

TorrentLab: Um Ambiente para Avaliação Integrada do Protocolo BitTorrent

Rodrigo B. Mansilha¹
Marinho P. Barcellos² (orientador)

¹ UNISINOS – Universidade do Vale do Rio dos Sinos, São Leopoldo, RS

²PUCRS – Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, RS

rmansilha@gmail.com, marinho@acm.org

Resumo. *BitTorrent é um protocolo para compartilhamento de conteúdo que tem recebido grande atenção da comunidade científica. Este trabalho apresenta TorrentLab, um ambiente de avaliação integrada, que comporta tanto simulação como experimentos em redes reais do protocolo BitTorrent. Dado um ensaio, resultados obtidos por simulação e experimento em rede real são comparados, mostrando estar em consonância com o comportamento esperado.*

Abstract. *BitTorrent is a file sharing protocol that has received great attention from the scientific community. This paper presents TorrentLab, a testbed in which BitTorrent simulations as well as live experiments can be performed under both controlled and uncontrolled settings. To help validate TorrentLab, swarms are evaluated, and results produced by simulation as well as live experiments are shown to be similar and in line with expected BitTorrent behavior.*

1. Introdução

Compartilhamento de arquivos em redes P2P tornou-se uma das aplicações mais relevantes da Internet, permitindo uma rápida disseminação de conteúdo entre usuários. Nesta categoria, BitTorrent é um dos protocolos mais populares, sendo utilizado por milhões de pessoas para compartilhar conteúdo [Das et al. 2006].

A literatura é rica em artigos sobre BitTorrent, a maioria dos quais requer algum tipo de avaliação quantitativa, com objetivos tais como elucidar o comportamento de sistemas atuais e avaliar o benefício de novas propostas. No entanto, existe uma carência por avaliações mais abrangentes, incluindo cenários mais realistas (com uma modelagem mais detalhada do protocolo) e avaliações em redes reais, com implementações populares. Classicamente, existem três metodologias de avaliação: *analítica*, *simulacional* e *experimental*, sendo cada método mais adequado a uma determinada situação. Idealmente, as técnicas de avaliação não deveriam ser usadas isoladamente, de forma a oferecer uma confiança maior na validade dos resultados. Independente da metodologia escolhida, existe sempre um custo associado à sua execução, potencialmente inviabilizando empregar mais de uma metodologia em conjunto. Um dos maiores desafios é o esforço envolvido ao se passar de uma metodologia para outra, como por exemplo entre simulação e avaliação experimental em uma rede real. Tanto a implementação como a plataforma de execução dos experimentos tende a ser completamente distinta. Neste contexto, o desafio enfrentado nesta pesquisa é avaliar o BitTorrent através de diferentes metodologias, de maneira abrangente (mais de uma metodologia), integrada (consistência entre metodologias) e sem-esforço.

Este artigo¹ apresenta a arquitetura, implementação e avaliação de um ambiente de avaliação integrada de redes BitTorrent e por tal denominado **TorrentLab**. Diferentemente de propostas anteriores, o ambiente comporta de maneira homogênea a execução

¹este artigo descreve os resultados de meu trabalho de Iniciação Científica e Trabalho de Conclusão, cujo objetivo foi o projeto e a implementação do TorrentLab

de avaliações baseadas em simulação e experimentos em rede real. Em termos gerais, a rede BitTorrent é primeiramente descrita pelo usuário em bom grau de detalhe através de um Ambiente de Modelagem. A especificação gerada serve de entrada para ambos os métodos de avaliação considerados. O TorrentLab é composto de mais três ambientes, sendo dois deles denominados **TorrentSim** e **TorrentExp**. Como o próprio nome já indica, o primeiro é responsável pelas simulações, enquanto o segundo pelos experimentos na rede. Ambos valem-se de uma infra-estrutura de computação distribuída para viabilizar a execução paralela de simulações ou a instanciação de agentes em máquinas remotas. As saídas produzidas, em ambos os casos, são consolidadas em um Ambiente de Consolidação de Resultados.

