos)

- **Ideal** = $(1000 \times \text{limite}) \times \min(\text{Tasks}, \text{núcleos})$
- **Disponível** = $(1000 \times \text{limite}) \times \text{núcleos} - \text{utilizados}$
- **PorAplicação** = $\frac{\text{Disponível}}{\text{aplicações restantes}}$
- **Fornecido** = $\min(\text{Ideal}, \text{PorAplicação})$

CPUReserve - Distribuição do Tempo de CPU

Aplicação	Tasks	Ideal	Disp.	PorApl.	Fornecido
a	2				
b	1				
c	5				

1 Ordenar por *Tasks*

2 Ideal = $600 \times \min(\text{Tasks}, 4)$

3 Laço

• Disponível = $600 \times 4 - \text{utilizados}$

• PorAplicação = $\frac{\text{Disponível}}{\text{aplicações restantes}}$

• Fornecido = $\min(\text{Ideal}, \text{PorAplicação})$

CPUReserve - Distribuição do Tempo de CPU

Aplicação	Tasks	Ideal	Disp.	PorApl.	Fornecido
b	1				
a	2				
c	5				

1 Ordenar por *Tasks*

2 Ideal = $600 \times \min(\text{Tasks}, 4)$

3 Laço

• Disponível = $600 \times 4 - \text{utilizados}$

• PorAplicação = $\frac{\text{Disponível}}{\text{aplicações restantes}}$

• Fornecido = $\min(\text{Ideal}, \text{PorAplicação})$

CPUReserve - Distribuição do Tempo de CPU

Aplicação	Tasks	Ideal	Disp.	PorApl.	Fornecido
b	1				
a	2				
c	5				

- 1 Ordenar por *Tasks*
- 2 Ideal = $600 \times \min(\text{Tasks}, 4)$
- 3 Laço

Disponível = $600 \times 4 - \text{utilizados}$

PorAplicação = $\frac{\text{Disponível}}{\text{aplicações restantes}}$

Fornecido = $\min(\text{Ideal}, \text{PorAplicação})$

CPUReserve - Distribuição do Tempo de CPU

Aplicação	Tasks	Ideal	Disp.	PorApl.	Fornecido
b	1	600			
a	2				
c	5				

- 1 Ordenar por *Tasks*
- 2 Ideal = $600 \times \min(\text{Tasks}, 4)$
- 3 Laço

Disponível = $600 \times 4 - \text{utilizados}$

PorAplicação = $\frac{\text{Disponível}}{\text{aplicações restantes}}$

Fornecido = $\min(\text{Ideal}, \text{PorAplicação})$

CPUReserve - Distribuição do Tempo de CPU

Aplicação	Tasks	Ideal	Disp.	PorApl.	Fornecido
b	1	600			
a	2	1200			
c	5				

1 Ordenar por *Tasks*

2 Ideal = $600 \times \min(\text{Tasks}, 4)$

3 Laço

Disponível = $600 \times 4 - \text{utilizados}$

PorAplicação = $\frac{\text{Disponível}}{\text{aplicações restantes}}$

Fornecido = $\min(\text{Ideal}, \text{PorAplicação})$

CPUReserve - Distribuição do Tempo de CPU

Aplicação	Tasks	Ideal	Disp.	PorApl.	Fornecido
b	1	600			
a	2	1200			
c	5	2400			

- 1 Ordenar por *Tasks*
- 2 Ideal = $600 \times \min(\text{Tasks}, 4)$
- 3 Laço

Disponível = $600 \times 4 - \text{utilizados}$

PorAplicação = $\frac{\text{Disponível}}{\text{aplicações restantes}}$

Fornecido = $\min(\text{Ideal}, \text{PorAplicação})$

CPUReserve - Distribuição do Tempo de CPU

Aplicação	Tasks	Ideal	Disp.	PorApl.	Fornecido
b	1	600			
a	2	1200			
c	5	2400			

- 1 Ordenar por *Tasks*
- 2 Ideal = $600 \times \min(\text{Tasks}, 4)$
- 3 Laço
 - 1 Disponível = $600 \times 4 - \text{utilizados}$
 - 2 PorAplicação = $\frac{\text{Disponível}}{\text{aplicações restantes}}$
 - 3 Fornecido = $\min(\text{Ideal}, \text{PorAplicação})$

