

# Um estudo do ganho de performance de um sistema multi-núcleo heterogêneo sobre um sistema homogêneo



Jeckson Dellagostin Souza

Universidade Federal do Rio Grande do Sul

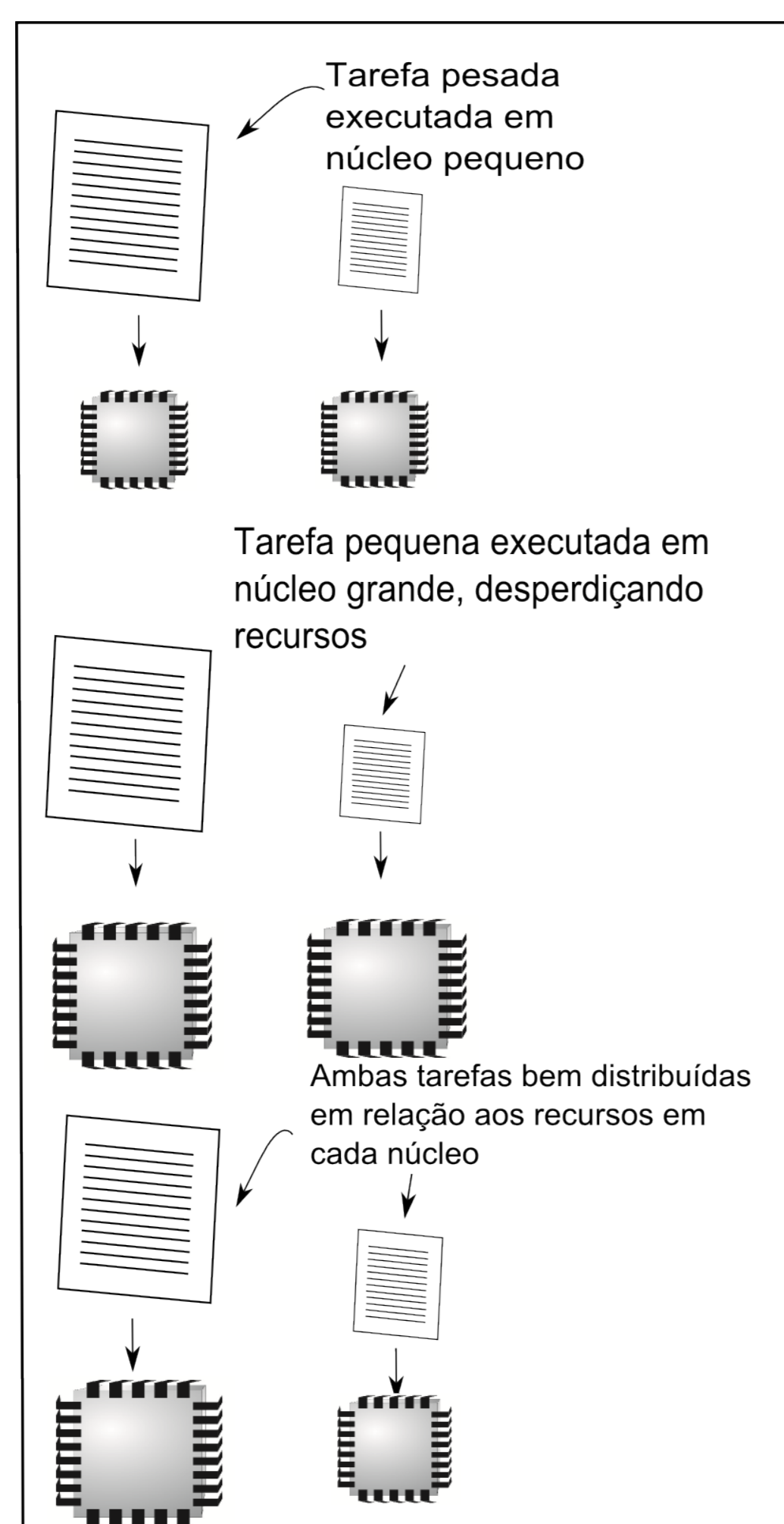
Instituto de Informática

Orientado por Prof. Dr. Antonio Carlos S. Beck Filho

## MOTIVAÇÃO

Historicamente, os processadores embarcados eram usados para executar operações **específicas e homogêneas** (sempre iguais). Por exemplo, um MP3 player era projetado apenas para decodificar sinais digitais de arquivos MP3 em sinais analógicos de áudio. Contudo, atualmente, os processadores embarcados em celulares, smart TVs e outros dispositivos eletrônicos, são capazes de executar aplicações diversas – **heterogêneas** – como o acesso a redes sociais, internet, execução de jogos, etc.

