

Avaliação do consumo de energia na execução do NAS *Parallel Benchmark (NPB)* em processadores ARM

Jorge Ximenes S. Junior, Alexandre Carissimi

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{jorge.silva,asc}@inf.ufrgs.br

Abstract. *Power consumption becomes a critical aspect in the definition of exascale HPC systems and many research efforts are made to address the problem of energy efficiency. A promising approach is the use of low power processors such as ARM Cortex to design HPC systems. There are many ARM models available in the market differing on performance and power consumption. This paper compares an ARM Cortex A7 with an ARM Cortex A15 using time-to-solution and energy-to-solution criterias when executing NAS Parallel Benchmark (NPB). These processors are used to build a cluster of 16 processing cores. The results showed that lower power does not always mean lower energy consumption, since execution time often also increases.*

Resumo. *O consumo de energia é um aspecto básico no projeto de sistemas HPC exascale e várias pesquisas buscam resolver o problema da eficiência energética. Uma estratégia é o uso de processadores de baixo consumo, como o ARM. No entanto, há vários modelos de ARM, com diferentes capacidades de processamento e de consumo de energia. Este artigo compara um ARM Cortex-A7 com um ARM Cortex-A15 usando como critério o tempo-até-solução e energia-até-solução na execução do benchmark paralelo NAS. Esses processadores foram empregados na concepção de dois clusters com 16 núcleos de processamento, cada. Os resultados mostram que nem sempre baixo consumo representa menor energia, já que os tempos de execução tendem a aumentar.*

1. Introdução

A eficiência energética e o consumo de energia tornaram-se uma das questões fundamentais no projeto de sistemas computacionais de alto desempenho (HPC – *High Performance Computing*). Cada vez mais, as aplicações científicas exigem um maior poder computacional, e há a previsão de chegar à escala de exaflops (*exascale*) até o final desta década. Se os sistemas *exascale* fossem construídos com a tecnologia atual, seu consumo seria da ordem de Giga Watts, o que equivaleria à produção de uma usina de energia nuclear de tamanho médio [Wehner 2009]. Por isso, já em 2008, especialistas alertavam que o consumo de energia aceitável para o *exascale* seria de 20 MW [Bergman 2008], o que limita a eficiência energética desses sistemas a 50 Gflops/W.

Existem diferentes propostas para alcançar o desempenho *exascale* respeitando as restrições de energia. Uma delas é a computação heterogênea, ou seja, o uso conjunto de processadores de arquitetura convencional (x86, sparc...) e não convencional, como os processadores gráficos. Outra proposta é o uso de processadores de baixo consumo como, por exemplo, ARM. O projeto Mont-Blanc [Project.eu 2015] é um dos primeiros a introduzir a ideia de um supercomputador baseado em ARM. As apostas do projeto

são em um SoC (*System-on-Chip*) heterogêneo que combina um processador ARM com um processador gráfico para atingir alta capacidade de processamento com baixo consumo de energia. Outro ponto favorável aos sistemas baseados em ARM é o seu menor custo, se comparado aos processadores baseados em arquitetura x86.

Em decorrência do crescente interesse nos processadores ARM, impulsionado, em parte, pelo seu emprego em *smartphones* e dispositivos móveis, a indústria de semicondutores tem investido em novos modelos de ARM que variam entre si em relação à quantidade de núcleos (*cores*), tamanho de cache, frequência de operação, etc. A contribuição deste trabalho é a comparação, sob o ponto de vista do consumo de energia, de dois modelos de ARM (Cortex A7 e Cortex A15) presentes nos kits de desenvolvimento *Cubietruck* e *Jetson*. O objetivo é avaliar o impacto no desempenho e no consumo energético, de cada uma dessas placas com vistas a construir um *cluster* de processadores ARM de baixo custo e baixo consumo.

Este artigo está organizado em 7 seções, com esta introdução. Na Seção 2, trabalhos relacionados, são vistos estudos que comparam a eficiência energética dos processadores ARM com processadores tradicionalmente usados em sistemas HPC. A Seção 3 aborda conceitos básicos de potência e energia, assim como formas de medir o consumo de energia. A Seção 4 apresenta as duas plataformas experimentais usadas neste estudo. Já a Seção 5 discute os *benchmarks*, os cenários de teste e a metodologia empregados na avaliação experimental. Os resultados obtidos, e a comparação entre o *cluster* de *Cubietruck* e de *Jetson*, são apresentados na Seção 6. Por fim, a Seção 7 resume as conclusões, a contribuição e perspectivas de trabalhos futuros.

2. Trabalhos relacionados

Como visto anteriormente, a eficiência energética e o consumo de energia é uma preocupação em HPC, pois é uma barreira para a obtenção de sistemas *exascale*. Vários estudos foram feitos para analisar o uso de ARM em substituição aos processadores convencionais (x86, Sparc e PowerPC) em sistemas de HPC. Tipicamente, essas análises usam métricas como custo de aquisição, eficiência energética e desempenho.

Um primeiro estudo que se tem conhecimento é aquele de [Roberts-Hoffman 2009], onde é feita uma comparação entre um ARM Cortex A8 e um processador Intel Atom N330 (de baixo consumo) usando quatro *benchmarks* diferentes. Os resultados obtidos mostraram que o processador Atom tem um desempenho superior ao ARM, mas perde em relação à eficiência energética, medida em MFlops/Watt.

