

Melhorando o Desempenho de SimPoints Utilizando Técnicas de Aprendizado de Máquina Não Supervisionado

Michelle K. Hamada, Denis G. Fantinato, Emilio Francesquini

Centro de Matemática, Computação e Cognição
Universidade Federal do ABC (UFABC)

michelle.hamada@aluno.ufabc.edu.br
denis.fantinato@ufabc.edu.br, e.francesquini@ufabc.edu.br

Resumo. *Em pesquisas voltadas para o desenvolvimento de arquiteturas de hardware são utilizadas simulações e benchmarks cujas execuções podem levar até alguns meses. A metodologia SimPoints, empregada para contornar os longos tempos de execução, é capaz de estimar os resultados da simulação de modo eficiente com alto grau de precisão. Utilizando o K-means, um número reduzido de SimPoints é escolhido automaticamente para representar a execução completa do programa. Porém, existem vias alternativas ao K-means que podem ser executadas. Neste artigo avaliaremos o uso dos Mapas Auto-Organizáveis de Kohonen na escolha dos SimPoints tendo como métrica aspectos como o tempo de execução, o número de SimPoints e a precisão da simulação.*

1. Introdução

O desenvolvimento de novas arquiteturas de hardware requer modelos de simulação cuja execução utilizando benchmarks pode levar meses dependendo do detalhamento das informações desejadas. Visando à redução desses longos tempos pode-se empregar a metodologia SimPoints [Hamerly et al. 2005], em que é possível estimar com boa precisão, e com apenas uma fração do tempo original da simulação, o comportamento do hardware.

No processo, determinam-se as regiões mais representativas da execução completa de um programa para estipularmos o seu comportamento. Para tanto, a execução do programa é dividida em blocos básicos e a sua execução representada por um vetor de blocos básicos (BBVs [Sherwood et al. 2003]) que indica para cada fase do programa quais blocos foram executados. Os índices do vetor representam cada um dos blocos básicos do programa e os valores quantas vezes aquele bloco foi executado durante um dado intervalo do programa. Assim, através da comparação entre BBVs de dois trechos de um programa, podemos comparar os seus comportamentos de maneira indireta através da distância de Manhattan. Em seguida os BBVs são agrupados utilizando K-means.

Entretanto, para uma escolha mais adequada dos SimPoints, métodos alternativos ao K-means podem ser empregados. Nesse artigo, propomos a utilização dos Mapas Auto-Organizáveis de Kohonen, conhecido também como SOM, para a geração de agrupamentos de forma mais autônoma, que não dependam fortemente da escolha inicial dos parâmetros. Isto permitiria, por exemplo, uma escolha automática do número de SimPoints. Dessa forma, esperamos que este se apresente como alternativa de maior eficiência em relação ao K-means ao ponderar aspectos como o tempo de execução, o número de SimPoints e a precisão da simulação de forma automática.

2. Simulation Points e Trabalhos Relacionados

Com a metodologia SimPoints proposta por Sherwood [Sherwood et al. 2003], é possível identificar as fases de execução mais representativas para estimar a execução completa do programa. Ela utiliza como base a observação que programas típicos têm diversas fases de execução em que estas são repetitivas. Portanto, temos uma maneira eficiente e precisa para contornar os longos tempos de execução ao se realizar combinações cuidadosas dos resultados da simulação da execução dos SimPoints. Mediante os resultados temos com um bom nível de precisão o comportamento do hardware para a execução completa do programa a ser avaliado. Portanto, é possível reduzir o tempo das simulações de semanas ou meses para poucas horas. O avaliador precisa apenas definir a relação entre a precisão e o tempo de execução, uma vez que o número de instruções de hardware a ser simulado utilizando esta abordagem é significativamente menor do que o necessário para a execução completa do benchmark.

Alguns trabalhos como o SMARTS [Wunderlich et al. 2003] tentam estimar o intervalo de confiança da simulação de forma mais precisa do que os SimPoints que, apesar de apresentar excelente precisão na prática, não oferece garantias sobre a precisão alcançada. Além disso, existem simuladores como Zsim [Sanchez and Kozyrakis 2013] e o Sniper [Carlson et al. 2011] que dão suporte ao uso de SimPoints ao acelerarem a execução de simulações utilizando uma técnica denominada *weave-and-bound* que consiste em paralelizar a execução quando a arquitetura possui mais de um núcleo de processamento. Contudo, esta técnica pode introduzir erros adicionais de precisão advindos desta divisão do trabalho entre os vários processadores.