CPUReserve - Distribuição do Tempo de CPU

Aplicação	Tasks	Ideal	Disp.	PorApl.	Fornecido
b	1	600	2400		
a	2	1200			
c	5	2400			

- 1 Ordenar por *Tasks*
- 2 Ideal = $600 \times \min(\text{Tasks}, 4)$
- 3 Laço
 - 1 Disponível = $600 \times 4 - \text{utilizados}$
 - 2 PorAplicação = $\frac{\text{Disponível}}{\text{aplicações restantes}}$
 - 3 Fornecido = $\min(\text{Ideal}, \text{PorAplicação})$

CPUReserve - Distribuição do Tempo de CPU

Aplicação	Tasks	Ideal	Disp.	PorApl.	Fornecido
b	1	600	2400		
a	2	1200			
c	5	2400			

- 1 Ordenar por *Tasks*
- 2 Ideal = $600 \times \min(\text{Tasks}, 4)$
- 3 Laço
 - 1 Disponível = $600 \times 4 - \text{utilizados}$
 - 2 PorAplicação = $\frac{\text{Disponível}}{\text{aplicações restantes}}$
 - 3 Fornecido = $\min(\text{Ideal}, \text{PorAplicação})$

CPUReserve - Distribuição do Tempo de CPU

Aplicação	Tasks	Ideal	Disp.	PorApl.	Fornecido
b	1	600	2400	800	
a	2	1200			
c	5	2400			

- 1 Ordenar por *Tasks*
- 2 Ideal = $600 \times \min(\text{Tasks}, 4)$
- 3 Laço
 - 1 Disponível = $600 \times 4 - \text{utilizados}$
 - 2 PorAplicação = $\frac{\text{Disponível}}{\text{aplicações restantes}}$
 - 3 Fornecido = $\min(\text{Ideal}, \text{PorAplicação})$

CPUReserve - Distribuição do Tempo de CPU

Aplicação	Tasks	Ideal	Disp.	PorApl.	Fornecido
b	1	600	2400	800	
a	2	1200			
c	5	2400			

- 1 Ordenar por *Tasks*
- 2 Ideal = $600 \times \min(\text{Tasks}, 4)$
- 3 Laço
 - 1 Disponível = $600 \times 4 - \text{utilizados}$
 - 2 PorAplicação = $\frac{\text{Disponível}}{\text{aplicações restantes}}$
 - 3 Fornecido = $\min(\text{Ideal}, \text{PorAplicação})$

CPUReserve - Distribuição do Tempo de CPU

Aplicação	Tasks	Ideal	Disp.	PorApl.	Fornecido
b	1	600	2400	800	600
a	2	1200			
c	5	2400			

- 1 Ordenar por *Tasks*
- 2 Ideal = $600 \times \min(\text{Tasks}, 4)$
- 3 Laço
 - 1 Disponível = $600 \times 4 - \text{utilizados}$
 - 2 PorAplicação = $\frac{\text{Disponível}}{\text{aplicações restantes}}$
 - 3 Fornecido = $\min(\text{Ideal}, \text{PorAplicação})$

CPUReserve - Distribuição do Tempo de CPU

Aplicação	Tasks	Ideal	Disp.	PorApl.	Fornecido
b	1	600	2400	800	600
a	2	1200	1800		
c	5	2400			

- 1 Ordenar por *Tasks*
- 2 Ideal = $600 \times \min(\text{Tasks}, 4)$
- 3 Laço
 - 1 Disponível = $600 \times 4 - \text{utilizados}$
 - 2 PorAplicação = $\frac{\text{Disponível}}{\text{aplicações restantes}}$
 - 3 Fornecido = $\min(\text{Ideal}, \text{PorAplicação})$

