

Universidade de Brasília  
Disciplina de Sistemas Multi Agentes 2009/1  
Professora Dra. Célia Ghedini Ralha

# SMA Torrent

Um cliente de bit torrent usando Sistemas Multiagentes



Manoel Campos da Silva Filho

<http://github.com/manoelcampos/SMA-Torrent>

# Agenda

- Apresentação do BitTorrent
- Elementos de uma rede BitTorrent
- SMA para Clientes de BitTorrent
- PEAS (Performance, Environment, Actuators, Sensors)
  - Características do ambiente e dos agentes
- Framework de desenvolvimento
- Arquitetura da solução, Protocolo de interação
- Recursos da aplicação, funcionamento
- Demonstração
- Desafios de implementação e limitações
- Conclusão e Referências

# Apresentação do BitTorrent

- Rede P2P (ponto-a-ponto) para compartilhamento de arquivos
- Não existe a figura do servidor na forma tradicional
- Os pontos (peers) podem ser clientes e/ou servidores
- Não existe problema de free-riding na especificação oficial (implementações como a do cliente BitThief[1] burlam a especificação)
- Os arquivos compartilhados são divididos em várias pequenas partes
- Conceito de enviar enquanto recebe partes de um arquivo

[1] <http://dcg.ethz.ch/projects/bitthief/>

## Elementos de uma rede BitTorrent

- Peer: cliente que envia ou recebe partes de um arquivo
- Arquivo .torrent: contém meta-dados a respeito do arquivo que deseja-se baixar.
- Torrent Indexer: WebServer que armazena arquivos torrents, possuindo um sistema de busca para os usuários localizarem torrents para arquivos desejados.
- Tracker: responsável por disponibilizar o endereço dos peers que armazenam arquivos a serem compartilhados. Cada arquivo torrent mantém o endereço de um ou mais trackers. Após início de um download ou upload, a comunicação entre os peers pode continuar sem o tracker. Ele não armazena informações sobre os arquivos.

# Elementos de uma rede BitTorrent

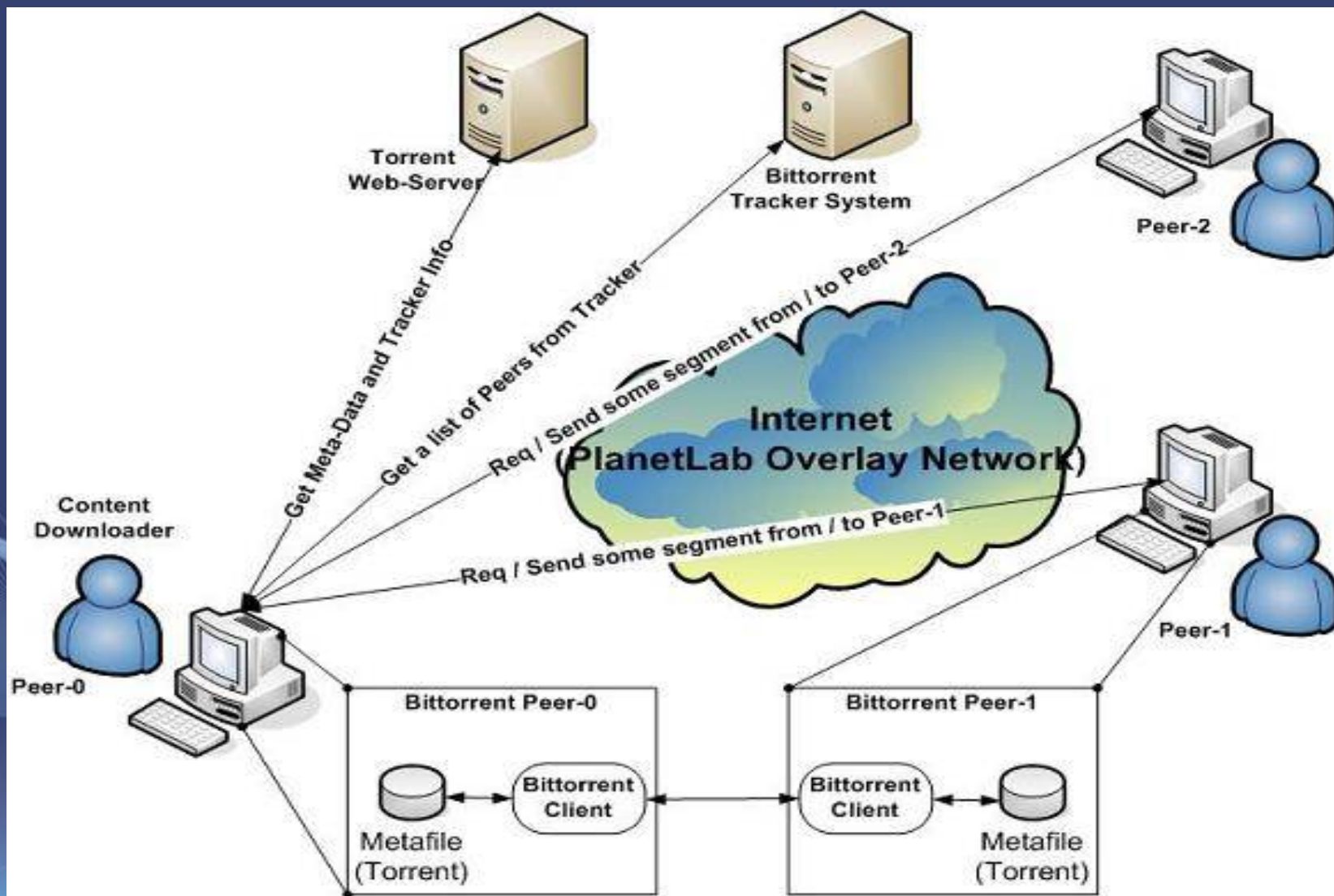
- Seeder: um peer que contém um arquivo completo e faz upload desse arquivo para os outros que solicitarem
- Leecher: um peer que não contém um arquivo completo, assim, além de enviar partes que ele já recebeu para outros peers que solicitarem, baixa as partes restantes de outros peers.