O TorrentLab foi desenvolvido durante bolsa de Iniciação Científica no projeto P2P-SeC² e concomitantemente através de um Trabalho de Conclusão de curso a ser apresentado em Junho de 2008, ambos sob a orientação do Prof. Marinho P. Barcellos. O autor começou a trabalhar no projeto fazendo uma "re-engenharia" do simulador aplicado nos trabalhos [Konrath et al. 2007b] e [Konrath et al. 2007a]. O simulador tornou-se mais eficiente e realista, sendo usado para se obter os resultados em [Mansilha et al. 2007]. Para validar esta versão, foi necessário executar experimentos em rede real e comparar seus resultados com simulações. Este processo demandou grande esforço manual, e levou o autor à idéia de se usar um *middleware* de grade para automatizar a instanciação remota de agentes de usuário P2P. Posteriormente, da necessidade de aproveitar os subprocessos equivalentes (modelagem e geração de gráficos) durante o processo de avaliação via simulação e experimentos, a idéia evoluiu para um ambiente integrado que contemplasse ambas as metodologias. O amadurecimento destas idéias resultou em dois artigos aceitos recentemente, um Qualis A Nacional (SBRC08) e outro Qualis A Internacional (ISCC08). Além disso, o TorrentLab viabilizou experimentos a serem publicados em um terceiro artigo recente Qualis A Nacional (SBRC08). Dado seu conhecimento sobre OurGrid, Prof. Francisco Brasileiro (UFCG) foi convidado e colaborou com os primeiros dois trabalhos citados. O auxílio de demais membros do projeto se restringiu a usar e testar o sistema fornecendo *feedback* como no caso do terceiro trabalho.

O restante do artigo está organizado como segue. Trabalhos relacionados são discutidos na Seção 2. A arquitetura do TorrentLab é apresentada na Seção 3, enquanto que aspectos de implementação são alvo da Seção 4. Para validar a proposta, na Seção 5 resultados de experimentos são analisados. Por fim, na Seção 6, são apresentadas as considerações finais e perspectivas de trabalhos futuros.

2. Trabalhos Relacionados

Esta seção apresenta trabalhos relacionados a formas de avaliação do BitTorrent e ambientes de avaliação similares em natureza ao TorrentLab. Primeiramente, são abordadas formas de avaliação do BitTorrent, que ajudam a demonstrar a empregabilidade do ambiente. Em seguida, trabalhos com propostas semelhantes são discutidos, apontando o diferencial do TorrentLab.

Entre as três metodologias de avaliação anteriormente citadas, a modelagem analítica é possivelmente aquela que permite um melhor entendimento sobre as propriedades globais do sistema. Porém, no caso do BitTorrent, seu funcionamento é amplamente influenciado pela escolha de valores para parâmetros básicos como tamanho máximo do conjunto de pares. Segundo [Bharambe et al. 2006], seria difícil, senão impossível, controlar um espaço tão amplo de possibilidades em uma configuração baseada em modelagem analítica.

²<http://p2p-sec.org/>

A complexidade do problema ou a dificuldade prática de se conduzir experimentos em larga escala leva a abordagem baseada em simulação a ser bastante popular. Exemplos de simulações de BitTorrent são [Bindal et al. 2006], que avalia uma proposta de melhoria ao protocolo, baseada em escolha de vizinhos mais próximos, a fim de diminuir tráfego entre ISPs sem prejudicar o desempenho do enxame [Konrath et al. 2007b, Konrath et al. 2007a, Mansilha et al. 2007], onde os autores avaliam o impacto de ataques de segurança sobre o BitTorrent.

Por fim, no campo experimental, em trabalhos como [Erman et al. 2005] e [Guo et al. 2007] traços são coletados através de pontos de monitoração e de uma versão instrumentada de um agente de usuário BitTorrent. Em diversos trabalhos, agentes de usuário BitTorrent são modificados de maneira a coletar dados e/ou estudar experimentalmente os mecanismos básicos do BitTorrent. São exemplos desta categoria de trabalho [Piatek et al. 2007] e [Sirivianos et al. 2007].

A seguir, são comentados os trabalhos mais próximos a este. Em [Naicken et al. 2007], os autores realizam uma análise superficial sobre simuladores em P2P e encontram uma gama deles, incluindo BitTorrent. Complementarmente à simulação, em [Nussbaum and Richard 2006], apresenta-se o P2PLab, um ambiente que permite a execução de experimentos em rede real, valendo-se de uma grade computacional como substrato. No entanto, o P2PLab exige FreeBSD como sistema operacional e demanda virtualização para execução de múltiplas instâncias de agente P2P na mesma máquina. Além disso, não é orientado para BitTorrent, e por isso não considera as especificidades deste ambiente. O P2PLab pode ser comparado, em termos de funcionalidades, com a infra-estrutura distribuída do TorrentLab. Parece não existir nenhuma proposta similar na literatura, que combine as abordagens e permita a execução de experimentos de investigação de sistemas BitTorrent.

