

1

Introdução

A grande capacidade de recursos que computadores atuais têm apresentado está mudando a forma como a computação é realizada. Centros de processamento de dados, onde máquinas eram dedicadas ao uso de uma única instituição, estão sendo integrados a infraestruturas distribuídas, formando grades computacionais, ou mesmo substituídos pelo acesso a recursos presentes nas chamadas nuvens computacionais [1, 2]. Uma das principais motivações para essa mudança consiste no melhor aproveitamento dos recursos das máquinas, reduzindo assim custos com refrigeração, manutenção e alocação de espaço físico.

Tanto em grades como em nuvens computacionais, uma prática comum é a utilização de uma única máquina física para a execução de diferentes aplicações. Essa prática de compartilhamento acarreta alguns desafios para garantir o desempenho das aplicações gerenciadas. Garantir que as aplicações de uma infraestrutura compartilhada não violem as políticas de uso, assim como garantir um desempenho mínimo esperado para essas aplicações, pode ser uma tarefa difícil quando não há mecanismos que limitem e garantam o uso de recursos nos servidores.

Na relação sistemas operacionais e aplicações, o sistema atua como o provedor de serviços, pois disponibiliza o uso de recursos de processamento, memória e operações de E/S. As aplicações, por sua vez, atuam como usuárias desses serviços, podendo requisitar diferentes qualidades de serviço (do termo inglês *Quality of Service* — QoS) ao sistema operacional. Frequentemente, a qualidade de serviço é quantificada pelo montante de serviço a ser provido em um determinado intervalo de tempo [3]. Dessa maneira, uma funcionalidade importante em sistemas de computação para se atingir QoS consiste no uso de reservas de recursos computacionais [4, 5].

Sistemas Operacionais de Propósito Geral (do termo inglês *General Purpose Operating System* — GPOS) não proveem meios para garantir qualidade de serviço. Com eles não é possível, por exemplo, especificar o montante de processamento que uma aplicação deve ser habilitada a consumir em um intervalo de tempo. Porém, as vantagens que o compartilhamento de servidores

é capaz de trazer têm feito aumentar a busca por maneiras de isolar o desempenho de aplicações em máquinas compartilhadas e que fazem uso de GPOSes. Algumas propostas para a solução desse problema incluem o uso de virtualização, a extensão do núcleo do sistema operacional (SO) com abstrações de reserva e o uso de ferramentas no nível do usuário.

Nos últimos anos, uma combinação de novas necessidades e tecnologias causou o ressurgimento do interesse pela técnica de virtualização [6, 7]. Trabalhos recentes mostram que essa técnica tem sido amplamente utilizada como uma solução para a provisão de qualidade de serviço, principalmente por possibilitar a construção de ambientes de execução isolados tanto por desempenho quanto por dados [8, 9, 10, 11, 12]. Por outro lado, muitas vezes, o uso de uma camada extra de *software* para a realização da virtualização torna-se inapropriado devido, por exemplo, a questões de sobrecarga no uso de recursos [13, 14, 15]. Extensões de núcleo do SO, por sua vez, apresentam sobrecarga mínima, mas podem não ser adequadas quando há indisponibilidade ou complexidade do código do sistema operacional a ser alterado. Instrumentar o código do sistema operacional para prover abstrações de reserva requer a modificação do escalonamento de processos, uma atividade essencial do sistema que, se mal implementada, pode tornar o SO inoperante como um todo.

Existe ainda a opção do uso de ferramentas que utilizam mecanismos providos pelo próprio sistema operacional para gerenciar o acesso aos recursos, sem ocasionar grandes sobrecargas de desempenho e nem alterações no núcleo do SO. A prática de gerenciamento de recursos adotada por essas ferramentas é mais próxima à abordagem de total separação de mecanismos e de políticas defendida por muitos pesquisadores da área de Sistemas Operacionais, incluindo Tanenbaum [7, 16] e Levin [17]. Esses pesquisadores argumentam a favor da clara divisão entre os mecanismos, providos pelo núcleo do SO, e as políticas, implementadas no nível do usuário.

Entre as desvantagens do uso de ferramentas no nível do usuário está o fato dessas ferramentas, tipicamente, só poderem ser utilizadas para o gerenciamento de aplicações do tipo *soft real-time*, isto é, aplicações que não necessitam de reservas rígidas de recursos. Porém, desde que as causas dessas desvantagens sejam bem entendidas e as suas restrições de uso respeitadas, ferramentas no nível do usuário podem ser de grande utilidade em diversos cenários atuais de computação, tais como grades oportunistas e nuvens privadas.

