

## Avaliação e Comparação de Simuladores para Redes Peer-to-Peer \*

Guilherme P. Pezzi, Nicolas Maillard,  
César A. F. De Rose

Instituto de Informática - Universidade Federal do Rio Grande do Sul  
CPAD-PUCRS/HP  
{pezzi, nmaillard}@inf.ufrgs.br, derose@inf.pucrs.br

### Introdução

Os ambientes Peer-to-Peer (P2P) vêm ganhando destaque desde o Napster, por não necessitarem de servidor central e permitirem o uso anônimo. Isto torna possível reunir, em uma rede, qualquer máquina conectada na Internet. Atualmente existem inúmeras redes P2P em funcionamento, entre elas *Freenet*, *Gnutella*, *Kazaa*, *eMule*, *Soulseek* e alguma dessas redes chegam a reunir milhões de usuários.

Apesar de a maioria das grandes redes P2P serem utilizadas exclusivamente para compartilhamento de arquivos, seu potencial pode ser utilizado para outras aplicações. Por exemplo, aplicações distribuídas que necessitem de muito poder de processamento poderiam aproveitar as máquinas ociosas da rede para serem executadas.

No entanto, devido a sua dinamicidade e escala, torna-se difícil comparar o comportamento e avaliar os algoritmos usados por estes ambientes (escalabilidade, gargalos de comunicações, etc.). Uma abordagem clássica para permitir comparar estes algoritmos é a simulação. Para obter uma boa medida de comparação a ferramenta de simulação deve ser genérica e permitir simular, no mesmo ambiente (cenário), diferentes algoritmos. A utilização de uma única ferramenta também é uma forma de dar uma certa garantia de uniformização das medidas que serão feitas nos algoritmos.

Este trabalho representa um estudo em andamento sobre avaliação e comparação de ambientes P2P. Inicialmente são definidos os critérios para avaliação e, após, são analisadas algumas das ferramentas existentes. Por fim, é feita a escolha da ferramenta que será utilizada para realizar as comparações.

### Objetivo

Os principais objetivos do presente trabalho são definir os requisitos dos simuladores para redes P2P; fazer um levantamento dos simuladores existentes (*Aurora*, *Chord*, *GridSim*, *SimJava*, *GeSSi*, etc.) e comparar a adequação deles para avaliação de redes P2P.

Na fase posterior deste trabalho, o simulador será usado para avaliar o desempenho de vários algoritmos P2P e poder compará-los. Será feito um estudo da possibilidade de adaptar os mesmos a fim de utilizá-lo para computação de alto desempenho.

---

\*Pesquisa feita em colaboração com a HP Brasil

## Ambientes P2P e simulação

São muitas as redes P2P existentes atualmente. Por exemplo, o *FreeNet* é uma rede totalmente descentralizada, que oferece anonimidade e uma estrutura que garante não haver possibilidade de censura dos seus conteúdos. Já o *eDonkey* utiliza uma hierarquia entre os clientes e destaca-se por ser uma das maiores redes de troca de arquivo em funcionamento.

Dada a escala e complexidade que os ambientes P2P atuais podem atingir, torna-se bastante inviável uma comparação dos algoritmos pela simples execução dos ambientes. Como na maioria dos ambientes em estudo não existe um servidor central, que tenha conhecimento de todos os nós conectados na rede, é muito difícil obter uma “fotografia” da rede em um dado instante de tempo sem alterar o comportamento da mesma. Assim sendo, a simulação se mostra a alternativa mais adequada para a comparação destes algoritmos.

A seguir serão apresentados alguns dos critérios importantes para simulação de ambientes P2P e relacionadas algumas das ferramentas que podem ser utilizadas para fins de simulação.

## Características das ferramentas

Algumas ferramentas foram desenvolvidas especificamente para a simulação de um ambiente. Outras permitem que sejam utilizadas para simular um ambiente genérico, ou seja, não são focadas em um algoritmo específico. As simulações utilizam uma lista de eventos que são notificados aos nós. É necessário definir um cenário, ou seja, a sequência de eventos que serão simulados.

Neste trabalho, considera-se alguns critérios para avaliar as ferramentas de simulação. A **forma de descrição do cenário**, que pode ser diretamente no código ou separadamente através de uma linguagem ou gramática fornecida pela ferramenta. Quanto mais simples a forma de representação do cenário, mais fácil fica para o usuário criar diferentes cenários. Depois, as ferramentas podem oferecer **diversas métricas** ou formas de se implementar novos métodos de medição do cenário simulado. Uma boa maneira de certificar que os diferentes algoritmos sejam submetidos às mesmas métricas é utilizando uma única ferramenta de simulação; A **forma de descrição do algoritmo** é importante, pois a descrição precisa ser flexível, permitindo que seja simples a especificação de algoritmos na ferramenta. Desta forma é possível comparar facilmente diversos algoritmos utilizando a mesma ferramenta e, portanto, com o mesmo conjunto de cenários e métricas de avaliação. Por fim, considera-se a **escalabilidade**: uma vez que os ambientes P2P são capazes de agregar recursos na ordem de até milhões de nós, a ferramenta deve ser capaz de suportar cenários com uma quantidade de nós realista. Neste texto é considerado um simulador escalável caso ele seja capaz de simular um número crescente e alto de nós (mesmo sem recorrer ao paralelismo).

