



Busca Distribuída Utilizando Comunicação em Grupo para a Resolução do Problema *8-Puzzle*

Acadêmico
Anderson Rodrigo Janke

Orientador
Prof. Paulo Fernando da Silva



Apresentação

- Introdução
 - Contextualização
 - Objetivos
- Fundamentação
- Desenvolvimento
 - Requisitos
 - Especificação
 - Implementação
 - Operacionalidade
 - Resultados e Discussão
- Conclusão
 - Extensões
- Referências Bibliográficas



Introdução

Contextualização



- Computadores ociosos
- Poder computacional
- Sistemas distribuídos
- Comunicação em grupo
- Busca distribuída

Introdução

Objetivos



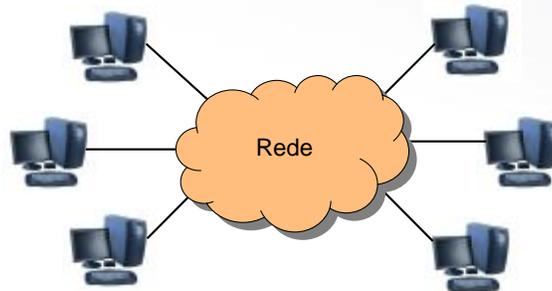
- Solucionar diferentes estados do problema *8-Puzzle* através de busca distribuída
- Utilizar o mecanismo de comunicação em grupo *Spread* na plataforma *Windows*
- Gerenciar 20 computadores interligados através de uma rede LAN

Fundamentação

Sistemas Distribuídos



- Conjunto de componentes interligados através de uma estrutura de comunicação, que cooperam e se coordenam entre si.

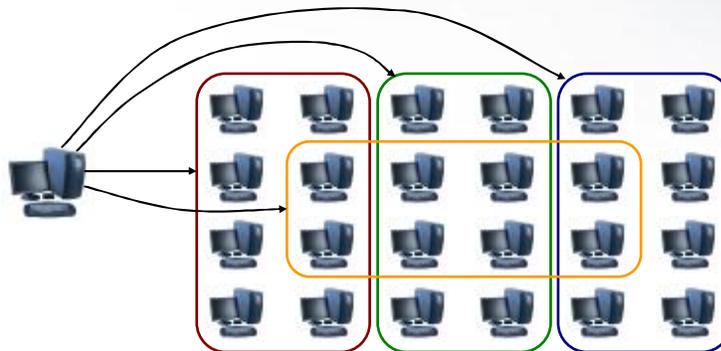


Fundamentação

Comunicação em Grupo



- Comunicação de um para muitos
- Agrupamento de processos
- Confiabilidade



Fundamentação

O problema 8-Puzzle



5	4	
6	1	8
7	3	2

Estado Inicial

1	2	3
8		4
7	6	5

Estado Final

Fundamentação

Algoritmos de busca



- Busca em largura
- Busca em profundidade
- Busca heurística
- Mecanismo de poda

Fundamentação

Biblioteca *wxWidgets*



- Permitir portabilidade
- Constitui uma biblioteca de classes
- Possui uma API padrão para todas as plataformas

Fundamentação

Trabalhos correlatos



- Aplicativo de troca de mensagens (GOULART, 2002, p. 86)
- Aplicativo de coleta de dados e gerência de computadores em uma rede (GOULART, 2002, p. 134)
- Aplicativo de unificação de *logs* para um *cluster* de servidores *Apache* (MODLOG, 2000)
- *AOL Communicator* (AOL, 2003)
- *Audacity* (AUDACITY)
- *AVG Antivirus* (GRISOFT, 2004)

Desenvolvimento

Requisitos Funcionais



- Possuir um aplicativo cliente
 - Controlar as configurações do problema
 - Gerenciar a busca entre os servidores
- Possuir um aplicativo servidor
 - Efetuar a busca pela solução do problema
 - Enviar estatísticas para o aplicativo cliente
 - Informar a solução para o aplicativo cliente caso encontre-a

Desenvolvimento

Requisitos Não Funcionais



- Arquitetura cliente/servidor
- Utilizar o mecanismo *Spread*
- Funcionar em uma rede LAN
- Compatível com *Windows 2000 Professional* ou superior
- Oferecer estabilidade e bom desempenho