Em [Arouca 2012] é feita uma comparação de consumo entre vários dispositivos que possuem processadores ARM e x86. Os testes são baseados no desempenho obtido na realização de tarefas como servidor web, servidor de banco de dados e aplicações de ponto flutuante. As conclusões são que os SoCs baseados em ARM, se analisados pelo critério desempenho por Watt, fornecem um bom resultado na construção de servidores, apresentando um ganho de 3 a 4 vezes em relação aos processadores x86, sob diferentes padrões de carga (requisições HTTP e SQL). No entanto, ao comparar a capacidade de processamento em ponto flutuante, a arquitetura x86 leva vantagem.

Em [Ou 2012] é feita uma comparação entre um *cluster* baseado em ARM e um *cluster* baseado em Intel x86 usando a eficiência energética e o custo como critérios. A avaliação foi feita usando um servidor web, um sistema de banco de dados, mantido em memória, e uma aplicação de codificação de vídeo. Os resultados mostram ganhos de

eficiência energética do ARM de, aproximadamente, 1,3 para servidor web, 1,21 para codificação de vídeo e entre 2,6 e 9,5 para o banco de dados. Em relação ao custo financeiro, a conclusão foi que o *cluster* de ARM é viável para aplicações que não exigem poder computacional intensivo (e.g. banco de dados e web). Porém, essa vantagem diminui à medida que mais poder computacional é exigido, pois são necessários mais processadores ARM para prover a mesma capacidade que o Intel.

Padoin et alli, [Padoin 2012] investigam a possibilidade de usar processadores ARM como blocos de construção para arquiteturas HPC comparando seu desempenho em relação ao Intel Xeon. Nesse artigo são introduzidas as métricas “tempo até a solução” (*time-to-solution*) e “energia até a solução” (*energy-to-solution*). Os resultados indicam que, apesar do ARM consumir menos energia (até 90%) que um Xeon, o tempo de execução, nos diferentes testes realizados, é de 9 a 900 vezes mais lento. Assim, como a energia é proporcional ao tempo, à conclusão foi que ainda é questionável o uso de ARM para alcançar *exascale* em aplicações com alta carga de processamento.

Göddecke et alli [Göddecke 2013] analisam o desempenho de um *cluster* de 96 nós baseados em processadores ARM (Cortex A9, *dual core*) para a resolução de três classes diferentes de métodos numéricos para a solução de equações diferenciais parciais. Os resultados obtidos indicam que o cluster de ARM se mostrou viável em termos de *energy-to-solution*, se comparado com uma arquitetura Intel.

O estudo apresentado neste artigo se diferencia dos anteriores por assumir que há uma tendência do uso de SoCs baseados em processadores ARM em sistemas HPC. Sabe-se que, em geral, os processadores ARM foram projetados para apresentar baixo consumo em detrimento de desempenho. Assim, o foco não é comparar o ARM com processadores convencionais, mas, dado a diversidade de arquiteturas ARM, compará-los entre si. O objetivo do trabalho é responder a questão: “Quão importante e qual o impacto de se optar por uma determinada arquitetura ARM em relação à outra?”.

3. Energia, potência e medição de consumo de energia

A energia é a grandeza física que representa a capacidade de um sistema realizar trabalho e é medida em Joules (J), ou em quilowatt-hora ($1\text{kWh}=3600\text{ kJ}$). Já a potência é a taxa com o que a energia é transferida durante a realização de um trabalho e é fornecida em Watt ($W = \text{Joules por segundo}$). A relação entre potência e energia é dada por $P = dE/dt$, sendo P a potência, E a energia e t o tempo.

Para medir o consumo de energia de equipamentos e instalações elétricas, se usa dispositivos como wattímetros, alicates de corrente alternada, multímetros e fontes controladas. No entanto, essas soluções medem a potência consumida pelo sistema inteiro. Como cada vez mais é importante considerar o consumo de energia, os fabricantes de sistemas computacionais e de processadores começaram a introduzir em seus projetos mecanismos para medir o consumo de componentes, e não do equipamento inteiro. Os dois recursos mais usados são: (i) *chips* dedicados para medir a tensão e a corrente de componentes – ou do sistema - de forma precisa e com uma alta taxa de atualização dos dados medidos; (ii) registradores especiais, internos aos processadores, em que o próprio processador atualiza, conforme executa operações, o seu consumo de energia.

Quando, por falta de mecanismos apropriados, não é possível medir o consumo de potência de um processador, o TDP (*Thermal Design Power*), informado pelo fabricante, é usado como métrica de consumo de energia [Huang 2008][Padoin 2012]. O

objetivo do TDP é indicar a quantidade máxima de energia dissipada por um processador em condições extremas, e serve para o dimensionamento de ventiladores e dissipadores. O uso do TDP como métrica de consumo é controversa por duas razões. Primeira, não há, entre os fabricantes, um método padrão para mensurar o TDP de seus produtos, o que leva aos valores informados serem alvo de grande discussão. Segunda, o TDP é o valor máximo de dissipação de energia e não o consumo de energia, em si. Por fim, há o ACP (*Average CPU Power*) que fornece o valor médio de potência gerada pelo processador em uso, porém, além de ser algo recente, só é encontrado em processadores AMD. O ACP permite que o próprio processador monitore quanto de energia está dissipando e, com base nisso, ajuste, ou não, sua frequência de operação.

4. Plataformas experimentais

Para analisar o consumo de energia e a viabilidade de construir um *cluster* baseado em ARM foram empregadas duas placas: *Cubietruck* e *Jetson*. Enquanto a escolha da *Cubietruck* se deu por uma análise de diversas placas disponíveis do mercado, de fácil aquisição e baixo custo, a *Jetson* foi escolhida por já haver 4 unidades disponíveis em nosso laboratório. Ambas as placas foram usadas na construção de *cluster* tipo *Beowulf*. A seguir, são descritas as principais características de cada uma dessas placas e de cada um dos *clusters* montados usando essas placas como nós de computação.