Apesar do SimPoints ter sido amplamente empregado, utiliza-se apenas uma única estratégia para a selecionar os pontos que serão usados pela simulação. No momento da classificação em fases, emprega-se uma simples métrica de similaridade e o algoritmo K-means para agrupamento dos intervalos de execução. Os bons resultados desta combinação podem ser a razão pela qual, até onde pudemos averiguar, não há na literatura estudos detalhados que descrevam como diferentes escolhas de métricas de similaridade e algoritmos de agrupamento influenciam a precisão e eficiência dos SimPoints e, consequentemente, das simulações que os utilizam.

3. Proposta e Metodologia

Buscando avaliar o impacto que diferentes escolhas de métricas de similaridade e diferentes algoritmos não supervisionados de aprendizado de máquina têm no desempenho e na precisão de simulações baseadas em SimPoints, implementamos o algoritmo K-means e o SOM para testar o volume de dados de alta dimensionalidade dado no processo de simulação para determinar os SimPoints.

O K-means requer parâmetros iniciais para a execução do seu algoritmo, uma vez que toda iteração é baseada na quantidade de centroides que será dada. O algoritmo inicia determinando o centroide de cada cluster de acordo com o valor fornecido pelo usuário e calcula o cluster a qual pertence cada dado. Assim, com as novas amostras de dados em cada cluster, acham-se os pontos que mais representam os novos conjuntos de dados dentro dos clusters e atualizam-se os centroides com estes pontos até que não haja mais alteração dos clusters. Contudo, o K-means depende do número de centroides a ser fornecido, da ordem de apresentação e inicialização dos dados. Estas são todas determinadas

pelo projetista, classificando-se assim nos principais problemas do K-means.

Existem, contudo, outras técnicas de aprendizado de máquina não supervisionado capazes de reduzir a dependência desses parâmetros. Em especial, o SOM, que além de fazer a clusterização, também mostra-se útil no problema de redução de dimensionalidade. O SOM busca estabelecer e preservar noções de vizinhança (preservação topológica) ao utilizar uma rede neural (em geral, bidimensional) para visualização e entendimento dos dados de forma que minimize a perda de informação [Kohonen 1998, Zuchini 2003]. A metodologia é baseada em aprendizagem competitiva que estabelece uma relação entre a vizinhança e os pesos dos neurônios no espaço de dimensão igual ao número de entradas. Utilizam-se três etapas: competição, aprendizado e discriminação dos agrupamentos. Na primeira etapa de competição, há uma concorrência que busca estabelecer o neurônio vencedor. Na segunda etapa, a de aprendizado, o neurônio que venceu terá seu peso e de sua vizinhança ajustado conforme a regra de influência que existe para a ativação de um neurônio a todos os demais. Finalizado o aprendizado, a conformação final dos neurônios fornecerá as informações necessárias para a discriminação de agrupamentos.

Como uma ferramenta robusta para análise e visualização de dados, o SOM apresenta diversas características relevantes para as atividades de análise dos dados a serem realizadas. Dentre as propriedades, o algoritmo tem a capacidade de representação da estrutura presente no espaço de dados em que é feita uma projeção não linear do espaço de dados de entrada em R^d para o espaço do arranjo em R^p , onde d é a dimensão do espaço de dados e p a dimensão do espaço da projeção (dos neurônios). Paralelamente, o algoritmo busca ao máximo preservar a topologia do espaço original mantendo uma relação de vizinhança entre os neurônios, pode-se assim operar com conjuntos volumosos de dados em que os dados podem possuir alta dimensionalidade. Outra característica é a possibilidade de análise da visualização, de detecção de agrupamentos e das suas relações ao se utilizar a matriz-U. A matriz-U (matriz de distâncias unificadas) possibilita a avaliação gráfica dos resultados obtidos, sendo gerada pela média aritmética das distâncias sobre os vetores de pesos de toda a vizinhança do neurônio e o seu próprio vetor de pesos.