Desenvolvimento

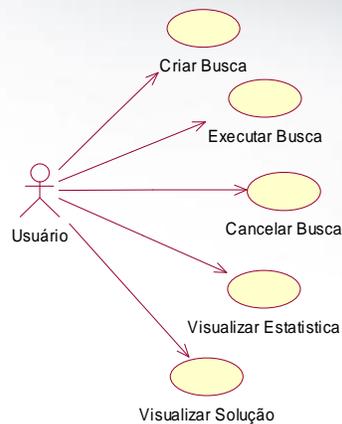
Especificação



- Técnicas e ferramentas utilizadas
 - Diagramas de casos de uso, classe e seqüência, da linguagem UML
 - Aplicativo *Rational Rose* versão 2003.06

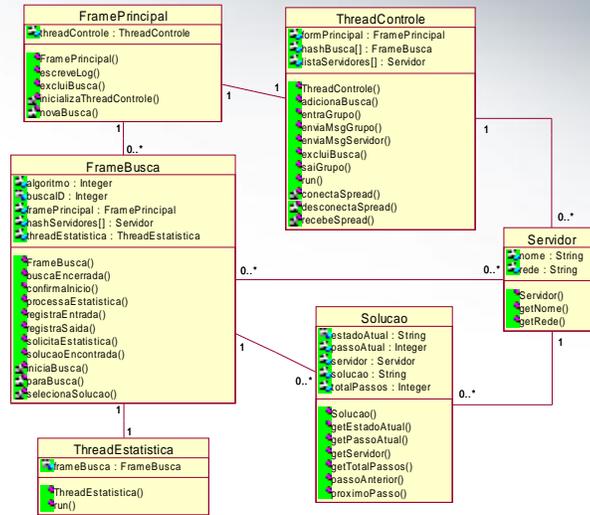
Desenvolvimento

Diagrama de Casos de Uso do Aplic. Cliente



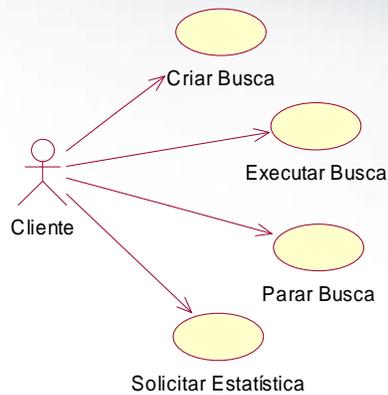
Desenvolvimento

Diagrama de Classes do Aplicativo Cliente



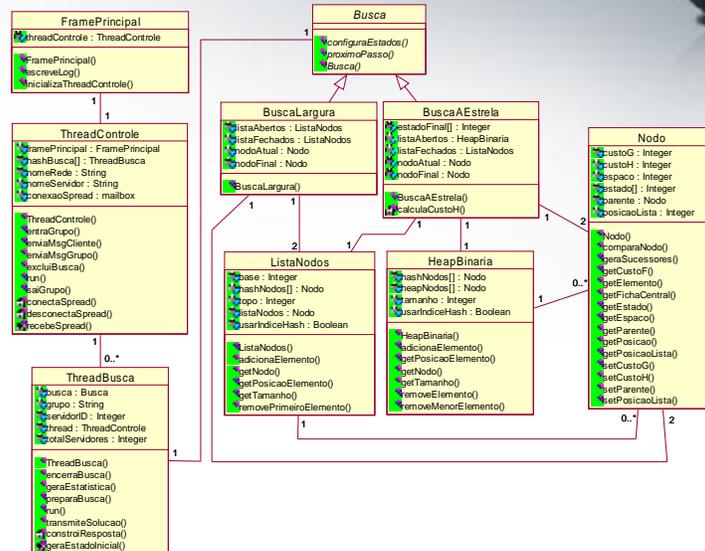
Desenvolvimento

Diag. de Casos de Uso do Aplic. Servidor



Desenvolvimento

Diagrama de Classes do Aplicativo Servidor



Desenvolvimento

Implementação



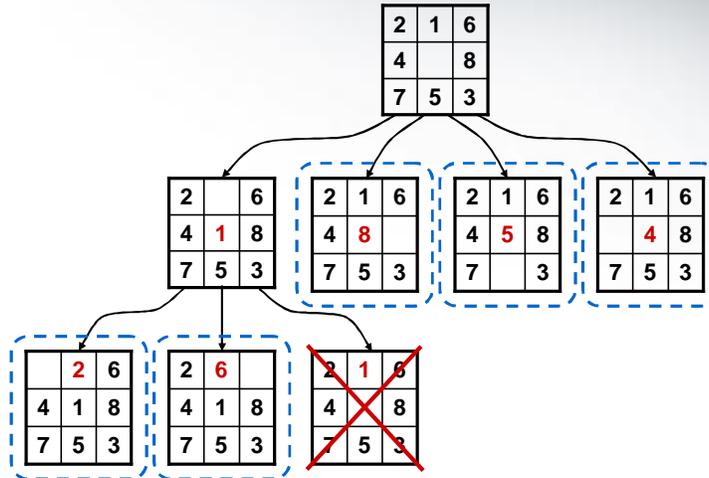
- Técnicas e ferramentas utilizadas
 - Mecanismo de comunicação em grupo *Spread*
 - Biblioteca *wxWidgets*
 - Aplicativo *Microsoft Visual C++ .Net* versão 7.1
 - Compilador *gcc* versão 3.3.3

Desenvolvimento

Implementação - Continuação



- Divisão de nodos

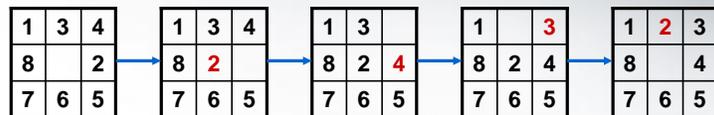


Desenvolvimento

Implementação - Continuação



- Solução



134802765 134820765 130824765 103824765 123804765

4 5 2 1 4

134802765 134820765 130824765 103824765 123804765

Desenvolvimento Operacionalidade



- Servidor *Spread*

```
D:\WC\spread-bin-3.17.2\win\spread.exe