***Cubietruck*:** A placa *Cubietruck* é composta por um SoC (*System-on-Chip*), que possui o nome comercial *Allwinner A20*, uma capacidade de 2 GB de memória RAM DDR3@480 MHz, uma memória *flash* NAND de 8 GB para ser usada como armazenamento permanente de dados e suporte a diversas conexões externas, entre elas, rede *GigaEthernet* (controlador RTL 8211E). Uma característica interessante da *Cubietruck* é o fato dela possuir um *chip* dedicado (AXP209) para monitorar o consumo instantâneo de corrente e tensão da placa inteira. O processador presente no SoC é ARMv7 de 32 bits, também conhecido como ARM Cortex-A7, que conta com dois núcleos (*cores*) operando a uma frequência de 1 GHz. Há ainda, nesse SoC, uma GPU Mali 400 MP2 com um relógio de 350 MHz.

Em relação ao software, a *Cubietruck* vem com *Android* instalado e tem suporte as seguintes distribuições GNU/Linux: *Cubieez*, *Lubuntu*, *Fedora*, *Debian* e *Ubuntu*. O sistema operacional pode ser instalado na memória *flash*, em um cartão micro SD, ou ainda, em HD externo. Nos EUA, o preço unitário da *Cubietruck* é cerca de US\$ 90,00.

***Jetson*:** A placa *Jetson* possui um SoC, cujo nome comercial é *Tegra K1*, e é composto por um processador ARM e uma GPU. O processador é um ARMv7-A de 32 bits, também conhecido como ARM Cortex-A15, com quatro núcleos que operam a uma frequência de 2,32 GHz. A GPU é uma *Nvidia Kepler GK20a @825 MHz*, com 192 Cuda *cores*. A placa conta com 2 GB de memória RAM do tipo DDR3L, frequência de 933 Mhz, e utiliza como armazenamento interno uma memória *flash* e *MultiMediaCard* (MMC) de 16 GB. Assim como a *Cubietruck*, a *Jetson* possui uma série de possibilidades de conectividade com o exterior, entre esses, suporte a *GigaEthernet* através do controlador RTL 8111GS. O preço de cada *Jetson* é de cerca de US\$ 192,00, nos EUA.

As principais distribuições de sistemas operacionais suportadas pela *Jetson* são: *Linux4Tegra*, *Gentoo Linux*, *Busybox* e *Android*. O sistema operacional pode ser instalado diretamente na memória de armazenamento, em cartão SD, em unidade de

disco externa, ou ainda, ser inicializado via interface de rede. A tabela 1 fornece uma comparação entre os processadores usados na *Cubietruck* e na *Jetson*.

Tabela 1. Comparativo entre os processadores ARM-V7 e ARM-V7a

	ARM-V7 (<i>Cubietruck</i>)	ARM-V7a (<i>Jetson</i>)
Processador/microarquitetura	ARM Cortex-A7 (32 bits)	ARM Cortex-A15 (32 bits)
Memória	DDR2/DDR3/DDR3L (@480 MHz)	DDR3L/LPDDR3 (@ 933 MHz)
Cache	L1: 32 KB (instruções)/32 KB(Dados) L2: 256 KB (compartilhada 2 cores)	L1: 32 KB(Instruções)/32 KB(Dados) L2: 2 MB (compartilhada 4 cores)
Largura de banda da memória	2 Gb/s	17 Gb/s
Frequência de operação	1 GHz	2.3 GHz
Quantidade de núcleos (cores)	2	4
Unidades de execução	1 unidade de execução, pipeline de 8 estágios, execução <i>in-order</i>	3 unidades de execução, pipeline de 15 estágios (ponto fixo) e até 17 (ponto flutuante), execução <i>out-of-order</i> .

Cluster ARM: A arquitetura *Beowulf* é composta por um nó servidor, que serve de *front-end* de acesso ao *cluster*, e um, ou mais, nós clientes. A conexão entre o nó servidor e os nós clientes usa uma rede isolada, implantada, tipicamente, com *switches* Ethernet. Um *cluster Beowulf* emprega apenas software livre.

Seguindo uma arquitetura *Beowulf*, foram construídos dois *clusters* de 16 processadores (*cores*). A definição dos *clusters* foi feita de maneira que eles tivessem a mesma quantidade de *cores* de processamento para facilitar a comparação entre eles. Dessa forma, o *CubieCluster* é formado por oito nós (placas *Cubietruck*), já que o processador de cada placa é *dual-core*. O *JetsonCluster*, por sua vez, é composto por quatro nós (placas *Jetson*), pois o processador em cada placa é *quad-core*. O sistema operacional usado nos nós é o GNU/Linux, sendo a *Cubietruck* usa uma distribuição *Debian Wheezy release 3.3*, núcleo versão 3.4.106, enquanto os nós *Jetson* empregam uma distribuição *Ubuntu 14.04* com núcleo versão 3.10.24.

O computador usado como nó servidor tem um processador Intel® Core™ i5-2357M de 1,40 GHz, *dual-core*, com suporte a HT (*Hyperthreading*), 4 GB de memória RAM DDR3 (@1333 Mhz) e placa de rede Gigabit Ethernet. O nó servidor, além de servir como *front-end* para acesso aos nós do *cluster*, via SSH, também funciona como servidor DHCP e de arquivos NFS. A distribuição Linux usada no servidor é Ubuntu 14.04, com *kernel* versão 3.13.0-35.