CPUReserve - Distribuição do Tempo de CPU

Aplicação	Tasks	Ideal	Disp.	PorApl.	Fornecido
b	1	600	2400	800	600
a	2	1200	1800	900	
c	5	2400			

- 1 Ordenar por *Tasks*
- 2 Ideal = $600 \times \min(\text{Tasks}, 4)$
- 3 Laço
 - 1 Disponível = $600 \times 4 - \text{utilizados}$
 - 2 PorAplicação = $\frac{\text{Disponível}}{\text{aplicações restantes}}$
 - 3 Fornecido = $\min(\text{Ideal}, \text{PorAplicação})$

CPUReserve - Distribuição do Tempo de CPU

Aplicação	Tasks	Ideal	Disp.	PorApl.	Fornecido
b	1	600	2400	800	600
a	2	1200	1800	900	900
c	5	2400			

- 1 Ordenar por *Tasks*
- 2 Ideal = $600 \times \min(\text{Tasks}, 4)$
- 3 Laço
 - 1 Disponível = $600 \times 4 - \text{utilizados}$
 - 2 PorAplicação = $\frac{\text{Disponível}}{\text{aplicações restantes}}$
 - 3 Fornecido = $\min(\text{Ideal}, \text{PorAplicação})$

CPUReserve - Distribuição do Tempo de CPU

Aplicação	Tasks	Ideal	Disp.	PorApl.	Fornecido
b	1	600	2400	800	600
a	2	1200	1800	900	900
c	5	2400			900

- 1 Ordenar por *Tasks*
- 2 Ideal = $600 \times \min(\text{Tasks}, 4)$
- 3 Laço
 - 1 Disponível = $600 \times 4 - \text{utilizados}$
 - 2 PorAplicação = $\frac{\text{Disponível}}{\text{aplicações restantes}}$
 - 3 Fornecido = $\min(\text{Ideal}, \text{PorAplicação})$

Exemplo de Uso

- Pode ser usado sem o InteGrade
- Disponível em <http://launchpad.net/cpureserve>
- Servidor:
 - `server 8000 50`
- Cliente:
 - `client 127.0.0.1:8000 prog arg1...argn`
- Adaptação:
 - `adapt 127.0.0.1:8000 12 (22 + 23) 75`
- *Status*:
 - `adapt 127.0.0.1:8000 status`

Exemplo de Uso

- Pode ser usado sem o InteGrade
- Disponível em <http://launchpad.net/cpureserve>

- Servidor:

- `server 8000 50`

- Cliente:

- `client 127.0.0.1:8000 prog arg1...argn`

- Adaptação:

- `adapt 127.0.0.1:8000 12 (22 + 23) 75`

- *Status*:

- `adapt 127.0.0.1:8000 status`

Exemplo de Uso

- Pode ser usado sem o InteGrade
- Disponível em <http://launchpad.net/cpureserve>
- Servidor:
 - `server 8000 50`
- Cliente:
 - `client 127.0.0.1:8000 prog arg1...argn`
- Adaptação:
 - `adapt 127.0.0.1:8000 12 (22 + 23) 75`
- *Status*:
 - `adapt 127.0.0.1:8000 status`

Exemplo de Uso

- Pode ser usado sem o InteGrade
- Disponível em `http://launchpad.net/cpureserve`
- Servidor:
 - `server 8000 50`
- Cliente:
 - `client 127.0.0.1:8000 prog arg1...argn`
- Adaptação:
 - `adapt 127.0.0.1:8000 12 (22 + 23) 75`
- *Status*:
 - `adapt 127.0.0.1:8000 status`

Exemplo de Uso

- Pode ser usado sem o InteGrade
- Disponível em `http://launchpad.net/cpureserve`
- Servidor:
 - `server 8000 50`
- Cliente:
 - `client 127.0.0.1:8000 prog arg1...argn`
- Adaptação:
 - `adapt 127.0.0.1:8000 12 (22 + 23) 75`
- *Status*:
 - `adapt 127.0.0.1:8000 status`