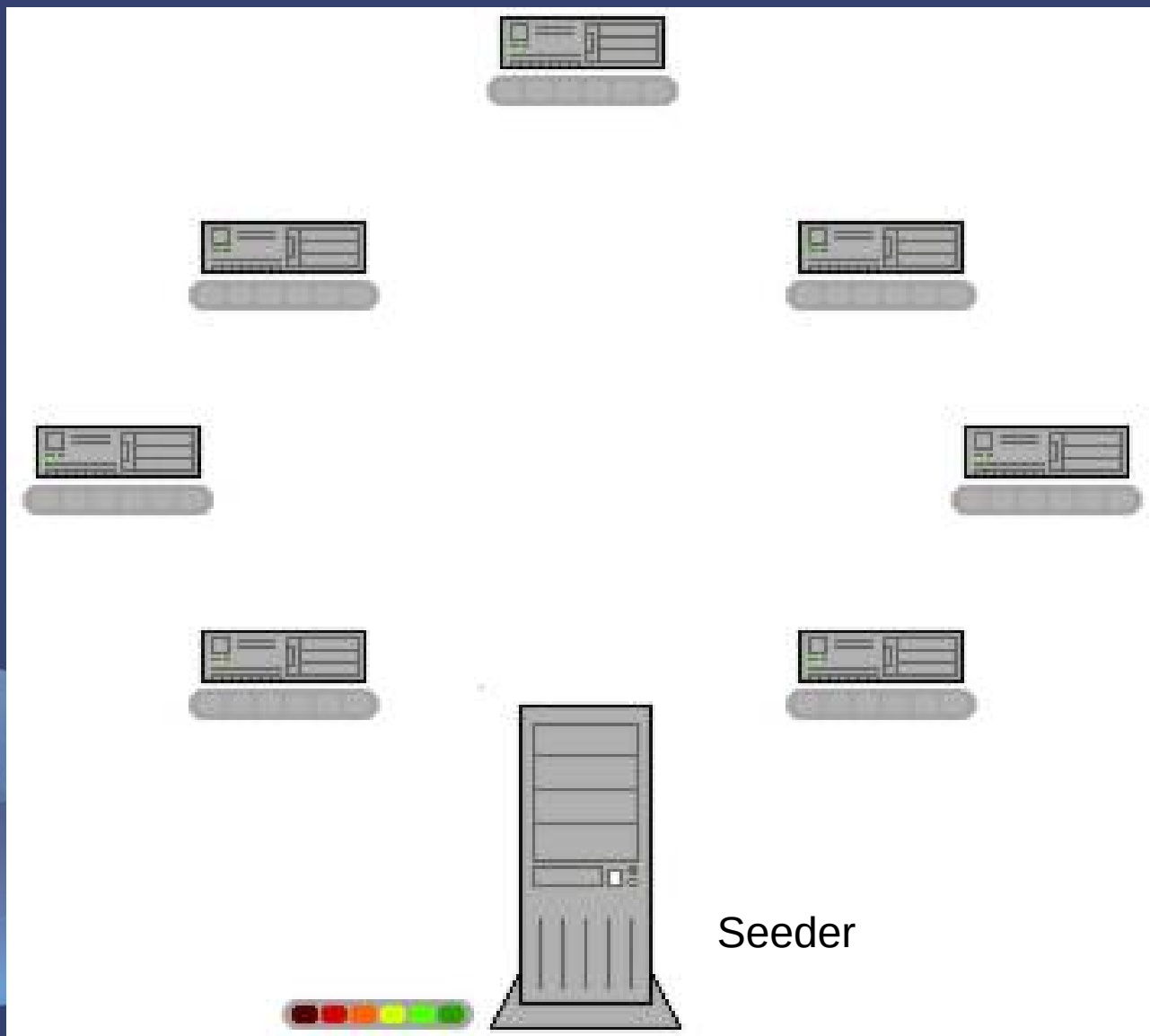
## Detalhes a serem tratados na implementação