A rede local interna, composta pelo nó servidor e os nós clientes, é formada por um *switch* Ethernet, modelo 3Com 4250T, dedicado a essa tarefa. Esse *switch* possui 48 portas Ethernet de 10/100 Mbps, e uma capacidade de encaminhamento de 10,1 milhões de pacotes por segundo. A notar que tanto a placa *Cubietruck*, quanto a *Jetson*, tem capacidade de *Gigabit* Ethernet, a escolha desse *switch* foi, única e exclusivamente, por sua disponibilidade no laboratório.

5. Metodologia

5.1. Benchmark

A avaliação do consumo de energia foi feita através da execução do *NAS Parallel Benchmark* (NPB). A escolha do NPB se deve ao fato dele ser amplamente usado para avaliar sistemas de processamento paralelo. O NPB é composto por um conjunto de aplicações derivadas da dinâmica de fluidos divididos em cinco núcleos (*kernel*) e três pseudo-aplicações: método de gradiente conjugado (CG), aplicação embarçosamente paralela (EP), transformada rápida de Fourier (FFT), ordenamento de inteiros (IS), *multigrid* (MG), solução de sistemas de equações lineares (LU), solução de equações bloco tridiagonal (BT) e solução de equação pentadiagonal (SP).

A principal diferença entre um *kernel* e uma aplicação é que, enquanto o primeiro representa o cerne da computação de métodos numéricos, o segundo simula a movimentação de dados e a computação existente em dinâmica de fluidos. Cada um deles é caracterizado por um padrão de acesso a memória, de comunicação e processamento. Todos são implementados em OpenMP e em MPI e são divididos, de acordo com a demanda computacional, em classes (S, W, A, B, C, D, E e F).

O núcleo EP é uma aplicação CPU intensiva onde a computação usa apenas dados privativos para avaliar o desempenho em ponto flutuante. Já os núcleos CG e MG têm como principais características um padrão de acesso irregular aos dados (tarefas não trabalham sempre com o mesmo conjunto de páginas de memória), alto consumo de memória (grandes conjuntos de estruturas de dados) e grande quantidade de acessos a eles. O CG é um *kernel* que utiliza o método *Conjugate Gradient* para calcular o menor auto-valor em uma matriz de grande dimensão e não estruturada. Por sua vez, o *kernel* MG utiliza o método *V-cycle Multigrid* para calcular a solução 3D para a equação de Poisson. O *kernel* IS tem um padrão irregular de acesso aos dados e usa a unidade de ponto fixo. Por fim, o núcleo FFT que é uma solução parcial de equações diferenciais, e faz um teste de comunicação, pois as tarefas trocam dados entre si de forma intensiva.

Quanto as pseudo-aplicações, a LU apresenta uma heterogeneidade muito grande no seu padrão de acesso a dados, pois esses são feitos de forma regular, mas intercalados por acessos irregulares. Já as pseudo-aplicações BT e SP realizam acessos regulares a dados nas suas diferentes fases da execução paralela.

5.2. Cenários de teste

Os testes foram realizados com a execução do NPB, classe B, nas versões OpenMP e MPI, em quatro cenários diferentes. O primeiro cenário é a execução de todos os núcleos e pseudo-aplicações, na versão OpenMP, em uma única placa *Cubietruck*. Foram usadas 1, 2 e 4 *threads*. O segundo cenário emprega o NPB, na versão MPI, no *cluster Cubietruck*. A quantidade de processos MPI lançados no *cluster* variou entre 2, 4, 8 e 16. Nas execuções com menos de 16 processos, cada placa *Cubietruck* recebeu um processo, ou seja, apenas um de seus *cores* estava sendo usado para processamento.

Um procedimento similar foi feito para a placa *Jetson*. Assim, o terceiro cenário é composto pela execução do NPB, na versão OpenMP, em uma única placa *Jetson*, também usando 1, 2 e 4 *threads*. E, por fim, o quarto cenário, que é a execução do NPB, versão MPI, no *cluster Jetson*. Novamente, a quantidade de processos MPI variou entre

2, 4, 8 e 16. Nos casos de 8 e 16 processos MPI, mais de um processo MPI foi alocado para um mesmo nó (placa *Jetson*), 2 e 4 processos, respectivamente.

5.3. Critérios de avaliação

Os critérios usados na comparação de desempenho e consumo de energia entre a *Cubietruck* e a *Jetson* são aqueles dados em [Padoin 2012]: *time-to-solution* e *energy-to-solution*. Ou seja, foram avaliados o tempo total até o término de cada um testes do NPB e, em seguida, esse valor foi multiplicado pelo consumo, em Watts, de cada uma das placas (*energy-to-solution*).

Os valores de tempo de execução representam a média das execuções de cada *kernel* e pseudo-aplicação do NPB, com um intervalo de confiança superior a 95%. A maioria das execuções se mostrou extremamente estável, com um desvio padrão de cerca de 1%, sendo, portanto, adequado o uso da média.

5.4. Medidas do consumo de energia

Um dos problemas enfrentados é como medir o consumo de energia de cada uma das placas usadas. Como visto na seção 4, a *Cubietruck* tem um *chip* dedicado para medição de tensão e corrente instantâneas. Entretanto, essa facilidade não foi usada por dois motivos. Primeiro, seria necessário determinar uma frequência de amostragem para obter os valores instantâneos, integrá-los no tempo e obter o consumo de potência; mas o *daemon* responsável por essa amostragem interferiria na execução da aplicação quando os dois *cores* fossem usados no processamento. Segundo, a placa *Jetson* não conta com essa mesma facilidade, e, então, buscou-se medir o consumo de energia nas duas placas usando um mesmo procedimento.