Exemplo de Uso

- Pode ser usado sem o InteGrade
- Disponível em `http://launchpad.net/cpureserve`
- Servidor:
 - `server 8000 50`
- Cliente:
 - `client 127.0.0.1:8000 prog arg1...argn`
- Adaptação:
 - `adapt 127.0.0.1:8000 12 (22 + 23) 75`
- *Status*:
 - `adapt 127.0.0.1:8000 status`

Exemplo de Uso

- Pode ser usado sem o InteGrade
- Disponível em `http://launchpad.net/cpureserve`
- Servidor:
 - `server 8000 50`
- Cliente:
 - `client 127.0.0.1:8000 prog arg1...argn`
- Adaptação:
 - `adapt 127.0.0.1:8000 12 (22 + 23) 75`
- *Status*:
 - `adapt 127.0.0.1:8000 status`

Exemplo de Uso

- Pode ser usado sem o InteGrade
- Disponível em `http://launchpad.net/cpureserve`
- Servidor:
 - `server 8000 50`
- Cliente:
 - `client 127.0.0.1:8000 prog arg1...argn`
- Adaptação:
 - `adapt 127.0.0.1:8000 12 (22 + 23) 75`
- *Status:*
 - `adapt 127.0.0.1:8000 status`

Exemplo de Uso

- Pode ser usado sem o InteGrade
- Disponível em `http://launchpad.net/cpureserve`
- Servidor:
 - `server 8000 50`
- Cliente:
 - `client 127.0.0.1:8000 prog arg1...argn`
- Adaptação:
 - `adapt 127.0.0.1:8000 12 (22 + 23) 75`
- *Status*:
 - `adapt 127.0.0.1:8000 status`

Exemplo de Uso

- Pode ser usado sem o InteGrade
- Disponível em `http://launchpad.net/cpureserve`
- Servidor:
 - `server 8000 50`
- Cliente:
 - `client 127.0.0.1:8000 prog arg1...argn`
- Adaptação:
 - `adapt 127.0.0.1:8000 12 (22 + 23) 75`
- *Status*:
 - `adapt 127.0.0.1:8000 status`

Tópicos

- 1 Introdução
 - InteGrade
 - Objetivo
- 2 **Programas Utilizados**
 - CPUReserve
 - **NCC**
 - InteGrade LRM
- 3 Integração
- 4 Resultados e Contribuições
 - Limitação de CPU
 - InteGrade

NCC - Introdução

- Escrito para MAC 413/5715, 2003
- Linguagem C++
- *Templates* apenas com variáveis a serem substituídas
- Consome pouca memória do nó (menos de 2 MB)
- Multiplataforma (*sockets*)

NCC - Introdução

- Escrito para MAC 413/5715, 2003
- Linguagem C++
- *Templates* apenas com variáveis a serem substituídas
- Consome pouca memória do nó (menos de 2 MB)
- Multiplataforma (*sockets*)

NCC - Introdução

- Escrito para MAC 413/5715, 2003
- Linguagem C++
- *Templates* apenas com variáveis a serem substituídas
- Consome pouca memória do nó (menos de 2 MB)
- Multiplataforma (*sockets*)

NCC - Introdução

- Escrito para MAC 413/5715, 2003
- Linguagem C++
- *Templates* apenas com variáveis a serem substituídas
- Consome pouca memória do nó (menos de 2 MB)
- Multiplataforma (*sockets*)

NCC - Introdução

- Escrito para MAC 413/5715, 2003
- Linguagem C++
- *Templates* apenas com variáveis a serem substituídas
- Consome pouca memória do nó (menos de 2 MB)
- Multiplataforma (*sockets*)

NCC - Alterações

- **Antes**
 - Método extenso para substituição de variáveis
 - Valores de variáveis explícitos no código
- **Após**
 - Comunicação com CPUReserve (status e adapt)
 - Substituição de variáveis orientada a objetos

NCC - Alterações

- Antes
 - Método extenso para substituição de variáveis
 - Valores de variáveis explícitos no código
- Após
 - Comunicação com CPUReserve (status e adapt)
 - Substituição de variáveis orientada a objetos