O CReAMS (Custom Reconfigurable Arrays for Multiprocessor Systems) é uma organização de **arquitetura reconfigurável e com múltiplos núcleos**. Isso quer dizer que ele é um processador que conta com vários núcleos e cada um desses núcleos é capaz de se **adaptar** para executar aplicações de forma mais rápida. No caso do CReAMS, a reconfiguração é obtida através de um **banco de unidades funcionais**. Essas unidades do banco são ativadas/desativadas conforme a necessidade da aplicação, a qual é dividida em várias partes e distribuída dentro do banco de forma a ser executada simultaneamente – paralelamente. Contudo, a eficiência de um processador CReAMS irá depender do tamanho desse banco e do quanto a aplicação em questão pode ser paralelizada. Além disso, como as aplicações são heterogêneas, ter um conjunto de núcleos que sejam iguais é ineficiente, pois aplicações mais “pesadas” – que têm mais capacidade de paralelização – levarão **mais tempo** do que seria necessário para executar em um núcleo com um banco pequeno, ao mesmo tempo que aplicações mais “leves” não ocupariam todos os recursos de um núcleo com um banco grande, de forma que esses recursos seriam **desperdiçados**.



Exemplo de distribuição de tarefas quando os núcleos são homogêneos (duas primeiras figuras) e heterogêneo (terceira figura).

## SOLUÇÃO PROPOSTA

Estender a organização de CReAMS criando versões deste com núcleos heterogêneos, ou seja, núcleos que possuam bancos de unidades funcionais com tamanhos diferentes.

## DESAFIOS

Queremos encontrar os tamanhos ideais para cada núcleo do processador de forma que as aplicações em geral tenham um **ganho na configuração heterogênea em relação as configurações homogêneas**. Contudo, as possibilidades de variar os núcleos são inúmeras e, portanto, tivemos que adotar uma metodologia para criar as configurações heterogêneas.

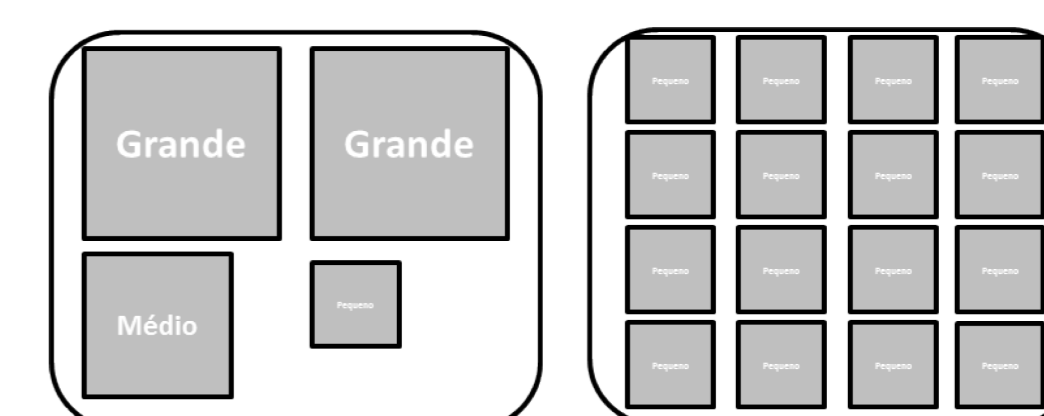
## METODOLOGIA

Para comparar as duas versões de CReAMS de forma justa, focamos a criação das configurações de forma que houve **paridade de área** entre elas, ou seja, ambas configurações heterogênea e homogênea ocupam o mesmo espaço físico dentro de um chip. Além disso, procuramos criar os bancos de unidades funcionais de acordo com as necessidades gerais da maioria das aplicações. Por exemplo, em geral as aplicações executam mais operações de soma e de lógica do que multiplicações, logo nossos bancos possuem mais somadores do que multiplicadores.

Os resultados obtidos foram extraídos via um **simulador de CReAMS**. Os scripts de simulação do CReAMS homogêneo foram adaptados de forma a poder executar o CReAMS heterogêneo. Adicionalmente, foram criados scripts para extração dos resultados, pois como muitas simulações foram feitas – para diversas aplicações e diversos números de núcleos –, o volume de arquivos com resultados tornou-se muito grande para extração manual.

