

# Unicluster: Uma Proposta de ICP para PAD

Rodrigo A. Real<sup>1</sup>, Mauro V. Real<sup>1</sup>, Nelson L. D. Filho<sup>1</sup>, Otávio R. Salvador<sup>2</sup>, Adenauer C. Yamin<sup>2</sup>

1 - Fundação Universidade Federal do Rio Grande - FURG  
{ec4real,dmtndf}@furg.br, mauro@dmc.furg.br

2 - Universidade Católica de Pelotas - UCPel  
otavio@freedom.ind.br, adenauer@atlas.ucpel.tche.br

## Introdução

A crescente disseminação da Internet e o aumento da velocidade de operação das redes e dos processadores, conduzem para a utilização destes recursos distribuídos como algo unificado, de acesso transparente e independente de localização. Esta nova perspectiva para *network computing* assume diferentes visões: *Internet computing*, *metacomputing*, *Grid Computing*, *scalable-computing*, e mais recentemente *Peer-to-Peer (P2P) computing*.

Por sua vez, a resolução de problemas numericamente intensivos é, em geral, computacionalmente dispendiosa, exigindo um elevado tempo de processamento. Muitos destes problemas apresentam características que facultam dividir o que deve ser processado em Unidades de Trabalho (UTs), podendo estas serem calculadas independentemente umas das outras. Alguns destes problemas podem ser classificados como de alta granulosidade, e deste modo possuem para cada UT, uma elevada razão entre o tempo empregado para o processamento e o tempo total necessário para sua instalação no processador remoto e o retorno dos resultados [CUL 99].

Objetivando reduzir o tempo para solução desse tipo de problemas, este trabalho apresenta uma proposta de ICP (*Internet Computing Platform*) composta por *servidores* distribuídos, alocando UTs, associados a um conjunto de *clientes* encarregados de realizar o processamento sobre as mesmas (vide figura 1). Nesta figura os servidores estão com diferentes cargas (representadas por triângulos). A topologia de interconexão dos servidores é determinada pelo administrador. O critério de visibilidade entre os servidores é a velocidade da interconexão de rede.

O Unicluster é produto do Projeto de Conclusão de Curso do autor no Curso de Engenharia de Computação da FURG. Neste artigo apresenta-se os principais aspectos da sua proposta, também são analisados os resultados atingidos com o uso do Unicluster na paralelização de uma aplicação de Engenharia Civil.

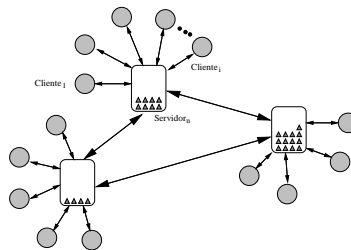


Figura 1: ORGANIZAÇÃO DISTRIBUÍDA DO UNICLUSTER

## Uniclust: Conceção

### Modelo

A arquitetura do Uniclust pode ser vista na figura 2, sua funcionalidade foi inspirada em trabalhos semelhantes ao SetiAtHome<sup>1</sup>. Os principais componentes que a integram são:

- *servidores* - compostos de dois fluxos de execução: (i) um dedicado a atender os pedidos de informações sobre o estado atual do sistema, e outro (ii) para atender às requisições de processamento e recebimento de resultados parciais;
- *clientes* - a proposta Uniclust contempla a existência de três tipos de programas clientes:
  - *processor clients*: clientes processadores que executam os algoritmos de interesse do usuário e trabalham sobre as UTs, retornando resultados ao final do processamento de cada uma;
  - *loader clients*: clientes que submetem aplicações e seus parâmetros para execução;
  - *status clients*: clientes que buscam informações sobre o estado do sistema (processamentos em andamento, o que já foi realizado, etc.), bem como resultados finais de problemas submetidos.