NCC - Alterações

- Antes
 - Método extenso para substituição de variáveis
 - Valores de variáveis explícitos no código
- Após
 - Comunicação com CPUReserve (status e adapt)
 - Substituição de variáveis orientada a objetos

NCC - Alterações

- Antes
 - Método extenso para substituição de variáveis
 - Valores de variáveis explícitos no código
- Após
 - Comunicação com CPUReserve (*status* e *adapt*)
 - Substituição de variáveis orientada a objetos

NCC - Alterações

- Antes
 - Método extenso para substituição de variáveis
 - Valores de variáveis explícitos no código
- Após
 - Comunicação com CPUReserve (`status e adapt`)
 - Substituição de variáveis orientada a objetos

NCC - Alterações

- Antes
 - Método extenso para substituição de variáveis
 - Valores de variáveis explícitos no código
- Após
 - Comunicação com CPUReserve (`status` e `adapt`)
 - Substituição de variáveis orientada a objetos

NCC - Exemplo

olamundo.ncc

```
<html>  
  <body bgcolor="#DED3E8">  
    <b><@MENSAGEM></b>  
  </body>  
</html>
```

NCC - Exemplo

OlaMundo.cpp

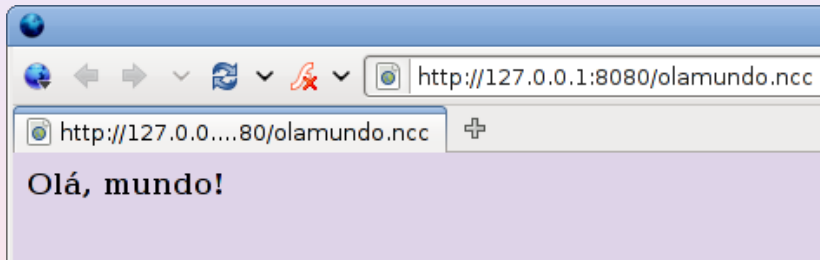
```
class OlaMundo : public WebPage {  
    public:  
        OlaMundo(string tpl): WebPage(tpl){}  
    private:  
        void generateVars() {  
            setVar("MENSAGEM", "Olá, mundo!");  
        }  
};
```

NCC - Exemplo

ncc.cpp

```
// ...  
else if (strcmp(file, "olamundo.ncc")==0) {  
    OlaMundo pagina(tpl);  
    tpl = pagina.getPage();  
}  
// ...
```

NCC - Exemplo



NCC - Resultado



→ 5% de CPU, um núcleo apenas
→ 40%, em cada núcleo
→ 80%

Resource Manager

CPU	
Limit	<input type="text" value="60"/> %
Cores	<input checked="" type="checkbox"/> 0 <input checked="" type="checkbox"/> 1
<input type="button" value="OK"/>	

Processes & Resource Usage

Application #	Process ID	Name	CPU (%)	Memory (kB)	Action
1	26293	matrix	30	380	<input type="button" value="Migrate"/>
2	26310	gridCalc	15	388	<input type="button" value="Migrate"/>
	26311	gridCalc	15	164	

Tópicos

- 1 Introdução
 - InteGrade
 - Objetivo
- 2 **Programas Utilizados**
 - CPUReserve
 - NCC
 - **InteGrade LRM**
- 3 Integração
- 4 Resultados e Contribuições
 - Limitação de CPU
 - InteGrade

InteGrade LRM

- Nós executam o *Local Resource Manager* (LRM)
- LRM utiliza *fork* e *execv*
- Chamada ao programa da grade foi prefixada

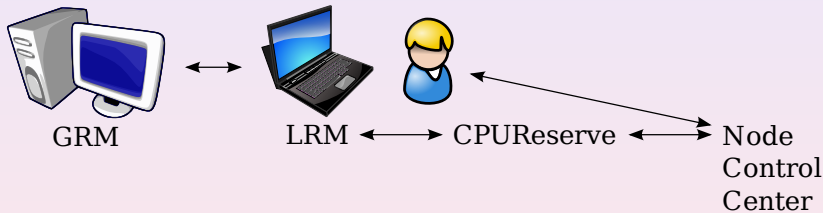
InteGrade LRM

- Nós executam o *Local Resource Manager* (LRM)
- LRM utiliza *fork* e *execv*
- Chamada ao programa da grade foi prefixada