- Cada cliente que implementa o protocolo pode escolher sua própria lógica para seleção dos peers para quem vai disponibilizar uma parte de um arquivo solicitado. Por exemplo, pode escolher aleatoriamente para qual peer disponibilizar, baixando as partes mais raras primeiro

# Elementos de uma rede BitTorrent



# Elementos de uma rede BitTorrent





## SMA para Clientes de BitTorrent: porque usar

- Os agentes podem negociar o download de partes de um arquivo
- Os agentes inteligentes podem decidir como escolher de quem pegará um parte de um arquivo, ou para quem enviar uma parte de um arquivo

## PEAS (Performance, Environment, Actuators, Sensors)

- Na 1a. ed de Russel e Norvig era denominado como PAGE – Perception, Actions, Goals, Environment
- O PEAS
  - Performance: baixar o arquivo na menor quantidade de tempo possível
  - Environment: definido a seguir
  - Actuators: conectar ao tracker para pegar a lista de agentes de onde baixar partes de um arquivo
  - Sensores: detectar quando novo agente está online para disponibilizar um pedaço de um arquivo

# Características do Ambiente

- Parcialmente observável: os agentes não podem, por exemplo, ter acesso a todas as partes do arquivo a qualquer momento que desejarem, apenas quando outros agentes disponibilizarem estas
- Estocástico: o próximo estado não depende somente do estado atual e das ações do agente, pois falhas imprevisíveis de software e hardware podem ocorrer.
- Episódico: o ambiente é episódico, pois o arquivo é dividido em partes onde cada uma é transferida numa conexão separada e independente das outras partes do arquivo

# Características do Ambiente

- Dinâmico: o ambiente pode mudar enquanto o agente está executando uma tarefa. Por exemplo, enquanto o agente está baixando um arquivo, a outro agente que está fornecendo pode ser fechado pelo usuário e o pedaço tem que ser pego de outro agente.
- Discreto: o ambiente possui um conjunto finito de estados, por exemplo, um agente peer está online, um pedaço do arquivo está disponível.
- Multi-Agente: o sistema é composto por vários agentes que se comunicam e trocam arquivos.

# Características dos agentes

- Agentes baseado em objetivos: o objetivo dos agentes será baixar o arquivo na menor quantidade de tempo, e não ser egoísta, sempre disponibilizando arquivos para outros agentes baixarem