A garantia de qualidade de serviço em servidores é altamente dependente do gerenciamento simultâneo de múltiplos recursos computacionais, pois a ausência de controle de um determinado recurso que se encontra escasso na máquina pode causar uma inversão de prioridade nas aplicações em execução.

Considere, por exemplo, uma aplicação CPU-intensiva que realiza acessos esporádicos ao disco rígido. Garantir uma fatia de processamento a essa aplicação pode melhorar o desempenho obtido, mas não garante a qualidade de serviço esperada caso não haja, juntamente com outras medidas, um controle concomitante de acesso ao disco, isto é, pouco adianta priorizar o acesso à CPU se os processos da aplicação serão tratados de maneira indistinta no acesso à mídia de armazenamento. No entanto, o que se vê é que o controle de recursos é realizado apenas parcialmente nos sistemas de gerenciamento no nível do usuário [18, 19, 20, 21, 22]. Muitas vezes, somente o processamento é controlado pelos gerenciadores de recursos, fato parcialmente explicado porque, em geral, o processamento consiste no gargalo da aplicação e porque controlar a E/S de dados na rede, no disco e em memória tende a ser uma tarefa mais complexa e de comportamento menos previsível. Nessa tarefa há de se considerar, por exemplo, a influência que o controle de E/S irá exercer sobre o desempenho da aplicação e a saturação de E/S do dispositivo considerado.

Conclui-se, portanto, que há uma lacuna a ser preenchida na área de gerenciamento de recursos no nível do usuário, mais especificamente nas tarefas de orquestrar o uso concomitante de diferentes recursos computacionais, assim como na provisão de facilidades para a flexibilização das regras de compartilhamento de recursos. Assim, é importante investigar as possibilidades e limitações de se implementar ferramentas de reserva de recursos no nível do usuário em sistemas operacionais atuais, verificando-se a efetividade dessas ferramentas e a sobrecarga de trabalho gerada por elas.

1.1 Objetivos, Abordagem e Contribuições

Este trabalho tem como objetivo estudar a viabilidade de se realizar reserva concomitante de recursos no nível do usuário, tendo como base os mecanismos tipicamente oferecidos pelos sistemas operacionais de propósito geral atuais. Esse estudo deve identificar vantagens e limitações das técnicas empregadas para o controle do uso de recursos e, assim, definir os cenários onde esse tipo de reserva pode ser empregado com sucesso. Para isso é preciso conduzir uma investigação a respeito da complexidade de implementação das técnicas utilizadas nos mecanismos de reserva, da sobrecarga causada por elas e de como as reservas devem ser gerenciadas para que sejam, de fato, garantidas.

A opção de priorizar a implementação de um código fora do núcleo do SO, em detrimento da precisão no controle de uso dos recursos, tal como acontece em mecanismos de gerenciamento implementados no nível do usuário, foi motivada pelos benefícios que as reservas trariam em cenários de computação

onde a característica de tempo real não está fortemente presente nas aplicações e onde a instalação de novos SOs ou máquinas virtuais é uma prática inviável ou inconveniente de ser realizada. Um exemplo de cenário assim pode ser encontrado em empresas cujas máquinas servidoras são compartilhadas entre diferentes usuários de uma mesma equipe.

Devido à existência de deficiências no controle dos recursos, é importante que o estudo proposto gere conhecimento suficiente para responder perguntas relativas ao grau de confiabilidade das reservas implementadas no nível do usuário. Para isso, é preciso uma ferramenta de apoio para a realização dos experimentos. Entre as ferramentas já existentes, nenhuma atendia integralmente aos nossos requisitos de limitar o consumo de recursos quando a taxa limite permitida para uso já foi consumida, e de garantir o acesso ao recurso quando a requisição está dentro da cota estimada de uso. Assim, optamos por implementar uma nova ferramenta que provesse tanto mecanismos para a limitação e garantia do uso de processamento como para a limitação e garantia do uso de largura de banda de disco, recursos identificados neste trabalho como sendo fontes de frequentes contenções em servidores compartilhados. Também optamos pela inserção na ferramenta de mecanismos para a extensão das regras de compartilhamento de recursos. Essas regras, chamadas de políticas de escalonamento, podem ser ativadas a cada inicialização do servido.