InteGrade LRM

- Nós executam o *Local Resource Manager* (LRM)
- LRM utiliza *fork* e *execv*
- Chamada ao programa da grade foi prefixada

Relacionamento entre os Programas



Tópicos

- 1 Introdução
 - InteGrade
 - Objetivo
- 2 Programas Utilizados
 - CPUReserve
 - NCC
 - InteGrade LRM
- 3 Integração
- 4 Resultados e Contribuições
 - Limitação de CPU
 - InteGrade

Vantagens e Desvantagens ao Limitar CPU

- Vantagens

- TI Verde
 - Economia de energia
 - Menos calor gerado
 - Economia com refrigeração
- Maior autonomia de baterias
- Menos ruído para refrigeração
- Mais nós voluntários

- Desvantagens

- Menor quantidade de CPU disponível para a grade?
- Perda de desempenho para o usuário local?

Vantagens e Desvantagens ao Limitar CPU

- Vantagens

- TI Verde

- Economia de energia
 - Menos calor gerado
 - Economia com refrigeração
 - Maior autonomia de baterias
 - Menos ruído para refrigeração
 - Mais nós voluntários

- Desvantagens

- Menor quantidade de CPU disponível para a grade?
 - Perda de desempenho para o usuário local?

Vantagens e Desvantagens ao Limitar CPU

- Vantagens

- TI Verde

- Economia de energia
 - Menos calor gerado
 - Economia com refrigeração
 - Maior autonomia de baterias
 - Menos ruído para refrigeração
 - Mais nós voluntários

- Desvantagens

- Menor quantidade de CPU disponível para a grade?
 - Perda de desempenho para o usuário local?

Vantagens e Desvantagens ao Limitar CPU

- Vantagens

- TI Verde

- Economia de energia
 - Menos calor gerado
 - Economia com refrigeração
 - Maior autonomia de baterias
 - Menos ruído para refrigeração
 - Mais nós voluntários

- Desvantagens

- Menor quantidade de CPU disponível para a grade?
 - Perda de desempenho para o usuário local?

Vantagens e Desvantagens ao Limitar CPU

- Vantagens

- TI Verde

- Economia de energia
 - Menos calor gerado
 - Economia com refrigeração
 - Maior autonomia de baterias
 - Menos ruído para refrigeração
 - Mais nós voluntários

- Desvantagens

- Menor quantidade de CPU disponível para a grade?
 - Perda de desempenho para o usuário local?

Vantagens e Desvantagens ao Limitar CPU

- Vantagens

- TI Verde

- Economia de energia
 - Menos calor gerado
 - Economia com refrigeração

- Maior autonomia de baterias

- Menos ruído para refrigeração
 - Mais nós voluntários

- Desvantagens

- Menor quantidade de CPU disponível para a grade?
 - Perda de desempenho para o usuário local?

Vantagens e Desvantagens ao Limitar CPU

- Vantagens

- TI Verde
 - Economia de energia
 - Menos calor gerado
 - Economia com refrigeração
- Maior autonomia de baterias
- Menos ruído para refrigeração
- Mais nós voluntários

- Desvantagens

- Menor quantidade de CPU disponível para a grade?
- Perda de desempenho para o usuário local?

Vantagens e Desvantagens ao Limitar CPU

- Vantagens

- TI Verde
 - Economia de energia
 - Menos calor gerado
 - Economia com refrigeração
- Maior autonomia de baterias
- Menos ruído para refrigeração
- Mais nós voluntários

- Desvantagens

- Menor quantidade de CPU disponível para a grade?
- Perda de desempenho para o usuário local?

Vantagens e Desvantagens ao Limitar CPU

- Vantagens

- TI Verde
 - Economia de energia
 - Menos calor gerado
 - Economia com refrigeração
- Maior autonomia de baterias
- Menos ruído para refrigeração
- Mais nós voluntários

- Desvantagens

- Menor quantidade de CPU disponível para a grade?
- Perda de desempenho para o usuário local?