Houve uma dificuldade adicional com a placa *Jetson*: a indisponibilidade de acesso ao *cluster Jetson* no momento das medidas. Então, a estratégia adotada foi a seguinte: assumir valores de consumo para a placa *Jetson* e para a placa *Cubietruck*. Assim, de acordo com os experimentos dados em [Emami 2015], quando apenas o processador é utilizado em sua plenitude, o consumo da *Jetson* é de 5 Watts. Para a *Cubietruck*, foram feitas medidas experimentais usando uma fonte controlada externa durante a execução do núcleo EP do *benchmark* NAS. Nesse caso, o consumo de energia medido foi: (i) 2,4 W, quando apenas um *core* foi utilizado; e, (ii) 3,25 W quando os dois *cores* foram usados. Esses valores são compatíveis com os obtidos em um experimento usando o *chip* dedicado AXP209, conduzido à parte, e dos valores apresentados em [Smith 2015]. Assim, nos resultados apresentados na Seção 6, assumiu-se, para todos os testes, de forma conservadora, que o consumo da *Jetson* é 5 W, e, para a *Cubietruck* considerou-se 2,4 W nos testes em que apenas um *core* é empregado e 3,25 W para os testes que usam os dois *cores* da mesma placa.

6. Resultados

O gráfico da Figura 1 mostra o tempo de execução (*time-to-solution*) para os testes realizados com o NPB, versão OpenMP, nas duas placas. O gráfico é útil para a visualização de tendências. A Tabela 2, por sua vez, fornece os valores do tempo de execução (*time-to-solution*) e consumo de energia (*energy-to-solution*), obtidos nos experimentos.

Inicialmente, no núcleo EP, é possível observar que a aceleração é linear com a adição de mais *threads* até o limite da quantidade de *cores* de cada uma das placas.

Ainda, a *Jetson* tem um desempenho 4 vezes superior ao da *Cubietruck*. Essa diferença é explicada pelo fato da frequência de operação ser duas vezes maior, e que o processador da *Jetson* (ARM-v7a) possui uma unidade de ponto flutuante melhor que o da *Cubietruck*. Por consequência, essa característica impacta no consumo energético já

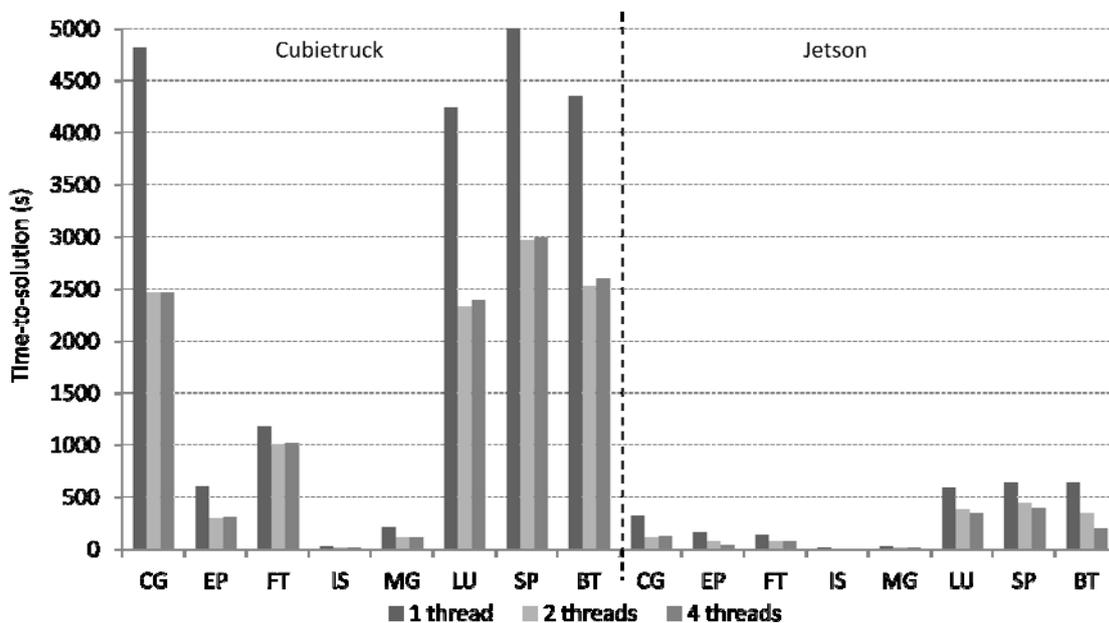


Figura 1. *Time-to-solution* para NPB (versão OpenMP)

Tabela 2. *Time-to-solution* e *energy-to-solution* para NPB (versão OpenMP)