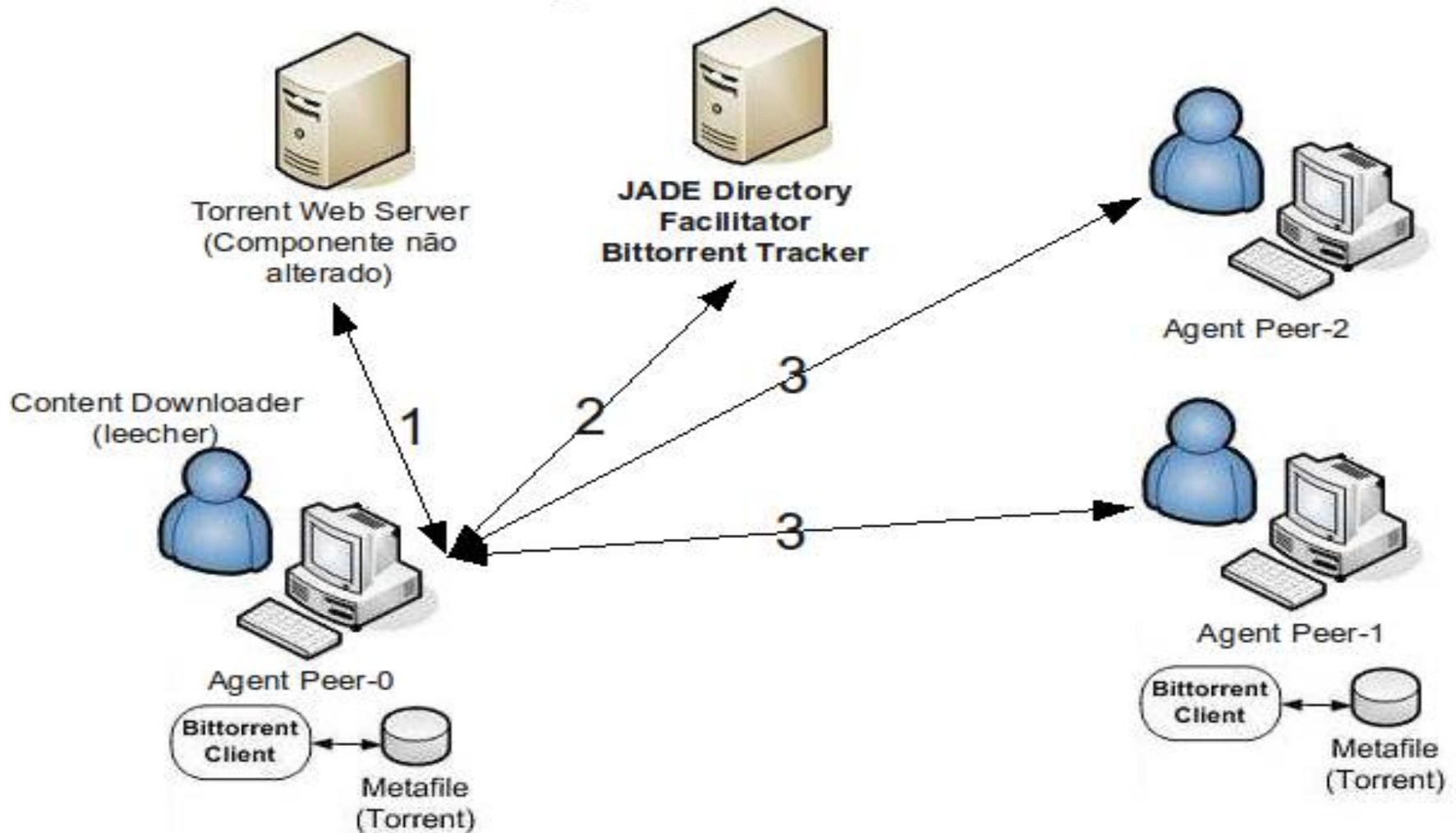


# Framework de Desenvolvimento

- Foi utilizado o framework Jade para desenvolvimento da aplicação

# Arquitetura da solução

## JADE Multi-agent BitTorrent Architecture



# Protocolo de Interação Linguagem de Comunicação

- Protocolo de interação Contract Net para negociar troca de partes de arquivos
- Mensagens ACL para troca de mensagens e partes

# Recursos da aplicação

- Adicionar, remover, pausar e reiniciar torrents (arquivos compartilhados)
- Adicionar torrents para obter um determinado arquivo
- Adicionar torrents para um arquivo que o usuário possui e que deseja compartilhar com outros
- Continuar a baixar um arquivo do mesmo ponto onde parou, mesmo depois de reabrir a aplicação

# Funcionamento da Solução

- Outros agentes (clientes) que estejam executando na plataforma e que tenham um arquivo para compartilhar registram cada arquivo, por meio de seu torrent, como um serviço no DF;
- o usuário que deseja baixar um arquivo obtém um torrent em um Web Server indexador;
- executa o cliente (agente) que possui uma GUI;
- adiciona o arquivo torrent à lista de arquivos do agente;



# Funcionamento da Solução

- o cliente (agente) inicia um TickerBehaviour FindPeersBehaviour para fazer solicitações recorrentes ao DF/Tracker por clientes (agentes/peers) remotos que tenham o mesmo arquivo para compartilhar. Os agentes com o mesmo arquivo são localizados pois cada arquivo que ele compartilha é registrado como um serviço no DF/Tracker;
- com essa lista é iniciado um SimpleBehaviour RequestPiecesBehaviour que executa alguns passos: enviar CPF para os agentes obtidos solicitando propostas (as propostas devem incluir as partes do arquivo solicitado que o cliente/agente/peer remoto possui), aguardar propostas e receber partes;