Partial Funding provided by the Defense Advanced Research Project Agency
(DARPA) and the National Security Agency (NSA) since 2000. The Spread
toolkit is not necessarily endorsed by DARPA or the NSA.

For a full list of contributors, see Readme.txt in the distribution.
WWW:      www.spread.org    www.cndc.jhu.edu    www.spreadconcepts.com
Contact:   spread@spread.org

: Version 3.17.02 Built 5/March/2004
=====
Conf_init: using file: spread.conf
Successfully configured Segment 0 (192.168.0.255:4883) with 1 proc:
    localhost: 192.168.200.1
Finished configuration file.
Conf_init: My name: localhost, id: 192.168.200.1, port: 4883
Membership id is (-1062680575, 1100001723)
-----
Configuration at localhost is:
Num Segments 1
  1          192.168.0.255    4883
             localhost      192.168.200.1
-----
```

Desenvolvimento Operacionalidade - Continuação



- Aplicativo Servidor

```
Processo Servidor (region)
-----
[15:22:06] Thread de controle de Spread iniciada
ID = Da4002
Prioridade = 50

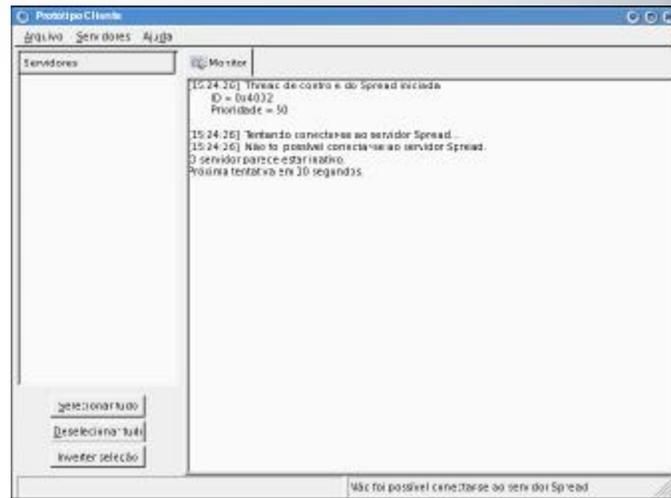
[15:22:06] Tentativa de conexão ao servidor Spread
[15:22:06] Não foi possível conectar-se ao servidor Spread.
O servidor não está estático.
Processo tentará em 120 segundos.

-----
Processo: Não foi possível conectar-se ao servidor Spread
```