3. Arquitetura do TorrentLab

Esta seção descreve a arquitetura do TorrentLab. As cinco subseções seguintes detalham os respectivos módulos e ambientes na ordem em que segue: Ambiente de Modelagem, Agregado ou Grade Computacional, TorrentSim, TorrentExp e Ambiente de Consolidação de Resultados.

3.1. Ambiente de Modelagem

O Ambiente de Modelagem oferece, de maneira homogênea, funcionalidade para modelagem de enxames para simulação e experimentos. O desafio é criar um mecanismo que atenda diferentes requisitos e sintaxes, identificando parâmetros comuns e permitindo a extensão para parâmetros específicos. Largura de banda dos agentes é um exemplo de parâmetro comum a qualquer simulação ou experimento real. Por outro lado, um usuário disposto a desenvolver, avaliar, ou testar um novo agente de usuário BitTorrent deve ter meios para variar um determinado parâmetro específico que não foi previsto.

Um enxame BitTorrent é descrito através de um conjunto de atributos. Um atributo é formado por um nome, um conjunto de parâmetros e suas unidades de medidas correspondentes. Atributos podem ser organizados em algumas categorias: conteúdo (torrent), população, configuração dos agentes e regras para determinação do fim do experimento.

3.2. Agregado ou Grade Computacional

A utilização de um conjunto de máquinas no TorrentLab aumenta a escalabilidade das simulações e dos experimentos. Do ponto de vista das simulações, a execução em um agregado ou em uma grade computacional permite reduzir o tempo de resposta de uma

campanha, à medida que as várias simulações que formam a campanha podem ser facilmente executadas em paralelo. Por outro lado, uma maior quantidade de máquinas faz com que seja possível instanciar mais agentes BitTorrent e executar experimentos reais com mais pares. Dessa forma, a execução de uma campanha de simulação ou de um experimento podem ambas serem vistas como a execução, sobre a infra-estrutura multi-processada, de uma “aplicação” paralela composta por várias tarefas.

O TorrentSim e o TorrentExp geram tarefas a serem processadas pela infra-estrutura de computação distribuída. Além disso, eles são responsáveis pelo escalonamento dessas tarefas na infra-estrutura de execução. Cada tarefa de uma “aplicação” TorrentLab executa três fases, quais sejam: i) preparação do ambiente (*stage in*), que é responsável por transferir para a máquina da infra-estrutura onde a tarefa irá executar (máquina remota) os dados de entrada e, se necessário, o executável do agente; ii) iniciação da execução do agente na máquina remota (*spawn*); e, iii) coleta de resultados (*stage out*), que é responsável por transferir os resultados da execução do agente da máquina remota para a máquina onde a aplicação foi iniciada. Após a instanciação e término ou falha das tarefas, os resultados são enviados para o respectivo ambiente.

3.3. Ambiente de Simulação: TorrentSim

O TorrentSim é responsável pela instanciação e controle de simulações. Ele oferece uma interface para um simulador de BitTorrent extensível baseado no arcabouço Simmcast [Barcellos et al. 2001]. A integração com o TorrentLab é relevante sobre três aspectos da avaliação através da simulação: modelagem de enxames, execução na infra-estrutura computacional, e comparação de resultados com o TorrentExp. Primeiro, a modelagem de enxames é facilitada porque pode-se reutilizar modelos garantindo equivalência dos parâmetros entre múltiplas execuções. Segundo, a integração com o TorrentLab simplifica execuções em lote através do agregado/grade, abstraindo todo o processo de instanciação e controle. Finalmente, há equivalência entre TorrentExp e TorrentSim no que tange parâmetros de entrada e métodos de consolidação de resultados, permitindo a comparação.

A Figura 1 ilustra a arquitetura do TorrentSim no âmbito do TorrentLab. Primeiramente, o Ambiente de Modelagem fornece o modelo do enxame a ser avaliado. O Ambiente de Simulação traduz o modelo para a sintaxe do simulador e gera uma quantidade de tarefas igual ao número de sementes aleatórias utilizadas. As tarefas, junto do simulador e dos parâmetros, são enviadas para a infra-estrutura computacional subjacente, que por sua vez, encarrega-se de distribuí-las entre os nodos que o compõe. Conforme as simulações terminem ou falhem, os resultados são retornados para o TorrentSim, que os formata para o Ambiente de Consolidação. Por fim, os dados são consolidados gerando os resultados finais.