Para permitir que o usuário possa iniciar e acompanhar suas execuções de qualquer lugar, sem precisar instalar *software* específico para tal, os *loader clients* e os *status clients* dispõem de interface *WEB*. Isto faculta que de qualquer navegador HTML o usuário possa acompanhar os processamentos de seus problemas. Eventos significativos, por exemplo término de um processamento, também são notificados por *email* ao usuário.

A política de distribuição de carga é *receiver-initiated*. Os nodos desocupados solicitam trabalho do servidor. Como política de balanceamento de carga cada nodo recebe somente uma Unidade de Trabalho, a velocidade com que o mesmo retorna os resultados e recebe uma nova UT caracteriza sua contribuição.

Para evitar que um nodo lento (nodo com menor poder computacional ou ocupado por outros serviços) retarde o final do processamento, uma vez distribuídas todas as UTs, aquelas cujos resultados não retornaram são distribuídas novamente. Para estas, o primeiro resultado que retornar caracteriza o fim do processamento da UT.

### Implementação

O Uniclust foi implementado em linguagem C [KER 78], utilizando-se as primitivas *sockets* e *threads* disponíveis em sistemas UNIX [STE 97]. O sistema oferece ao usuário a possibilidade de construção das rotinas de processamento em todas linguagens compatíveis com o compilador GCC, dentre elas: Fortran77, C++ e Objective-C. Particularmente a compatibilidade com o Fortran, devido a sua elevada disseminação em áreas aplicadas se mostra oportuna. Um exemplo concreto é o problema utilizado para exemplificar a operação do Uniclust neste artigo. O algoritmo de cálculo do *processor client*

<sup>1</sup>Search Extraterrestrial Intelligence at Home - <http://setiathome.ssl.berkeley.edu>

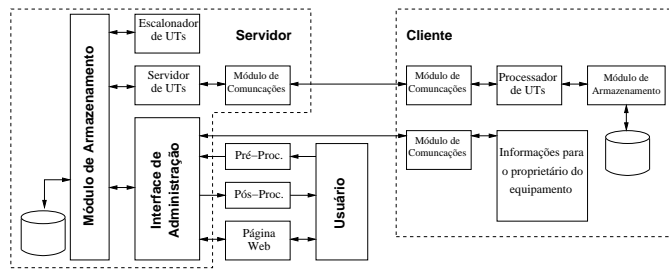


Figura 2: ARQUITETURA DO UNICLUSTER

foi gerado a partir de código escrito em Fortran77. O Unicluster foi construído pensando também na portabilidade do sistema como um todo. O sistema foi testado em plataformas Linux, SunOS e Windows [RED 2001].

A interface *WEB* do Unicluster utiliza a linguagem PHP e o Apache como servidor HTTP. Esta interface, além dos aspectos de gerência operacional já comentados, também apresenta um registro histórico das execuções do usuário simplificando a tarefa de observar tendências no comportamento.

## Unicluster: Uma Aplicação em Engenharia Civil

A análise de vigas e pilares de concreto normalmente é feita utilizando-se valores médios para as propriedades dos materiais e valores nominais para as dimensões da estrutura. Contudo, as propriedades dos materiais e as dimensões da estrutura sempre apresentam uma certa dispersão em torno de seus valores médios. Para avaliar os efeitos destas incertezas sobre a resposta estática da estrutura, é empregado o método de Monte Carlo. Os dados de entrada de cada análise são gerados aleatoriamente, de acordo com suas distribuições de probabilidade. As análises são repetidas inúmeras vezes, obtendo-se diversas respostas, que são armazenadas. Após, um processamento estatístico permite determinar a média, o desvio padrão e estimar a distribuição de probabilidade de deslocamentos, deformações e tensões [REA 2001].