Para verificar a eficácia da ferramenta desenvolvida, é preciso que a aplicação a ter o consumo de recursos limitado faça uso intensivo dos recursos gerenciados pela ferramenta. Aplicações multicamadas apresentam esse perfil. Com esse tipo de aplicação, é possível avaliar a eficácia de reservas isoladas e de reservas concomitantes de recursos de processamento, disco e rede. Porém, para validar as reservas providas é preciso assegurar que gargalos de desempenho gerados por planejamentos de desempenho mal feitos não ocorram. Assim, ao considerar uma aplicação de múltiplas camadas é necessário definir um método para a decomposição de qualidade de serviço, ou seja, é preciso estabelecer uma forma de mapear requisitos de alto nível como, por exemplo, interações por segundo em diferentes requisitos de mais baixo nível, tais como taxa de processamento e larguras de banda de disco e de rede, a serem atribuídas aos diversos componentes da aplicação.

Por não se tratar do objetivo principal desta tese, o método de decomposição de QoS usado nos experimentos deve ser simples, para que seja facilmente implementado, e efetivo, para que reflita de maneira confiável valores mínimos de reservas de recursos. Porém, o método empregado pode ser definido sem a preocupação de produzir resultados similares ou melhores que os encontrados no estado da arte. A ideia é que os valores gerados na decomposição de

QoS sirvam de base para a parametrização inicial dos montantes de recursos que uma aplicação necessita, sem que gargalos de desempenho sejam artificialmente gerados e erroneamente atribuídos ao mal funcionamento da ferramenta de reservas.

Com a abordagem bem definida de realizar todo o gerenciamento de recursos fora do núcleo do SO e sem a ciência da aplicação, até mesmo a fase de decomposição de QoS deve acontecer sem instrumentações do código original da aplicação.

As contribuições deste trabalho incluem:

- Os resultados do estudo investigativo sobre as possibilidades de se implementar mecanismos de reserva de recursos no nível do usuário, identificando as limitações impostas por essa prática, assim como as vantagens de se utilizá-la. O estudo deve esclarecer dúvidas sobre: a possibilidade de se garantir e limitar o uso de recursos com técnicas implementadas sem a instrumentação de códigos do sistema operacional; as dificuldades de se implementar o controle de E/S de dados no nível do usuário; as formas de se utilizar os mecanismos implementados; o grau de confiabilidade do controle de consumo; e as limitações dos mecanismos de reserva as quais impõem restrições aos cenários em que esses mecanismos podem ser utilizados.
- Uma ferramenta capaz de prover reservas de recursos de processamento e largura de banda de disco. Com o diferencial de ser totalmente implementada no nível do usuário e por apresentar políticas de escalonamento flexíveis e de fácil implantação, essa ferramenta pode ser utilizada em muitos cenários de computação compartilhada.

Indiretamente, este trabalho também avalia um método simples para a decomposição de QoS. O método proposto, combinado com o mecanismo de reserva de recursos, é capaz de parametrizar reservas a partir de requisitos de qualidade de alto nível, mapeando esses requisitos em montantes de recursos de processamento e largura de banda de disco.

Parte das contribuições desta tese já se encontra publicada [23, 24, 25] e sendo utilizada em outros trabalhos [26, 27].

1.2 Estrutura da Tese

Este documento está organizado da seguinte forma. Inicialmente, no Capítulo 2 são apresentados os principais trabalhos relacionados com esta tese, abordando as características desses trabalhos que, de alguma forma, inspiraram

o nosso estudo. O Capítulo 3 é dedicado à descrição da ferramenta de reserva de recursos proposta, apresentando detalhes de como essa ferramenta foi implementada e de quais fatores influenciaram na escolha das técnicas utilizadas para o controle do uso de recursos. No Capítulo 4 são apresentados experimentos para a verificação da eficácia e a avaliação de desempenho da ferramenta de reserva desenvolvida. No mesmo capítulo, é apresentado um estudo de caso em que a ferramenta implementada gerencia o consumo de recursos de uma aplicação multicamadas e que faz uso simultâneo de recursos. Nesse estudo, um método para a decomposição de QoS é apresentada como uma maneira simples de parametrizar reservas. Por fim, no Capítulo 5 são apresentadas as conclusões deste trabalho juntamente com uma revisão das contribuições desta pesquisa e a indicação de trabalhos futuros.