Vantagens e Desvantagens ao Limitar CPU

- Vantagens

- TI Verde
 - Economia de energia
 - Menos calor gerado
 - Economia com refrigeração
- Maior autonomia de baterias
- Menos ruído para refrigeração
- Mais nós voluntários

- Desvantagens

- Menor quantidade de CPU disponível para a grade?
- Perda de desempenho para o usuário local?

Vantagens e Desvantagens ao Limitar CPU

- Vantagens

- TI Verde
 - Economia de energia
 - Menos calor gerado
 - Economia com refrigeração
- Maior autonomia de baterias
- Menos ruído para refrigeração
- Mais nós voluntários

- Desvantagens

- Menor quantidade de CPU disponível para a grade?
- Perda de desempenho para o usuário local?

Resultados Experimentais

Casos

- 1 Sem aplicações da grade
- 2 Uma aplicação com um processo e outra com dois a 20%
- 3 Uma aplicação com um processo e outra com dois a 90%

	Média	Desvio Padrão
1	8,65	1,35
2	8,43	0,94
3	8,17	1,56

Tabela: Tempo para carregar Firefox

Resultados Experimentais

Casos

- 1 Sem aplicações da grade
- 2 Uma aplicação com um processo e outra com dois a 20%
- 3 Uma aplicação com um processo e outra com dois a 90%

	Média	Desvio Padrão
1	8,65	1,35
2	8,43	0,94
3	8,17	1,56

Tabela: Tempo para carregar Firefox

Resultados Experimentais

Casos

- 1 Sem aplicações da grade
- 2 Uma aplicação com um processo e outra com dois a 20%
- 3 Uma aplicação com um processo e outra com dois a 90%

	Média	Desvio Padrão
1	8,65	1,35
2	8,43	0,94
3	8,17	1,56

Tabela: Tempo para carregar Firefox

Resultados Experimentais

Casos

- 1 Sem aplicações da grade
- 2 Uma aplicação com um processo e outra com dois a 20%
- 3 Uma aplicação com um processo e outra com dois a 90%

	Média	Desvio Padrão
1	8,65	1,35
2	8,43	0,94
3	8,17	1,56

Tabela: Tempo para carregar Firefox

Resultados Experimentais

	Média	Desvio Padrão
1	18,28	1,50
2	18,72	0,68
3	18,86	0,90

Tabela: Tempo para carregar Openoffice

Tópicos

- 1 Introdução
 - InteGrade
 - Objetivo
- 2 Programas Utilizados
 - CPUReserve
 - NCC
 - InteGrade LRM
- 3 Integração
- 4 Resultados e Contribuições
 - Limitação de CPU
 - **InteGrade**

InteGrade

- Maior facilidade para limitação de mais recursos
- Melhor organização, separando os projetos

InteGrade

- Maior facilidade para limitação de mais recursos
- Melhor organização, separando os projetos

Links

- **Página oficial do Projeto InteGrade**
 - <http://www.integrate.org.br>
- **Página oficial do CPUReserve**
 - <https://mwlab.tecgraf.puc-rio.br/MWLab/projects/cpureserve>
- **Downloads e repositórios dos códigos deste trabalho**
 - <https://launchpad.net/integrate>
 - <https://launchpad.net/cpureserve>
 - <https://launchpad.net/ncc>

Artigos I



Goldchleger, Andrei

InteGrade: Um Sistema de Middleware para Computação em Grade Oportunista

Dissertação de mestrado na área de Ciência da Computação. IME-USP, 2004.



Reis, Valéria Q. e Cerqueira, Renato F. G.

A tool for isolating performance in general-purpose operating systems

MGC '08: Proceedings of the 6th international workshop on Middleware for grid computing, 2008.

Artigos II



Luz, Giulian D., Rogério e Santos, Guaraci
Interface para o NCC

*MAC 413/5715 - Tópicos (Avançados) de Programação
Orientada a Objetos, IME-USP, 2003.*