Testes	Cenários	Cubietruck			Jetson		
		1 thread	2 threads	4 threads	1 thread	2 threads	4 threads
EP	Tempo (s)	602,47	304,55	306,39	159,45	80,43	40,78
	Energia (J)	1445,93	989,79	995,76	797,25	402,15	203,90
CG	Tempo (s)	4819,76	2465,97	2468,18	325,24	177,15	125,36
	Energia (J)	11567,43	8014,41	8021,60	1626,20	885,75	626,80
FT	Tempo (s)	1186,57	1005,29	1024,08	134,33	82,76	78,25
	Energia (J)	2847,78	3267,21	3328,26	671,65	413,80	391,25
IS	Tempo (s)	32,90	20,38	20,62	19,37	10,24	5,79
	Energia (J)	78,97	66,26	67,04	96,85	51,20	28,95
MG	Tempo (s)	210,73	115,99	117,26	26,39	18,24	15,42
	Energia (J)	505,75	376,98	381,11	131,95	91,20	77,10
LU	Tempo (s)	4245,30	2337,05	2388,78	597,91	391,06	342,72
	Energia (J)	10188,73	7595,42	7763,56	2989,55	1955,30	1713,60
SP	Tempo (s)	5326,94	2970,66	2989,84	636,60	448,69	400,50
	Energia (J)	12784,67	9654,67	9716,99	3183,00	2243,45	2002,50
BT	Tempo (s)	4358,80	2530,60	2603,54	647,02	346,57	204,05
	Energia (J)	10461,13	8222,72	8461,52	3235,10	1732,85	1020,25

que, apesar da *Jetson* consumir mais potência, ela executa em menos tempo. Cabe lembrar que esses resultados ainda favorecem a *Cubietruck*, pois enquanto seu consumo é de 2.4W, para um *core*, e 3.25 W para dois *cores* (valores medidos), para a *Jetson* assumiu-se sempre o pior caso, isso é, um consumo de 5W.

Na execução do *kernel* IS, se constata que a aceleração da *Cubietruck* é 1,6 para a adição de um *core* a mais, enquanto que na *Jetson* essa aceleração é de 1,89. Essa diferença decorre do fato que a unidade de ponto fixo da *Jetson* ser melhor que a do *Cubietruck*. Além disso, a *Jetson* possui uma largura de banda de acesso à memória bem melhor. Ainda, a *Jetson* é 1,7 vezes mais rápida que a *Cubietruck*. Assim, mesmo com um maior consumo de potência, a *Jetson* teve um menor consumo energético.

Os *kernels* CG e MG, e as pseudo-aplicações LU, SP e BT, são caracterizados por acessar grande quantidade de dados, com padrões irregulares e regulares. Nesses testes, se observa uma diferença em tempos de execução entre 7 a 14 vezes a favor da *Jetson*. Isso é explicado, principalmente, pela organização de memória da *Jetson* que, além de possuir uma cache maior, tem uma largura de banda de acesso superior ao da *Cubietruck*. De novo, em função do menor tempo de execução (*time-to-solution*), a *Jetson* obteve um melhor desempenho no consumo energético (*energy-to-solution*).

Por fim, o *kernel* FT, que é caracterizado por uma forte comunicação e sincronização entre as tarefas. Nesse caso, se identifica que a adição de novas *threads* e *cores* não trouxe um ganho proporcional (baixa escalabilidade). Além disso, as medidas pioraram para a *Cubietruck*. Atribui-se esses resultados ao comportamento da cache (falso compartilhamento). Mais uma vez, a *Jetson* teve um desempenho superior ao da *Cubietruck* (cerca de 9 vezes), o que se reflete no consumo energético.

Os resultados obtidos com a execução do NPB, versão MPI, nos dois *clusters* são dados na Figura 2 e na Tabela 3. Como no OpenMP, a Figura 2 ilustra, para efeitos de comparação visual, os tempos de execução obtidos em cada um dos *clusters*, e a Tabela 3 explicita os valores obtidos. Cabe lembrar que os processos MPI foram lançados um por placa, a exceção do caso de 16 processos no *CubieCluster* (2 processos MPI por placa) e dos casos de 8 e 16 processos no *Cluster Jetson* (2 e 4 processos MPI por placa, respectivamente). Assim, o *energy-to-solution* é calculado multiplicando o tempo de execução (*time-to-solution*) pelo número de placas e pelo consumo correspondente, 2,4W, 3,25W, ou 5W, conforme o caso.

O comportamento apresentado pelo MPI foi similar àquele do OpenMP para todos os *kernels* e pseudo-aplicações, ou seja, o *Cluster Jetson* sempre apresentou melhor consumo energético que o *Cluster Cubie*, já que a *Jetson* foi de 4 a 10 vezes mais rápida que a *Cubietruck*. Os testes com as aplicações BT e SP precisam executar com uma quantidade de processos em quadrado perfeito (1, 4 e 16), o que explica a ausência das medidas para os casos de 2 e 8 processos na Tabela 3.

A adição de novos nós de cálculo faz com que a potência consumida aumente proporcionalmente. O desejável é que esse aumento seja compensado com a diminuição do consumo de energia, pois, ao se dobrar a quantidade de processadores, se espera ter uma execução pela metade do tempo. Mas, isso só é verdade para aplicações que tem ótima escalabilidade, como o *kernel* EP. Nos demais *kernels* e nas pseudo-aplicações, por haver necessidades de sincronização, ou de comunicação, a adição de novos nós aumentou o consumo de potência e de energia. Porém, sempre que se aumentou a quantidade de *cores* usados nas placas (casos de 16 processos na *Cubietruck* e de 8, ou

16, processos na *Jetson*), se teve uma redução no consumo de energia. Isso porque a adição de um *core* a mais, na mesma placa, reduz o tempo de execução, mas não onera, na mesma proporção, o consumo de energia.