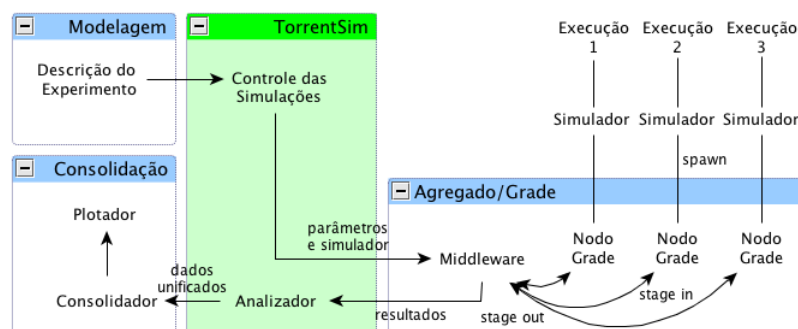


Figura 1. Arquitetura do TorrentSim

3.4. Ambiente de Experimentos: TorrentExp

O objetivo do TorrentExp é permitir que experimentos sejam conduzidos com implementações de BitTorrent em “configurações reais”. Estas podem ser divididas em três tipos, em ordem decrescente de controle e sincronização: uma rede local controlada, provavelmente parte de um agregado de alto desempenho, uma rede de longo alcance com Qualidade de Serviço, parte de uma infra-estrutura de grade, ou este último caso sem quaisquer garantias de *QoS*.

Para permitir a execução de experimentos de acordo com o modelo de enxame inserido, o sistema deve possuir meios para definir quando executar agentes remotos (e quais agentes) sobre uma determinada rede. O sistema deve garantir, ou mais fracamente presumir (por exemplo através de superprovisionamento), que os nodos selecionados possuam no momento recursos disponíveis suficientes para instanciação do agente (por exemplo, largura de banda).

A Figura 2 ilustra a arquitetura do TorrentExp, como inserido no TorrentLab. Primeiramente, o TorrentExp recebe do Ambiente de Modelagem o modelo do enxame a ser avaliado e, de acordo com o mesmo, gera tarefas e transmite-as para um elemento intermediário chamado *Orquestrador*. Este é responsável pela instanciação e sincronização das tarefas na infra-estrutura distribuída. O middleware encarrega-se de distribuir as tarefas entre os nodos que o compõe. Conforme os agentes terminem de cumprir seu papel no enxame, a tarefa termina e os resultados são retornados para o TorrentExp. O nodos liberados são aproveitados pelo Orquestrador para instanciar a chegada de novos pares ao enxame. Ao final da participação de todos os pares, as saídas são processadas e formatadas para o Ambiente de Consolidação. Este, por sua vez, consolida os dados, gerando os resultados finais.

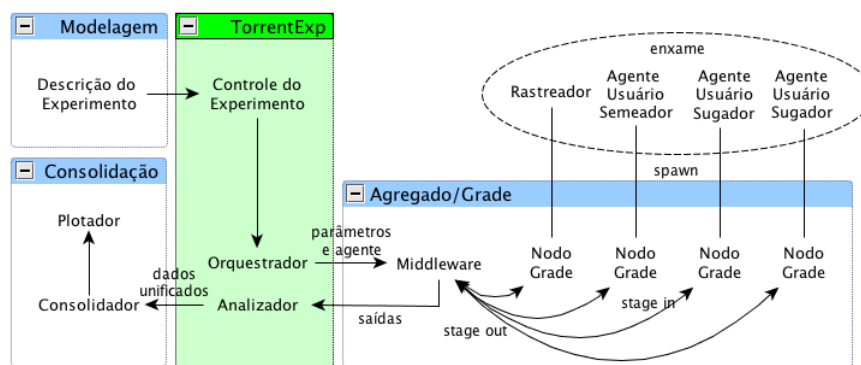


Figura 2. Arquitetura do TorrentExp

3.5. Ambiente de Consolidação de Resultados

O Ambiente de Consolidação de Resultados é responsável por “unificar” os rastros, tanto gerados pelo TorrentSim (e extensões do mesmo) como por diferentes agentes executados no TorrentExp. Cada agente e simulador apresenta syntaxes de saídas diferentes, além de diferir quanto a tipo, unidade e glanuraridade de dados de saída. No entanto, como a proposta é a mesma, encenar enxames BitTorrent, espera-se que a partir de traços de fontes diferentes seja possível identificar dados com semântica em comum.