4. Conclusão e Trabalhos Futuros

Diante dos pontos observados entre os algoritmos, propomos a avaliação do impacto causado pela substituição do K-means pelo SOM que, apesar de ter maior complexidade em sua execução, têm o potencial de gerar SimPoints mais precisos e, assim, obter estimativas mais confiáveis do comportamento da execução. Com o uso do SOM, esperamos suprir um dos principais problemas do K-means: a necessidade de um valor inicial para que o algoritmo atribua os centroides para cada agrupamento como é possível ver na Figura 1. Nela, foi determinado para o K-means a quantidade de 20 centroides e, para o SOM, 20 neurônios numa grade de 4x5 (ambos representados por pontos pretos). No K-means, os centroides se aglomeraram cada um em um determinado agrupamento dos dados sem ficar nítido o número de clusters, enquanto que, no SOM, os neurônios se agruparam em dois locais no espaço de forma que temos claramente a detecção de dois clusters.

O SOM realiza de forma automática os agrupamentos sem a necessidade de um valor determinando a quantidade de agrupamentos na inicialização do algoritmo. Contudo, há na literatura [Zuchini 2003] descrições sobre como melhorar e tornar os resultados encontrados mais eficientes e que abranjam melhor o plano dos dados conforme ocorrem

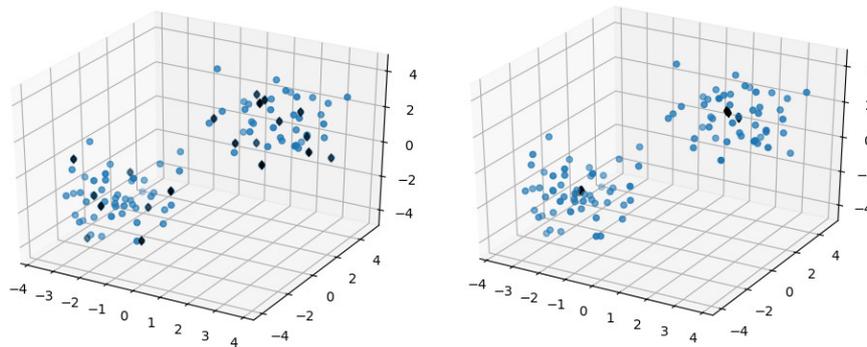


Figura 1. Centroides do K-means vs neurônios do Mapa Auto-Organizável de Kohonen em um conjunto de dados que contém dois clusters.

os avanços e análise dos dados. Para isso, podemos, por exemplo, alterar o arranjo bi-dimensional. Portanto, conforme os resultados forem sendo obtidos ele possa abranger mais eficientemente os dados a cada execução do algoritmo. Isso facilita a interpretação do mapa produzido e faz com que a topologia represente melhor os dados de entrada de forma que os resultados obtidos sejam mais eficientes.

A fim de garantirmos que os resultados a serem encontrados serão relevantes e não apenas uma consequência de nossa escolha de programas, é essencial que utilizemos uma amostra representativa. Para efetuar tal escolha de maneira justa, utilizaremos os pacotes de benchmarks SPEC CPU 2006 e SPEC CPU 2017, rotineiramente utilizados como referência nesta área de pesquisa. Analisaremos, então, o impacto que estas abordagens alternativas de aprendizado de máquina não supervisionado possuem para a escolha dos SimPoints e na qualidade dos SimPoints selecionados.

Referências

- Carlson, T. E., Heirmant, W., and Eeckhout, L. (2011). Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation. In *2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pages 1–12.
- Hamerly, G., Perelman, E., Lau, J., and Calder, B. (2005). Simpoint 3.0: Faster and more flexible program phase analysis. *Journal of Instruction Level Parallelism*, 7(4):1–28.
- Kohonen, T. (1998). The self-organizing map. *Neurocomputing*, 21(1-3):1–6.
- Sanchez, D. and Kozyrakis, C. (2013). Zsim: Fast and accurate microarchitectural simulation of thousand-core systems. In *ACM SIGARCH Computer architecture news*, volume 41, pages 475–486. ACM.
- Sherwood, T., Perelman, E., Hamerly, G., Sair, S., and Calder, B. (2003). Discovering and exploiting program phases. *IEEE Micro*, 23(6):84–93.
- Wunderlich, R. E., Wenisch, T. F., Falsafi, B., and Hoe, J. C. (2003). Smarts: accelerating microarchitecture simulation via rigorous statistical sampling. In *30th Annual International Symposium on Computer Architecture, 2003. Proceedings.*, pages 84–95.
- Zuchini, M. H. (2003). Aplicações de mapas auto-organizáveis em mineração de dados e recuperação de informação. Dissertação de mestrado, Univ. Estadual de Campinas.