# Funcionamento da Solução

- depois de enviado CFP, e a cada proposta recebida, o cliente (agente) escolhe aleatoriamente qual pedaço do arquivo solicitar do cliente (agente/peer) remoto;
- o cliente envia mensagem ACCEPT\_PROPOSAL informando que aceitou a proposta do agente remoto, depois de ter escolhido qual parte ele deseja obter, indicando a este que envie a parte solicitada;
- o agente remoto envia a parte solicitada numa mensagem INFORM, estando a parte solicitada, dentro do conteúdo da mensagem;
- o agente solicitante salva a parte recebida no final de um arquivo temporário;

# Funcionamento da Solução

- o cliente pode receber partes de vários outros clientes (peers) remotos;
- enquanto estiver recebendo partes, ele pode também servir outros clientes remotos, enviados partes que ele já possui. Para isso, cada cliente tem um comportamento `CyclicBehaviour` `ReceiveRequestsBehaviour` que fica constantemente aguardando requisições;
- para cada arquivo compartilhado, é adicionado um conjunto dos behaviours citados para fazer requisições por novas partes desses arquivos, exceto o behaviour que aguarda requisições, que é único para cada cliente e não finaliza.



DEMONSTRAÇÃO

# Desafios de implementação

- Compreender a estrutura do arquivo torrent e o formato de codificação b-encode utilizado nele;
- Implementar uma classe java para permitir a leitura e geração de arquivos torrent;
- Compreender e implementar a troca de mensagens de acordo com o protocolo de BitTorrent;
- Obtenção de partes do arquivo;
- Montagem de arquivo com as partes aleatoriamente recebidas do arquivo compartilhado e ordenação das partes gerando o arquivo final.



# Limitações

- A aplicação não lê arquivos torrent que tenham mais de um tracker registrado, apesar de este campo não ser usado, pois o tracker passou a ser o DF do Jade;
- Nem todos os recursos do protocolo BitTorrent foram implementados;
- Os clientes (agentes jade) não podem interoperar com uma rede bit torrent tradicional, pois a arquitetura é diferente, devido ao fato de que os clientes não se comunicam com um tracker em um servidor web, o mesmo que clientes de bit torrent tradicionais usam, e sim com o DF do Jade.

# Conclusões

- O desenvolvimento da solução permitiu o conhecimento da plataforma Jade e a superação de vários desafios de implementação, além da compreensão de conceitos de agentes, como serviços, negociação e comportamentos;
- O sistema desenvolvido serviu como prova de conceitos dos fundamentos de agentes e dos recursos disponíveis na plataforma Jade;

# Conclusões

- permitiu alcançar maior conhecimento dos detalhes de implementação de agentes, ao se trabalhar com comportamentos concorrentes, o Contract Net e os serviços do DF do Jade;
- mostrou a viabilidade do desenvolvimento de uma solução de torrent baseada em SMA, mas requer estudo e melhorias para averiguar se a solução pode se tornar realmente superior às implementações tradicionais;
- não foi possível comparar o desempenho da solução contra as redes tradicionais de bit torrent devido ao curto tempo, pois iria requerer uma extensa fase de simulações e testes.

# Referências

- G. Weiss, “Multiagent Systems: A Modern Approach to Distributed Modern Approach to Artificial Intelligence, MIT Press, 1999.
- Java Agent Development Framework, <http://jade.tilab.com/>, Acessado em 22/06/2009.
- Bittorrent fórum, [www.bittorrent.org](http://www.bittorrent.org), Acessado em 08/06/2009.
- B. Fabio, C. Giovanni and G. Dominic, Developing Multi Agent Systems with JADE, Wiley, 2007.
- W. Michael, An introduction to multiagent systems, John Wiley & Sons, 2002.
- JADE/JADEX, <http://wiki.les.inf.puc-rio.br/uploads/e/ef/07-02-JADE-JADEX.ppt>, Acessado em 30/06/2009.
- C. Giovanni, Jade Programming Tutorial for Beginners, <http://jade.tilab.com/doc/index.html>, Acessado em 22/06/2009.
- J.A. Pouwelse, P. Garbacki, D. H. J. Epema and H. J. Sips, An introduction to the BitTorrent Peer-to-Peer File-Sharing System, Delft University of Technology, The Netherlands, 2004.
- Bittorrent Protocol Specification v1.0, <http://wiki.theory.org/BitTorrentSpecification>, Acessado em 12/06/2009.
- C. M. Henrique et. al., Multi-Agent System Model of a BitTorrent Network, “Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing”, 2008.
- C. Bram, Incentives Build Robustness in BitTorrent, BitConjurer.org, 2003.
- J.A. Pouwelse, P. Garbacki, D. H. J. Epema and H. J. Sips, The BitTorrent P2P File-Sharing System: Measurements and Analysis, Delft University of Technology, The Netherlands.