4. Aspectos de Implementação

A arquitetura apresentada na seção anterior foi materializada através da construção de um protótipo, demonstrando as principais características da proposta. Esta seção oferece uma breve descrição da implementação, ressaltando os aspectos mais relevantes.

Para a implementação do protótipo, foram assumidas premissas simplificatórias. Dentre elas, destacam-se a escolha pelo ambiente Unix e agregado em rede local com pelo menos 100 Mbps de largura de banda.

4.1. Ambientes de Modelagem e de Consolidação de Resultados

O Ambiente de Modelagem implementado baseia-se na definição de uma linguagem e um analisador léxico-sintático correspondente. Para descrever o modelo, pode ser utilizado qualquer editor de textos. O analisador processa o arquivo contendo a descrição do experimento procurando erros gramaticais. Caso algum erro seja encontrado, o programa termina informando o motivo. Caso contrário, as informações são encapsuladas em um objeto que é enviado para o TorrentSim e o TorrentExp. Ambos os ambientes processam este objeto, lidando apenas com parâmetros de seu interesse. O resultado do tratamento de cada parâmetro por cada ambiente é informado ao usuário.

O Ambiente de Consolidação, por sua vez, provê uma interface para a ferramenta GNU Gnuplot. Foram implementados os geradores para os gráficos mostrados na Seção 5.

4.2. Grade Computacional

Dentre as diferentes opções de middleware para grades computacionais, escolheu-se o OurGrid. Esse middleware de código aberto dá suporte à construção de grades computacionais P2P, possibilitando a agregação de um número considerável de recursos.

O OurGrid provê um escalonador de aplicações que se adequa perfeitamente à execução das campanhas de simulação do TorrentSim e que pode ser ligeiramente adaptado para atender aos requisitos do escalonamento de tarefas no caso de experimentos do TorrentExp. Um dos princípios de projeto do OurGrid é a facilidade de instalação e manutenção da grade. Finalmente, o middleware já está bastante maduro, uma vez que dá suporte a uma grade computacional pública que está em produção desde dezembro de 2004³.

Com o objetivo avaliar enxames de tamanho maior que o número de máquinas disponíveis, é possível executar múltiplos agentes BitTorrent em paralelo no mesmo hardware. Isto pode ser feito de duas formas: a primeira, alternativa adotada no projeto do TorrentLab, é instanciar múltiplos nodos do middleware por máquina e em cada nodo apenas um agente BitTorrent. A segunda, em contraste, é instanciar apenas um nodo por máquina e sobre ele múltiplos agentes BitTorrent. Esta última é desvantajosa porque leva a um sub-aproveitamento de recursos: todos os agentes instanciados sobre um mesmo nodo precisariam deixar o enxame para que a tarefa terminasse e assim liberasse o nodo para ser reaproveitado.

4.3. TorrentSim

Em trabalhos recentes [Konrath et al. 2007b, Konrath et al. 2007a, Mansilha et al. 2007], os autores fazem uso de um simulador de BitTorrent para avaliar aspectos de segurança nesse protocolo. A evolução deste simulador deu origem ao TorrentSim. As propriedades de extensibilidade e modularidade do Simmcast são herdadas pelo TorrentSim, de forma que novas simulações de BitTorrent no TorrentLab são feitas estendendo-se o arcabouço disponível.

O OurGrid provê facilidades para escalonar aplicações do tipo “saco-de-tarefas” (*bag-of-tasks*), que são aplicações paralelas em que o processamento é dividido entre tarefas independentes que não precisam trocar informação entre si. A execução de uma

³<http://status.ourgrid.org/> apresenta o estado da grade em tempo real.

campanha de simulação envolve a execução de diversas simulações independentes e se constitui no exemplo típico de uma aplicação saco-de-tarefa. Dessa forma, o TorrentSim usa diretamente o escalonador de tarefas do OurGrid para executar de forma paralela cada campanha recebida.

4.4. TorrentExp

O ambiente de experimentos implementado presentemente assume que a infra-estrutura distribuída é instanciada em uma rede local com relógios físicos frouxamente sincronizados (com erro máximo da ordem de dezenas de ms). A seguir é discutida a implementação dos elementos que constituem o TorrentExp.