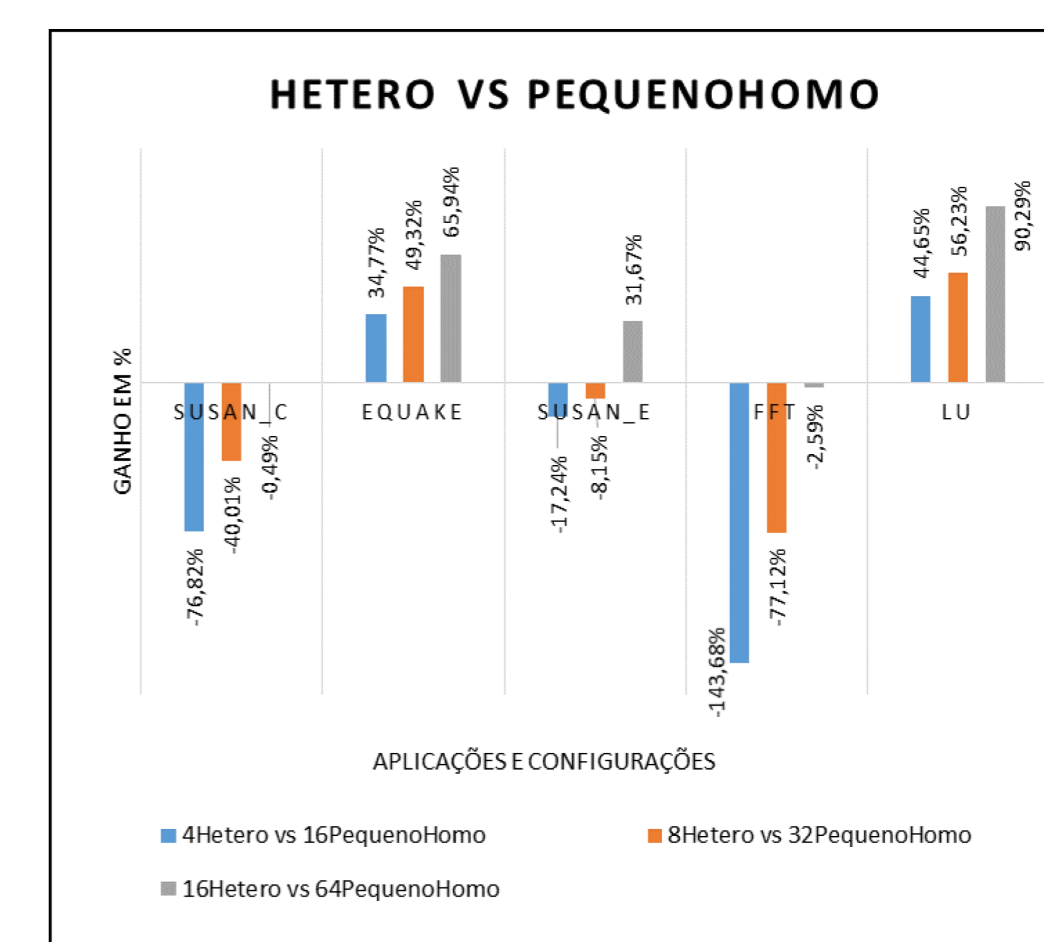
## RESULTADOS

Criamos uma configuração heterogênea para **ilustrar o potencial desta organização**. Nela, 50% dos núcleos são grandes, 25% são médios e os outros 25% são pequenos. Por exemplo, se forem usados 4 núcleos, 2 deles serão grandes, 1 médio e 1 pequeno. Comparamos ela com outras duas configurações homogêneas, uma delas com núcleos grandes (**GrandeHomo**) e outra com núcleos pequenos (**PequenoHomo**). No caso da configuração de núcleos grandes, essa é **duas vezes menor do que a heterogênea**, ou seja, iremos comparar a configuração heterogênea com uma homogênea de duas vezes mais núcleos (4Hetero com 8GrandeHomo, 8Hetero com 16GrandeHomo, etc) para obter a paridade de área. Já a configuração pequena é **oito vezes menor**.

