

LISTA DE SIGLAS

HCP – High-Performace Computing - Computação de Alto Desempenho

DCG – Desktop Computer Grid - Grade de Computadores Domésticos

SEA – Software Engineering Artifact - Artefato de Engenharia de Software

SIMD – Single Instriction, Multiple Data - Unica Sequência de Instruções para Multiplas

Fontes de Dados

GPGPU – General Purpose Graphic Processing Unit - Unidade de Processamento Gráfico de Propósito Geral

P2P – Peer 2(to) Peer - Ponto à Ponto

SUMÁRIO

1 INTRODUÇÃO	3
1.1 Motivação	3
1.2 Problema	4
1.3 Proposta	4
1.3.1 <i>Objetivos</i>	4
1.4 Justificativa dos Objetivos e sua Relevância Científica	5
Referências Bibliográficas	6

1 INTRODUÇÃO

A prática da simulação em ambientes virtuais se uniu à revisão literária e a experimentação como terceiro pilar do processo de desenvolvimento científico e industrial (CIRNE et al., 2006). Tais simulações são um exemplo das incontáveis aplicações que processam grandes quantidades de dados e que precisam de resultados o mais rápido possível (à esta categoria de aplicações se dá o nome de "computações de alto desempenho- HPC), seja na academia ou na indústria. Pode-se prover cada usuário com acesso a um dispositivo capaz de realizar tais computações, o que é dispendioso, ou usar soluções como computadores de grande porte (*Mainframes*), computação em *cluster*, computação em nuvem e computação em grade.

Uma solução que envolve computadores de grande porte é focada no compartilhamento de uma única grande máquina entre diversos usuários, o que é eficiente e dispendioso. Computação em nuvem é um modelo muito conhecido pelo público geral, englobando plataformas como *Google Cloud* e *Amazon Web Services*. Neste modelo, softwares inteiros são executados em ambiente remoto pertencente à uma certa organização, que submete ao usuário apenas suas vias de entrada de dados e saída de resultados, não sendo necessário que este realize alguma contribuição computacional significativa. Computação em grade é um modelo parecido - pedidos de computações são enviados para serem executados em máquinas parceiras. Contudo, máquinas distintas podem pertencer à organizações distintas, e cada membro da rede pode disponibilizar seus dispositivos para computações de outrem, aumentando a capacidade do sistema como um todo. Máquinas numa grade podem operar independentemente sem um controle central de processamento, necessário no modelo de *cluster*.

Uma grade pode ser implementada utilizando apenas computadores domésticos, como as encontradas em residências e ambientes de trabalho (este modelo é denominado DCG). Assim, uma organização pode compartilhar e otimizar o aproveitamento do potencial computacional do parque de dispositivos que já possui. (CEDERSTRÖM, 2010; SCHWIEGELSHOHN et al., 2010; CIRNE et al., 2006)

1.1 Motivação

No contexto de demanda muito superior à oferta, é pouco produtivo que indivíduos desperdicem quantidades significativas de recursos. Contudo, o uso típico de uma máquina doméstica, mesmo na indústria, é caracterizado por longos períodos de relativamente baixa atividade entremeados por surtos de demanda de computações, para as quais os recursos de uma máquina típica são normalmente insuficientes (CITATION NEEDED). A capacidade computacional não utilizada de uma máquina durante seu período de baixa atividade é considerada desperdiçada. Este trabalho foi inspirado pelo desejo de aproveitar tal potencial, otimizando o uso de recursos de uma organização ou indivíduo.

1.2 Problema

No atual ecossistema de projetos DCG, poucos artefatos de engenharia de software (SEAs) são disponibilizados para o público (SCHWIEGELSHOHN et al., 2010). Este é um obstáculo à ser superado para facilitar a análise dos custos de implantação e desenvolvimento de soluções por DCG.