## Resultados

Foram realizadas três execuções da aplicação de Engenharia Civil. As duas primeiras foram executadas em um ambiente heterogêneo utilizando SunOS 5.8, sob duas estratégias: (i) múltipla entrega de UTs (vide tabela 1), o tempo total de processamento paralelo foi de 3988s, perdeu-se em comunicação e gerência do paralelismo 20s (0.5%); (ii) entrega única de UTs, nesta execução obteve-se tempo de processamento de 4236s. O fato deste tempo ser maior deve-se às máquinas de menor poder computacional postergarem o final do processamento.

A terceira execução do problema foi realizada em um ambiente homogêneo utilizando Linux, com máquinas de uso exclusivo (vide tabela 2). Obteve-se 5251s como

Tabela 1: RESULTADOS DA EXECUÇÃO 1 (AMBIENTE HETEROGÊNEO)

| Hostname | Estação             | Tempo total (s) | UTs | Tempo médio (s) | Percentual de contribuição |
|----------|---------------------|-----------------|-----|-----------------|----------------------------|
| bolicho  | SparcStation 5      | 3907            | 28  | 139.54          | 9.36                       |
| coxilha  | UltraSparc 10       | 3960            | 70  | 56.67           | 23.41                      |
| guaiaica | UltraSparc 10       | 3968            | 79  | 50.23           | 26.42                      |
| jacui    | SparcEnterprise 150 | 3863            | 38  | 101.66          | 12.71                      |
| poncho   | SparcStation 20     | 3927            | 28  | 140.25          | 9.36                       |
| prenda   | Sparc Classic       | 3384            | 4   | 846.00          | 1.34                       |
| spica    | UltraSparc 5        | 3865            | 53  | 72.92           | 17.73                      |

Tabela 2: RESULTADOS DA EXECUÇÃO 3 (AMBIENTE HOMOGÊNEO)

| Hostname   | Estação          | Tempo total (s) | UTs | Tempo médio (s) | Percentual de contribuição |
|------------|------------------|-----------------|-----|-----------------|----------------------------|
| apollo     | PentiumII-266MHz | 5191            | 59  | 87.98           | 19.73                      |
| challenger | PentiumII-266MHz | 5251            | 62  | 84.69           | 20.74                      |
| columbia   | PentiumII-266MHz | 5191            | 59  | 87.98           | 19.73                      |
| mir        | PentiumII-266MHz | 5186            | 59  | 87.90           | 19.73                      |
| pathfinder | PentiumII-266MHz | 5133            | 61  | 84.15           | 20.40                      |

tempo total de processamento paralelo, e foi consumido menos de 1s em comunicação e gerência. O *speedup* com 5 processadores obtido para esta execução foi de 4.57.

## Conclusões

Desenvolver aplicações distribuídas em ambientes heterogêneos, de uso compartilhado e de comportamento imprevisível, como empregado pelo Unicluster (Internet), ainda é uma tarefa difícil. Neste artigo apresentou-se uma proposta de *Internet Computing Platform* que atingiu bom nível de desempenho em diversas aplicações. Foram destacados os resultados atingidos por uma aplicação de Engenharia Civil, os quais caracterizam sua possibilidade de uso tanto em ambiente homogêneo (*cluster* dedicado), como em uma rede de estações heterogêneas e de uso compartilhado.

## Referências

- [CUL 99] CULLER, D. **Parallel computer architecture**: a hardware and software approach. San Francisco, California: Morgan Kaufmann, 1999.
- [KER 78] KERNIGHAN, B. W.; RITCHIE, D. M. **C a linguagem de programação**. Rio de Janeiro: Editora Campus, 1978.
- [REA 2001] REAL, M.; CAMPOS, F. A. Aplicação do método de monte carlo no estudo de vigas e pilares de concreto armado. **Revista Teoria e Prática na Engenharia Civil**, Rio Grande, n.2, p.35–44, Maio 2001.
- [RED 2001] RED HAT, I. **Cygwin**. RTP, NC: <http://www.cygwin.com>, 2001.
- [STE 97] STEVENS, W. R. **Unix network programming**. Upper Saddle River, New Jersey: Prentice-Hall, 1997. v.2.