## Principais ferramentas

A Tabela 1 resume as principais características das ferramentas já consolidadas e conhecidas para simulação: o Aurora [ORA 2001] é o simulador usado pelo FreeNet. O Chord [STO 2001] é um projeto do MIT de um algoritmo distribuído para o *lookup* numa rede P2P, que disponibilizou um simulador desenvolvido em C. O SimGrid [BUY 2002] é um simulador para ambientes de Grid desenvolvido por Buyya.

Tabela 1: Principais ferramentas existentes para simulação de ambientes P2P

Ferramenta	Linguagem	Algoritmo	Nós simulados
Aurora	C++	FreeNet	200.000
Chord	C	Chord	16.384
GridSim	Java	Genérico	68

Desta tabela, e conforme os requisitos mencionados, vê-se que o *Aurora* e o *Chord* não são adequados para este trabalho, pois não oferecem recursos genéricos para possibilitar a programação de diferentes algoritmos em um simulador só. O *SimGrid* não parece cumprir o critério de escalabilidade que se requer, isso porque os seus desenvolvedores privilegiaram definições de alto nível dos recursos simulados. Já quanto à forma de descrição de cenário, apenas o *Chord* disponibiliza uma gramática de alto nível para facilitar sua programação.

Por essas razões, decidiu-se utilizar um simulador experimental, chamado GeSSi.

## GeSSi

O *GeSSi* é uma ferramenta experimental e está sendo desenvolvida na linguagem *Java*. O *GeSSi* se propõe a ser uma ferramenta genérica e escalável e utiliza uma estrutura baseada em eventos, que podem ser descritos utilizando uma gramática específica. Esta ferramenta também permite que sejam programados e simulados diversos algoritmos utilizando uma API genérica. O *GeSSi* já está funcional e foi usado em 2003 para simular o ambiente *I-Cluster* [RIC 2003].

A Figura 1 mostra um exemplo da utilização de memória do *GeSSi* em função do número de eventos carregados pelo simulador em um cenário com 100 mil nós, numa rede *I-Cluster*. Este trabalho foi realizado no CPAD (Centro de Pesquisa em Alto desempenho), através do convênio FACIN-PUCRS/HP Brasil. O fato do *GeSSi* utilizar pouca memória para simular tantos nós valida sua escalabilidade.

## Considerações Finais

Como foi mostrado na seção anterior, não se encontrou uma ferramenta adaptada que disponha de todos requisitos para permitir a comparação de algoritmos de forma simples. As ferramentas que utilizam o modelo sequencial (*Aurora* e *Chord*) são específicas

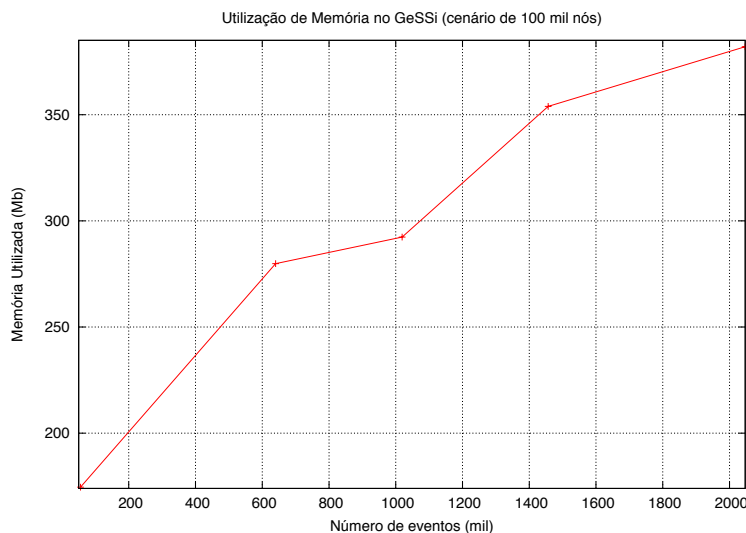


Figura 1: Utilização de memória no GeSSi

para um algoritmo, enquanto as genéricas (*GridSim*) possuem uma limitação no que diz respeito ao número máximo de nós simulados.

Dado o contexto, optou-se pela utilização da ferramenta *GeSSi* uma vez que ela é a que melhor preenche os requisitos para simular diferentes algoritmos com uma quantidade de nós que se aproxime das realidade das redes P2P atuais.

Como continuação deste trabalho, será feita uma validação do *GeSSi* através da especificação e simulação do algoritmo da rede *FreeNet*, bem como a comparação do desempenho do mesmo com outros algoritmos P2P.

## Referências

- [BUY 2002] BUYYA, R.; MURSHED, M. Gridsim: a toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. **Concurrency and computation: practice and experience**, v.14, p.1175–1220, 2002.
- [ORA 2001] ORAM, A. **Peer-to-peer, harnessing the power of disruptive technologies**. [S.l.]: O'Reilly, 2001.
- [RIC 2003] RICHARD, B. **I-cluster**: agrégation des ressources inexploitées d'un intranet et exploitation pour l'instanciation de services de calcul intensif. 2003. Tese (Doutorado em Ciência da Computação) — INPG.
- [STO 2001] STOICA, I. et al. Chord: a scalable peer-to-peer lookup service for internet applications. In: **ACM SIGCOMM 2001**, 2001. **Proceedings...** [S.l.: s.n.], 2001.