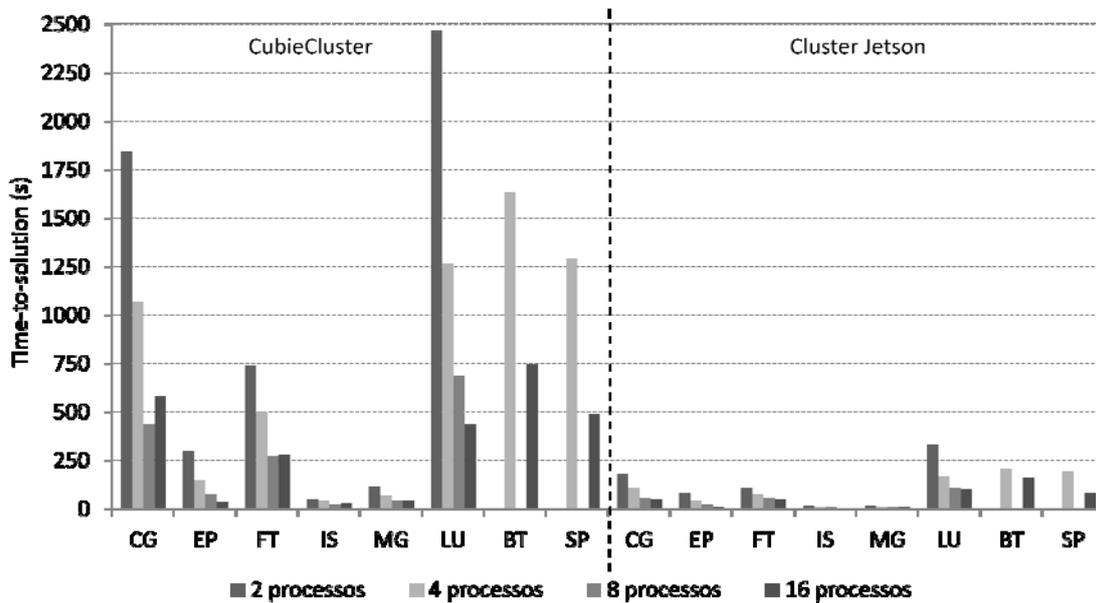


Figura 2. *Time-to-solution* para NPB (versão MPI)

Tabela 3. *Time-to-solution* e *energy-to-solution* para NPB, para 2, 4, 8 e 16 processos MPI

Testes	Cenários	Cluster Cubietruck				Cluster Jetson			
		2	4	8	16	2	4	8	16
EP	Tempo (s)	299,40	150,07	75,36	38,04	82,87	41,25	20,92	11,11
	Energia (J)	1437,13	1440,71	1447,01	989,27	822,70	825,00	418,40	222,20
CG	Tempo (s)	1843,11	1066,11	439,27	578,99	182,07	110,44	53,39	51,10
	Energia (J)	8846,96	10235,25	8434,12	15053,83	1820,70	2208,80	1067,80	1022,00
FT	Tempo (s)	735,99	499,89	269,57	276,22	110,22	74,36	53,85	47,80
	Energia (J)	3532,76	4798,99	5175,88	7181,74	1102,20	1487,20	1077,00	956,00
IS	Tempo (s)	48,62	40,42	22,78	28,63	13,99	8,20	6,65	5,87
	Energia (J)	233,37	388,11	437,50	744,43	139,90	164,00	133,00	117,40
MG	Tempo (s)	116,19	67,98	44,76	43,73	14,65	8,85	6,55	6,61
	Energia (J)	557,72	652,64	859,41	1137,04	146,50	177,00	131,00	132,20
LU	Tempo (s)	2468,46	1266,04	689,02	439,03	332,83	167,96	108,26	103,42
	Energia (J)	11848,63	12154,05	13229,33	11414,90	3328,30	3359,20	2165,20	2068,40
SP	Tempo (s)		1635,08		748,20		204,73		162,18
	Energia (J)		15696,82		19453,36		4094,6		324,60
BT	Tempo (s)		1290,07		489,39		197,02		84,67
	Energia (J)		1284,73		12724,29		1970,02		1893,20

Em relação ao MPI, um ponto interessante a ressaltar é que tanto a *Cubietruck* quanto a *Jetson* apresentaram uma boa escalabilidade com a adição de nós de cálculo. Esse resultado complementa o estudo comparativo entre MPSoCs com ARM, feito por [Padoin 2013], que verificou problemas de escalabilidade usando placas do modelo *Pandaboard* durante a execução do benchmark *Linpack* (HPL). Essa diferença decorre do fato que a *Cubietruck* e a *Jetson* possuem *chipset* específico para suporte à rede Ethernet, enquanto que na *Pandaboard* a capacidade de rede é via barramento USB, o que limita a sua banda passante. Por fim, em [Ximendes 2013] são fornecidos resultados complementares, com outras configurações de testes entre a *Cubietruck* e a *Jetson*.

7. Conclusão

Este artigo compara dois *clusters* de 16 *cores* de processamento, onde os nós são placas que empregam SoCs baseados em ARM: *Cubietruck* (Cortex A7) e *Jetson* (Cortex A15). O objetivo é avaliar o impacto no desempenho e no consumo energético, de cada uma dessas placas, com vistas a construir um *cluster* de processadores ARM. Os critérios usados na comparação foram *time-to-solution* e *energy-to-solution*.

A contribuição deste trabalho foi mostrar que o tipo de processador ARM a ser usado na construção de sistemas HPC é importante: nem sempre o processador com o menor consumo de potência, oferece o menor consumo de energia.

Nos testes realizados, os tempos de execução (*time-to-solution*) da *Jetson* foram sempre melhores que os da *Cubietruck*. Essa diferença decorre da arquitetura do processador da *Jetson* ter várias melhorias em relação ao da *Cubietruck*. Porém, em nosso contexto, o importante não é o desempenho em si, mas o consumo energético, e as melhorias da *Jetson* representam um maior consumo de potência. No entanto, os resultados mostraram que a menor potência exigida pela *Cubietruck* não se traduziu em menor consumo de energia, pois os tempos de execução aumentaram. Assim, a placa *Jetson* sempre forneceu melhores resultados nos critérios *time-to-solution* e *energy-to-solution*.