Orquestrador. Este elemento é responsável por escalonar os agentes do usuário para execução na grade computacional. Esses agentes devem ser escalonados de forma a entrarem no enxame nos momentos definidos na descrição do experimento. Para aproveitar a infra-estrutura oferecida pelo OurGrid a execução de um experimento foi transformada na execução paralela de vários jobs na grade, cada uma consistindo de uma única tarefa. Assim, o papel do Orquestrador se resume a adicionar cada job na fila de entrada do escalonador do OurGrid, obedecendo a ordem de tempo de chegada no enxame especificada na configuração do experimento.

Agentes de usuário. Adotou-se como agente padrão o Azureus (3.01), por este ser o cliente mais utilizado atualmente. Parâmetros fixos são previamente configurados em uma instância em local pré-determinado enquanto parâmetros variáveis são configurados em tempo de execução.

Agente rastreador. A instanciação do rastreador foi implementada de outra maneira: executa-se um agente rastreador paralelamente ao ambiente. Isto permite o acompanhamento em tempo real dos enxames instanciados. Para a implementação do protótipo, optou-se pelo rastreador interno do Azureus (3.01).

5. Avaliação do Ambiente

Para entender quão próximos os resultados produzidos pelo TorrentLab estão da realidade, conduziu-se um estudo preliminar cujo objetivo é comparar quantitativamente valores para métricas importantes gerados pelo TorrentSim e TorrentExp. Esta seção apresenta os resultados e conclusões obtidas com esse estudo.

A rede onde foi instanciado o TorrentLab é composta por aproximadamente 30 máquinas (Pentium 4 / 1 GB RAM) em um enlace de 100 Mbps e agrupadas em um par de comutadores. As seguintes configurações foram adotadas: conteúdo de 60 MB, 1 semeador inicial, com largura de banda de download e upload iguais a 2048 Kbps e 512 Kbps, respectivamente, e razão de contribuição 2. O número de sugadores foi variado entre 4 e 128, sendo que os mesmos dispõem de uma largura de banda de 512 e 128 Kbps para download e upload, entram ao mesmo tempo e adotam razão de contribuição alvo 1. Como agente de usuário e rastreador, adotou-se o Azureus (3.01).

5.1. Comparação com experimentos

Esta avaliação comparou o desenvolvimento de um enxame executado via TorrentSim e TorrentExp. Com isto, os mesmos foram “co-validados” ao demonstrar que produzem resultados semelhantes. Uma similaridade entre as curvas indicaria, mas não provaria, que os ambientes retornam valores confiáveis.

De forma indireta (não explorada neste artigo), esta avaliação ajudou a demonstrar a apropriabilidade do Ambiente de Modelagem e de Consolidação, pois ambos foram empregados para criar o experimento e analisar os resultados, neste caso de forma gráfica.

O Ambiente de Modelagem é usado para descrever o enxame e se vale do mesmo conjunto de parâmetros para os casos experimental e simulacional. A aplicabilidade do Ambiente de Consolidação foi indicada ao permitir a definição de métodos de criação de gráficos em apenas um local, de forma integrada.

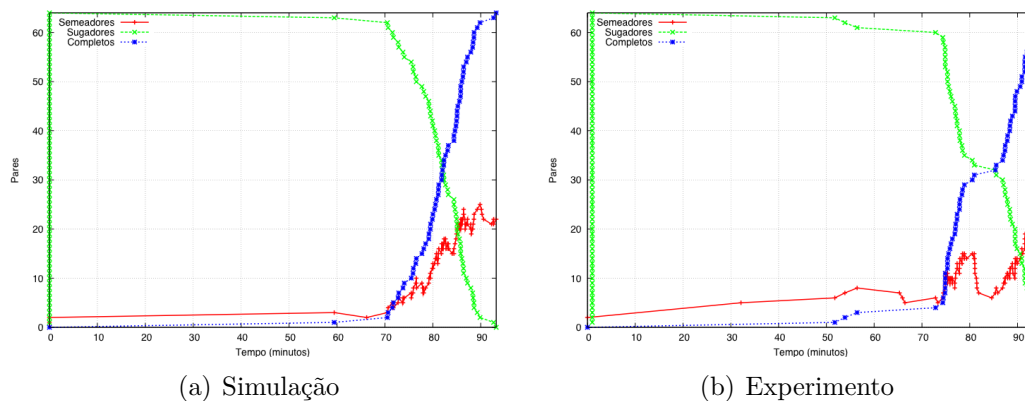


Figura 3. Comparação entre ambientes TorrentSim e TorrentExp

As Figuras 3(a) e 3(b) apresentam a dinâmica de dois enxames, um real e outro simulado, que permitem a comparação entre os dois ambientes de avaliação. As curvas representam a população de pares em termos de semeadores, sugadores, e downloads completos no enxame ao longo do tempo (o eixo X representa tempo em minutos). O enxame definido foi configurado com 64 sugadores e executado sob o TorrentSim e o TorrentExp múltiplas vezes.