Desenvolvimento Operacionalidade - Continuação



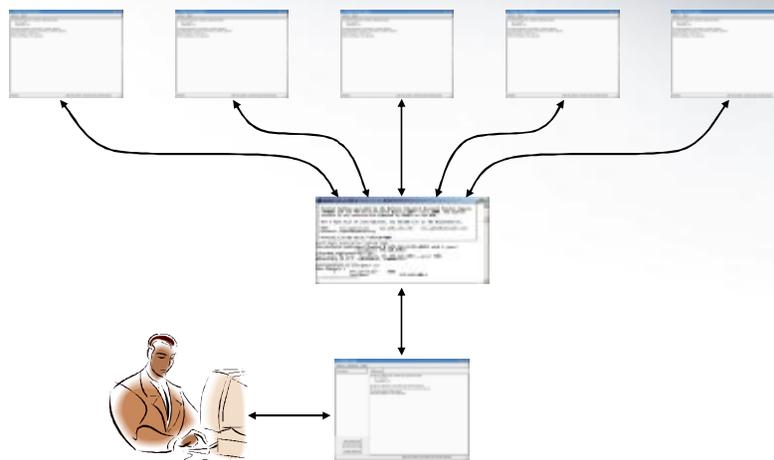
- Aplicativo Cliente



Desenvolvimento Operacionalidade - Continuação



- Visão Geral



Desenvolvimento

Operacionalidade - Continuação



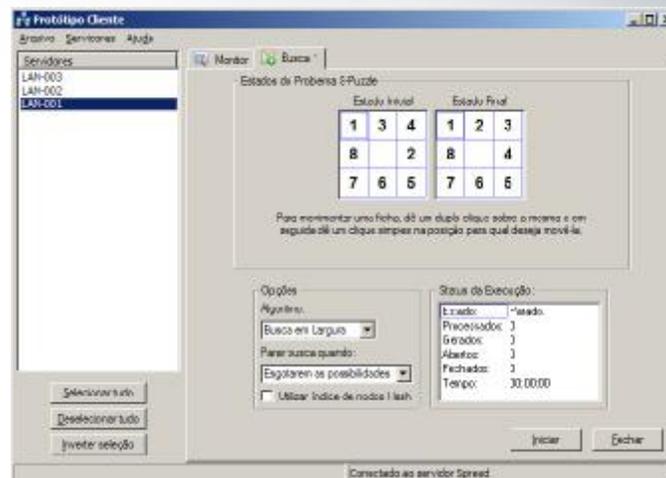
- Solucionar o problema *8-Puzzle* com os estados:
 - Inicial: 2, 1, 6, 4, 0, 8, 7, 5, 3
 - Final: 1, 2, 3, 8, 0, 4, 7, 6, 5
- Utilizando 1 e 3 servidores
- Utilizando variações das configurações de algoritmo e índice de nodos *hash*

Desenvolvimento

Operacionalidade - Continuação



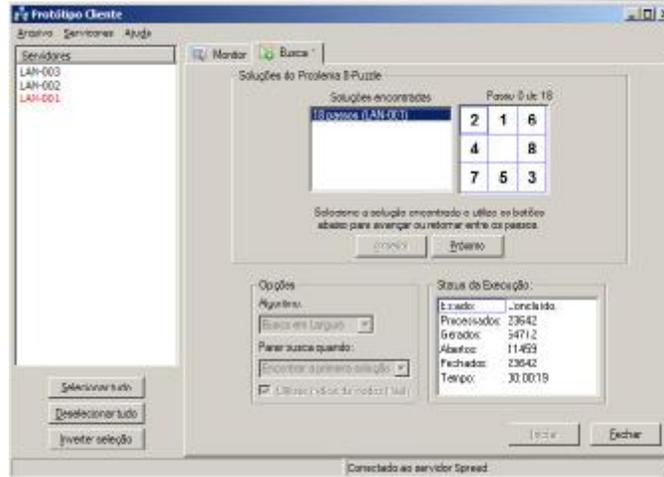
- Configurando o aplicativo Cliente



Desenvolvimento Operacionalidade - Continuação



- Verificando solução



Desenvolvimento Resultados e Discussão



Busca	Servidores	Algoritmo	Índice hash	Nodos		Concluído	Tempo (m:ss)
				Gerados	Processados		
1	1	Largura	Não	64712	23642	Sim	3:22
2	1	Largura	Sim	64712	23642	Sim	0:19
3	1	A*	Não	49995	18409	Sim	2:15
4	1	A*	Sim	49995	18409	Sim	0:17
5	3	Largura	Não	162483	60843	Sim	1:59
6	3	Largura	Sim	183640	67762	Sim	0:16
7	3	A*	Não	96793	35576	Sim	0:41
8	3	A*	Sim	119065	43863	Sim	0:10