A arquitetura de todos os sistemas citados nos trabalhos correlatos analisados é baseada na submissão de códigos portáteis para execução em máquinas de parceiros (ANDERSON, 2003; CIRNE et al., 2006; CEDERSTRÖM, 2010; SCHWIEGELSHOHN et al., 2010; ZHAO; LIU; LI, 2011). Diversos sistemas utilizam virtualização do ambiente de execução para executar os códigos de parceiros, o que não só esconde as características individuais de cada máquina (e o potencial computacional das mesmas) como adiciona mais uma camada de programas que consomem recursos (THAIN; TANNENBAUM; LIVNY, 2005; ZHAO; LIU; LI, 2011; CIRNE et al., 2006). O paradigma de portabilidade, portanto, limita a eficiência das computações, pois impossibilita otimizações de código personalizadas para a arquitetura cada máquina, e também o uso de recursos aceleradores de execução atípicos como co-processadores SIMD e GPGPUs.

1.3 Proposta

Neste trabalho são apresentados os artefatos de engenharia de software e realizada a implementação de um sistema denominado *Assistance*, um aplicativo que permite que uma máquina envie, diretamente, problemas lógicos para serem solucionados por parceiros. *Assistance* difere de uma grade de computadores típica por ser baseado numa arquitetura de passagem de mensagens e implementações locais, em oposição ao modelo de virtualização e portabilidade de códigos executáveis. Cada máquina da grade possui sua própria implementação de uma seleção de algoritmos e aplicações genéricos e comuns, executados sobre dados enviados por máquinas parceiras. Esta arquitetura permite mais flexibilidade no uso dos diferentes recursos de cada máquina, e mais segurança para seus proprietários, uma vez que não são executados códigos de terceiros.

1.3.1 Objetivos

1. Produzir todos os AESs necessários para descrever a arquitetura e princípios lógicos do sistema *Assistance*;
2. Implementar *Assistance* como um sistema de passagem de mensagens P2P para realizar DGC;
3. Medir o tempo de implantação da plataforma *Assistance* em diferentes ambientes de rede;
4. Medir a performance de diversas aplicações na plataforma *Assistance* em diversos ambientes de execução;

5. Avaliar a dificuldade de implementação de uma aplicação compatível com a plataforma *Assistance*
6. Medir a dificuldade de adaptar uma aplicação já existente de forma a se tornar compatível com a plataforma *Assistance*;

1.4 Justificativa dos Objetivos e sua Relevância Científica

O objetivo primário deste trabalho se deve à nota deixada por (SCHWIEGELSHOHN et al., 2010), que ressaltam a necessidade de mais AESs para que soluções de DGC possam ser melhor estudadas. (THAIN; TANNENBAUM; LIVNY, 2005) reforça tal noção, afirmando que o núcleo do sistema Condor não foi alterado desde 1988, responsabilizando as boas práticas de engenharia de software envolvidas na elaboração do sistema.

O segundo, quarto, quinto e sexto objetivos são voltados para a validação científica e viabilidade funcional do paradigma de passagem de mensagens sem distribuição de códigos executáveis, uma vez que tal arquitetura não foi utilizada em quaisquer trabalhos correlatos analisados.

O terceiro objetivo foi inspirado pela informação mencionada por (CEDERSTRÖM, 2010), sobre o custo de implantação de um sistema de DGC simples. No caso, o custo, em tempo, foi considerado inadequado para outros contextos de uso, como dispositivos móveis e redes "*in-promptu*", redes organizadas circunstancialmente.

Referências Bibliográficas

ANDERSON, David P. Public computing: Reconnecting people to science. In: **Conference on Shared Knowledge and the Web**. [S.l.: s.n.], 2003. p. 17–19.

CEDERSTRÖM, Andreas. **On using Desktop Grid Computing in software industry**. 2010. Dissertação (Mestrado) — School of Engineering, Blekinge Institute of Technology, Ronneby, Suécia.

CIRNE, Walfredo et al. Labs of the world, unite!!! **Journal of Grid Computing**, Springer, v. 4, n. 3, p. 225–246, 2006.

SCHWIEGELSHOHN, Uwe et al. Perspectives on grid computing. **Future Generation Computer Systems**, Elsevier, v. 26, n. 8, p. 1104–1115, 2010.

THAIN, Douglas; TANNENBAUM, Todd; LIVNY, Miron. Distributed computing in practice: The condor experience. **Concurrency and Computation: Practice and Experience**, Wiley Online Library, v. 17, n. 2-4, p. 323–356, 2005.

ZHAO, Han; LIU, Xinxin; LI, Xiaolin. A taxonomy of peer-to-peer desktop grid paradigms. **Cluster Computing**, Springer, v. 14, n. 2, p. 129–144, 2011.