O par de gráficos na Figuras 3(a) e 3(b) mostra claramente que, pelo menos para o cenário considerado, os resultados são bastante semelhantes, não apenas no resultado final (tempo médio de download), mas também na evolução do enxame. Apesar de omitidos deste trabalho por uma questão de espaço, os resultados foram igualmente animadores em uma gama de cenários avaliados.

5.2. Avaliação da proporcionalidade de enxames

O objetivo desta subseção é analisar, em caráter preliminar, o impacto de se usar um conjunto limitado de recursos para executar um experimento com um enxame de proporções maiores, bem como comprovar que os resultados produzidos se encontram em linha com o esperado segundo a literatura.

Foram executados experimentos em que se mediu, para diferentes tamanhos de enxame, o tempo médio de download observado por pares. Uma das propriedades mais conhecidas e valorizadas em BitTorrent é sua escalabilidade ([Massouli and Vojnovi 2005]): devido à divisão em peças e a distribuição das mesmas entre pares, o tempo médio de download cresce suavemente no início com o tamanho do enxame e então estabiliza, dependendo das características do enxame tais como fase (*flash crowd*, estável ou em término [Bindal et al. 2006]), razão de contribuição que os pares obedecem, largura de banda de download e upload, e topologias das redes de sobreposição e física.

A Figura 4 apresenta os resultados obtidos com o TorrentSim e TorrentExp, para enxames entre 4 e 128 pares. Os eixos X e Y representam, respectivamente, o tamanho do enxame e tempo médio de download, ambos em escala logarítmica. Como demonstrado na figura, a curva de simulação representa um comportamento suave (correspondente ao que se esperava encontrar) e próximo à curva experimental para enxames entre 8 e 64 pares. Para enxames de 4 e 128 pares, nota-se uma diferença mais acentuada. Para o

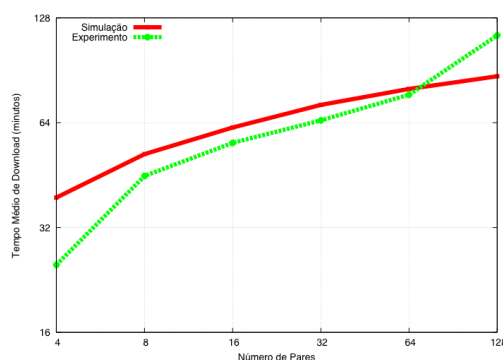


Figura 4. Comparação entre resultados obtidos com TorrentSim e TorrentExp

primeiro caso, não foi considerado significativo o erro dada a peculiaridade de um enxame com 4 pares. Já no segundo caso, houve um crescimento anômalo no tempo médio, gerando uma discrepância notável entre os resultados de simulação e experimental. Após uma campanha de experimentos de avaliação em que estudaram-se os rastros e resultados parciais, concluiu-se que tal se deve a um esgotamento dos recursos físicos (processador, memória e rede) dos equipamentos empregados, bastante limitados.

6. Conclusões e Trabalhos Futuros

BitTorrent está se tornando um padrão de fato no compartilhamento de arquivos e distribuição de conteúdo em larga escala, motivando assim muitos esforços de pesquisa relacionados ao protocolo. A investigação sobre o funcionamento das implementações atuais e a avaliação de propostas de modificações requer o emprego de uma ou mais metodologias: analítica, simulacional ou experimental, preferencialmente em combinação. Entretanto, as dificuldades de montar cenários e gerar múltiplos experimentos em sistemas paralelos e comparáveis, bem como a escassez de recursos, acabam por limitar as investigações a apenas uma técnica.

Neste contexto, TorrentLab é apresentado como ambiente para avaliação integrada do BitTorrent. O mesmo foi estruturado de forma a permitir a extensão para novas classes de agentes, simuladores, plotadores e infra-estruturas computacionais distribuídas. Até onde se sabe, esta é a primeira proposta de um ambiente de avaliação específico para BitTorrent que integra simulação e experimentos em redes reais. A implementação apresentada ilustra as relações entre as partes e serve como prova-de-conceito, contribuindo para demonstrar sua viabilidade.