Em relação ao custo financeiro, o *cluster* de 16 *cores* de processamento, considerando apenas os nós de processamento, é de cerca de US\$ 768,00 (4 placas x US\$ 192,00), com a *Jetson*, contra US\$ 720,00 (8 placas x US\$ 90,00) com a *Cubietruck*. Ou seja, o *cluster* de *Jetson* possui quase o mesmo custo que o de *Cubietruck*, mas oferece um melhor consumo energético. Portanto, os nós baseados em *Jetson*, apesar do maior consumo de potência, são, entre as duas opções analisadas, os mais indicados para construção de um *cluster* ARM.

Como trabalhos futuros pretende-se investigar outros modelos de ARM e executar, além do NPB, o benchmark *Linpack* (HPL). A motivação para uso do HPL é devido ao fato dele ser usado para mensurar a eficiência energética para a elaboração do ranking *Top Green 500*. Outro ponto a considerar é, em medidas futuras, fornecer a relação MFlop/Watt. A influência no desempenho e no consumo de energia ao realizar o ajuste dinâmico de voltagem e frequência baseado na carga necessária de processamento, mecanismo conhecido como DVFS (*Dynamic Voltage and Frequency Scaling*), é outro fator que pode ser explorado.

Por fim, há a questão de medir o consumo de energia do sistema inteiro, pois, seria interessante avaliar o consumo dos dispositivos de E/S, em especial, da interface de rede. Neste trabalho, pela particularidade da placa *Cubietruck* possuir um *chip* específico (AXP209), que fornece o consumo instantâneo de tensão e corrente da placa

inteira, foram realizados experimentos a parte comparando o consumo do uso intensivo do processador (núcleo EP do NPB) com os núcleos NPB que usam troca de mensagens MPI. Nesses testes, não se detectou aumentos significativos de consumo de energia ao se usar a rede, porém, não é possível afirmar se isso é devido a problemas de precisão e amostragem do AXP209, do padrão de comunicação (baixo) ou que, se em comparação ao processador, o consumo de energia do *chipset* Ethernet não contribui de forma significativa para o consumo total da placa. Ainda, para o dimensionamento físico de capacidade elétrica (fios, disjuntores, etc) é importante determinar o valor de pico de consumo de energia, já que a métrica de desempenho por Watt pode mascarar esse valor.

Referências Bibliográficas

- Aroca, R.V.; Gonçalves, L.M.G. (2012). “Towards green data centers: A comparison of x86 and ARM architectures power efficiency”. *Journal of Parallel Distributed Computing*, 72 (2012), pag. 1170-1780.
- Bergman, K.; et alli. (2008). “Exascale computing study: Technology challenges in achieving exascale systems”. DARPA, Tech. Rep.
- Emami, S.; (2015). Jetson/Computer vision performance. Acessado em junho 2015. (Disponível em: http://elinux.org/Jetson/Computer_Vision_Performance).
- Göddecke, D.; et alli. (2013). “Energy efficiency vs. performance of the numerical solution of PDEs: An application study on a low-power ARM-based cluster”. Elsevier. *Journal of Computational Physics*. Vol. 237, 15 March 2013, pag. 132–15.
- Huang, W. et alli. (2008). “Many-core design from a thermal perspective”. In.: ACM. *Proceeding of the 45th annual Design Automation Conference*. Pag. 746-749.
- Ou, Z.; Pang, B.; Deng, Y.; Nurminen, J.; Yla-Jaaski, A.; Hui, P. (2012). “Energy and Cost efficiency analysis on ARM based Cluster”. *Proceedings of the 2012 12th IEEE/ACM Int. Symposium on Cluster, Cloud and Grid Computing*, pag. 115-123.
- Padoin, E.L.; Oliveira, D.; Velho, P.; Navaux, P. (2012). “Time to Solution and Energy to Solution: A comparison between ARM and Xeon”. In.: IEEE. *3rd Workshop on Applications for Multi-Core Architecture (WANCA 2012)*. pg. 48-53.
- Padoin, E.L et alli. (2013) “Análise de desempenho, escalabilidade e eficiência energética de MPSoCs com processadores ARM”. In.: *Conferencia Latino Americana de Computación de Alto Rendimiento (CLCAR)*, San José, Costa Rica, 2013.
- Project.Eu Montblanc (2015). “Mont-Blanc Home Page”. Acessado em Junho 2015. (Disponível em: <http://www.montblanc-project.eu>)
- Roberts-Hoffman, K.; Hegde, P. (2009), “ARM Cortex-A8 vs Intel Atom: Architectural and benchmark comparisons”. Dallas: University of Texas at Dallas.
- Smith, J.; Hamilton, A.; (2015). “Massive Affordable computing using ARM processors in high energy physics”, In: IOP Publishing. *Journal of Physics*. V.608, n.1, p 12001.
- Ximendes, J.S.; (2015) “Avaliação do consumo energético do *benchmark* NAS em processadores ARM”. Trabalho de Conclusão de Curso. Engenharia de Computação. Instituto de Informática. UFRGS. 67 pág.
- Wehner, M.; Oliner, L.; Shalf, J. A real cloud computer. *Spectrum, IEEE*, IEEE, v. 46, n. 10, p. 24–29, 2009.