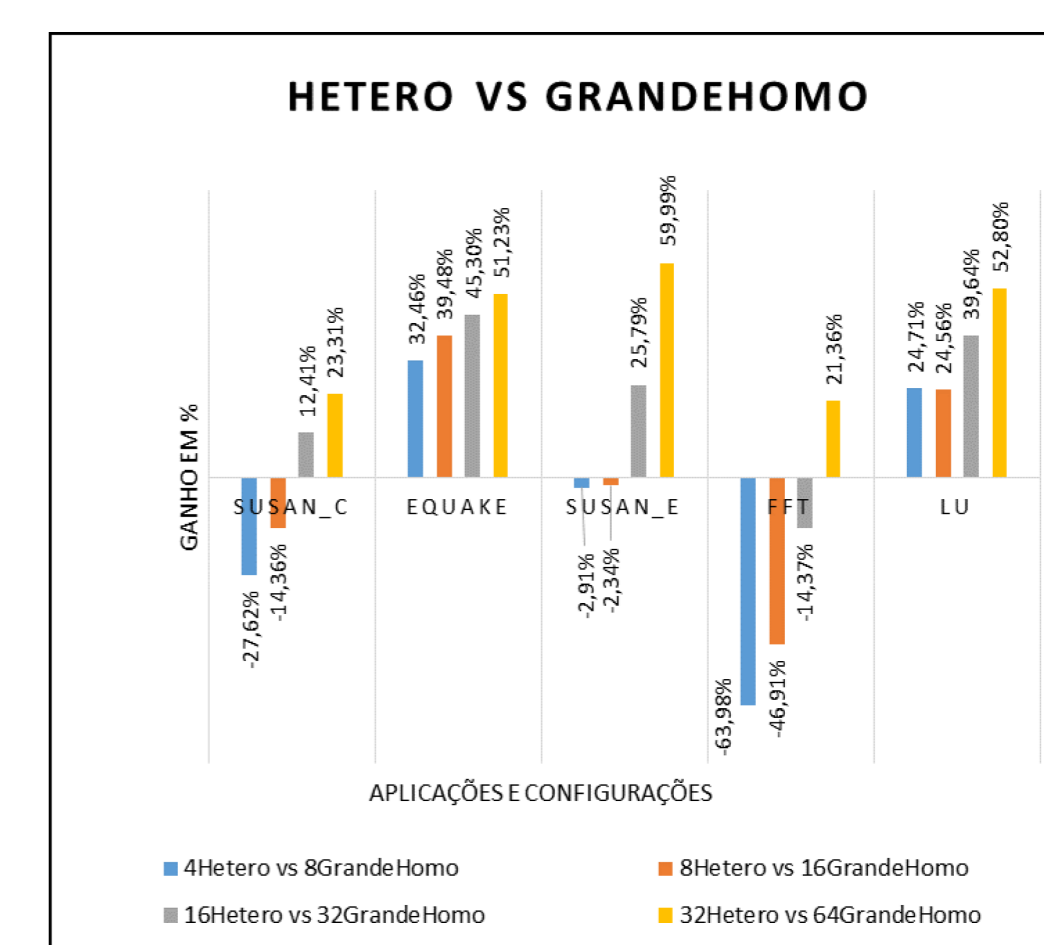


Exemplo da configuração heterogênea de 4 cores comparada com uma homogênea igual em área composta de 16 núcleos pequenos (PequenoHomo).

Os resultados mostram **aumentos de desempenho de até 90%** na comparação com a configuração homogênea pequena e de 60% na grande. Algumas aplicações, contudo, apresentam grandes perdas de desempenho (devido ao tipo de paralelismo que elas exploram, essas aplicações se beneficiam mais do número extra de núcleos que a configuração homogênea oferece), mas observamos que **conforme o número de núcleos cresce, essa perda de desempenho diminui**. Isso acontece devido ao efeito negativo da **comunicação entre os núcleos**. Quanto mais núcleos a configuração tiver, maior será o custo para eles se comunicarem, e isso torna-se **outra vantagem da configuração heterogênea testada**, já que ela é menor do que ambas as homogêneas.



Ganho em % da configuração heterogênea contra a homogênea pequena.



Ganho em % da configuração heterogênea contra a homogênea grande.

## CONCLUSÕES E TRABALHO FUTURO

Mostramos neste trabalho que usar uma organização heterogênea **pode trazer benefícios de desempenho** para algumas aplicações. Além disso, observamos uma tendência que pode ser explorada: **resultados ainda mais positivos podem ser obtidos quando se aumenta ainda mais o número de núcleos da configuração**.

Em trabalhos futuros, diversas novas abordagens poderão ser exploradas. Por exemplo, obter dados de potência, para verificar o ganho de energia que a organização heterogênea pode obter, ou criar um escalonador de tarefas mais inteligente, que torne a alocação dos núcleos ainda mais eficiente.