O ambiente implementado foi utilizado em duas avaliações. A primeira comparou a evolução em termos de população de pares e downloads completados de um mesmo enxame, investigado tanto no TorrentSim como no TorrentExp. Os gráficos demonstram a similaridade entre os dois métodos. A segunda avaliação examinou uma propriedade fundamental do BitTorrent, sua escalabilidade, permitindo uma comparação entre resultados obtidos usando as duas metodologias de avaliação. Os resultados indicam, ainda que em escala limitada, que há uma consistência entre os valores gerados pelo TorrentSim e TorrentLab, e que os mesmos estão em linha com a literatura.

Como trabalhos futuros, são vislumbradas três iniciativas principais. Primeiro, a execução de experimentos adicionais, no sentido de melhor validar o TorrentLab e fazer um ajuste fino do protótipo já em operação. Segundo, reavaliar as questões pertinentes ao Orquestrador: dimensionamento de recursos de hardware, tratando de aspectos como gargalos de rede e capacidade de processamento das máquinas. Terceiro e último, enriquecer o ambiente com a adição de novos agentes de usuário BitTorrent.

Referências

- Barcellos, M., Muhammad, H., and Detsch, A. (2001). Simmcast: a simulation tool for multicast protocol evaluation. In *XIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2001)*, pages 418–433.
- Bharambe, A. R., Herley, C., and Padmanabhan, V. N. (2006). Analyzing and improving a bittorrent networks performance mechanisms. In *25th IEEE International Conference on Computer Communications (INFOCOM 2006)*, pages 1–12.
- Bindal, R., Cao, P., Chan, W., Medved, J., Suwala, G., Bates, T., and Zhang, A. (2006). Improving traffic locality in bittorrent via biased neighbor selection. pages 66–66.
- Das, S., Tewari, S., and Kleinrock, L. (2006). The case for servers in a peer-to-peer world. In *2006 IEEE International Conference on Communications*, volume 1, pages 331–336, Washington, DC, USA. IEEE Computer Society.
- Erman, D., Ilie, D., and Popescu, A. (2005). Bittorrent session characteristics and models. In *Proceedings of the 3rd International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks*, pages P30/1–P30/10.
- Guo, L., Chen, S., Xiao, Z., Tan, E., Ding, X., and Zhang, X. (2007). A performance study of bittorrent-like peer-to-peer systems. *IEEE Journal on Selected Areas in Communications*, 25(1):155–169.
- Konrath, M. A., Barcellos, M. P., and Mansilha, R. B. (2007a). Attacking a swarm with a band of liars: evaluating the impact of attacks on bittorrent. In *The Seventh IEEE International Conference on Peer-to-Peer Computing (IEEE P2P 2007)*. IEEE.
- Konrath, M. A., Barcellos, M. P., Silva, J. F., Gaspary, L. P., and Dreher, R. (2007b). Atacando um enxame com um bando de mentirosos: vulnerabilidades em bittorrent. In *XXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2007)*, volume 2, pages 883–896.
- Mansilha, R. B., Konrath, M. A., and Barcellos, M. P. (2007). Corrupção, mentiras e isolamento: avaliação de impacto de ataques a bittorrent. In *VII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSEG 2007)*.
- Massouli, L. and Vojnovi, M. (2005). Coupon replication systems. In *SIGMETRICS '05: Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, volume 33, pages 2–13, New York, NY, USA. ACM Press.
- Naicken, S., Livingston, B., Basu, A., Rodhetbhai, S., Wakeman, I., and Chalmers, D. (2007). The state of peer-to-peer simulators and simulations. *SIGCOMM Comput. Commun. Rev.*, 37(2):95–98.
- Nussbaum, L. and Richard, O. (2006). Lightweight emulation to study peer-to-peer systems. In *20th International Parallel and Distributed Processing Symposium, 2006 (IPDPS 2006)*, pages 8 pp.+, Washington, DC, USA. IEEE Computer Society.
- Piatek, M., Isdal, T., Anderson, T., Krishnamurthy, A., and Venkataramani, A. (2007). Do incentives build robustness in bittorrent? In *NSDI'07*, Cambridge, MA.
- Sirivianos, M., Park, J. H., Chen, R., and Yang, X. (2007). Free-riding in bittorrent with the large view exploit. In *6th International Workshop on Peer-to-Peer Systems (IPTPS 2007)*, Bellevue, WA, US.