Desenvolvimento

Resultados e Discussão – Continuação



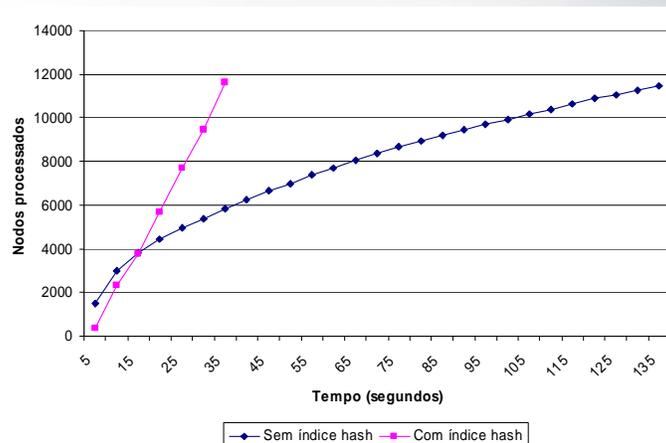
- Distribuição é um método eficiente
- Busca A* é mais rápida que Busca em Largura
- Mecanismo *Spread* é prático e estável
- Protótipo com a biblioteca *wxWidgets* é estável no *Windows* e instável no *Linux*
- Busca com utilização de índice de nodos por *hash* é mais rápida

Desenvolvimento

Resultados e Discussão – Continuação



- Índice de nodos *hash*

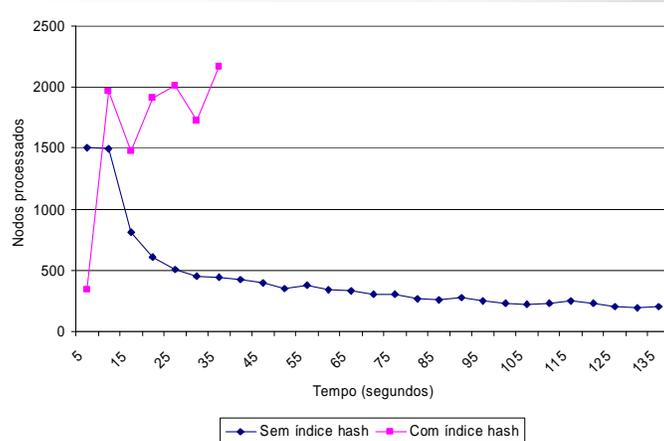


Desenvolvimento

Resultados e Discussão – Continuação



- Índice de nodos *hash*



Desenvolvimento

Resultados e Discussão – Continuação



- Trabalhos correlatos
 - Forma de utilização do mecanismo *Spread* é a mesma
 - Aplicativos com *wxWidgets* tem desempenho similar no *Windows*, porém maior estabilidade no *Linux*

Conclusão



- Todos os objetivos foram alcançados
- Biblioteca *wxWidgets* é vantajosa para multiplataforma
- Demais ferramentas foram adequadas para o desenvolvimento
- Protótipo limitado ao problema *8-Puzzle*, intolerante a falhas

Conclusão

Extensões



- Portar para a linguagem *Java* em forma de *applet*
- Aperfeiçoar para redes WAN
- Tolerância e recuperação de falhas
- Filtro de respostas
- Outros algoritmos
- Outras áreas

Referências Bibliográficas



- AOL. **Aol communicator**, Dulles, 2003. Disponível em: <<http://www.aolepk.com/communicator/index.html>>. Acesso em 12 nov. 2004.
- AUDACITY. **Audacity**, [s.l.], [s.d]. Disponível em: <<http://audacity.sourceforge.net/>>. Acesso em 12 nov. 2004.
- GOULART, Ademir. **Avaliação de mecanismos de comunicação em grupo para ambientes wan**. 2002. 169 f. Dissertação (Mestrado em Ciências da Computação) - Universidade Federal de Santa Catarina, Florianópolis.
- GRISOFT. **Avg anti-virus**, Lidicka, 2004. Disponível em: <<http://www.grisoft.com>>. Acesso em 12 nov. 2004.
- MODLOG. **Mod_log_spread**: a tool for distributed auditing and monitoring, [s.l.], 2000. Disponível em: <http://www.lethargy.org/mod_log_spread/>. Acesso em: 11 ago. 2004.