Vol.2014-ARC-210 No.1
Vol.2014-OS-129 No.1
2014/5/14

Web サービスのリクエストに着目したサーバ計算機の消費電力特性

山田 浩史[1,a)]　河合 英宏[2,b)]　大島 訓[2,c)]

概要：データセンタにおけるサーバ計算機の消費電力は膨大であり，データセンタを構築する上で大きな障壁となっている．サーバ計算機の消費電力を抑えるために，近年低処理能力 CPU を搭載した省電力サーバ計算機が登場している．Intel Atom プロセッサや ARM プロセッサを搭載してサーバ計算機自身の消費電力を抑え，データセンタの省電力化に寄与することが期待されている．本研究の目的は，データセンタにおける典型的なワークロードに対する省電力サーバ計算機の電力特性を定量的に示すことにある．本研究では，HTTP サーバ，Key-Value ストア (KVS) に焦点を絞り，省電力サーバ計算機の消費電力当たりのスループットを計測する．本実験を通じて，リクエストサイズに応じてサーバマシンを選定することで消費電力を抑えられる，受信するリクエストサイズに応じたサーバ計算機を活用できるよう異なる命令セットをもつ CPU 間で仮想マシン移送を実現できる手法が必要などの知見を得ることができた．

1. Introduction

データセンタにおけるサーバ計算機の消費電力は膨大であり，データセンタを構築する上で大きな障壁となっている．電力の供給量は限られているため，データセンタの処理能力を向上させためにサーバ計算機の単純に追加することは難しく，供給電力によってデータセンタの処理能力が制限されてしまう．また，電力コストも甚大であり，データセンタを有するベンダにとっては深刻な問題となっている．たとえば，1.3 MW 規模のデータセンタの年間電力コストは 1 億 2 千万円にものぼる [1]．また，Google の年間電気代は 38000 万ドルにも及ぶ [2]．

サーバ計算機の消費電力を抑えるために，近年低処理能力 CPU を搭載した省電力サーバ計算機が登場している．Intel Atom プロセッサや ARM プロセッサを搭載してサーバ計算機自身の消費電力を抑え，データセンタの省電力化に寄与することが期待されている．省電力サーバは，通常のサーバ計算機よりはピーク性能は劣るものの，電力当たりのスループットは高い．そのため，限られた供給電力量のなかで，これらをうまく組み合わせることでデータセンタの処理能力の向上が望める．実際に，研究分野では高性能なサーバ計算機と省電力サーバ計算機とを混在させたデータセンタの構築方法が提案されつつある [3], [4]．

本研究の目的は，データセンタにおける典型的なワークロードに対する省電力サーバ計算機の電力特性を定量的に示すことにある．本研究では，HTTP サーバ，Key-Value ストア (KVS) に焦点を絞り，省電力サーバ計算機の消費電力当たりのスループットを計測する．実際に Xeon プロセッサが搭載したハイエンドサーバ計算機 (Xeon)，Intel Atom プロセッサが搭載したサーバ計算機 (Atom)，ARM プロセッサが搭載した計算機 (Arm) を用意して，その上で Apache と Memcached を稼働させ，リクエストするデータサイズを変動させることで，そのときの消費電力当たりのスループットを計測する．

本実験を通じて以下のことがわかった．

- リクエストするデータサイズに応じてサーバ計算機を選択することで，消費電力当たりのスループットを向上できる．データが小さいものは CPU バウンドとなりやすいため，Xeon といった高性能なサーバ計算機に任せた方がよく，データが大きいものはネットワークバウンドとなりやすいため，Arm や Atom といった省電力サーバ計算機に任せた方がよい．
- KVS はネットワークバウンドになりやすいため，Arm や Atom といったサーバマシンを利用することで大幅な消費電力削減が見込める．
- Arm の消費電力効率は Web サーバや KVS いずれの場合も顕著である．そのため，Arm を複数台用意し

[1] 東京農工大学
2-24-16 Nakacho Koganei-shi, 184-8588, Japan
[2] 日立製作所横浜研究所
292 Yoshidacho, Totsuka-ku, Yokohama, 224-0817, Japan
[a)] hiroshiy@cc.tuat.ac.jp
[b)] hidehiro.kawai.ez@hitachi.com
[c)] satoshi.oshima.fk@hitachi.com

て Xeon よりも消費電力当たりのスループットを高められるかを検証する必要がある．KVS はもともとスケールアウトするアプリケーションなので，Arm を複数台並べる方式は有効に働く可能性が高い．

- 仮想マシン技術を考慮することでより高い消費電力を見込める．仮想マシン技術を用いて，複数サーバを同一マシンに集約することができる．この際に，Xeon 1 台で複数の仮想マシンを稼働させた方がよいのか，Arm を複数台用意してサービスを稼働させた方がよいのかは今後検証すべき課題である．その際，本研究で作成したベンチマークを利用することで，リクエストサイズに応じたサーバの配置方法が明確になる．
- リクエストの種類の変動に応じて，適切なサーバマシンへとサービスを移送する技術が求められる．たとえば，リクエストされるデータサイズが小さいときには Xeon に配置し，サイズが大きくなったら Arm に配置することが可能な技術が消費電力を削減する上では重要となる．仮想化技術が用いられているデータセンタでは，Arm から Xeon，またその逆は CPU アーキテクチャが異なるマシン間での移送になるため，現在の技術では移送は不可能である．そのため，CPU アーキテクチャが異なるマシン間の移送を実現する技術が必要となる．

本論文の構成は以下のとおりである．2 章で関連研究について述べる．3 章にて本実験の具体的な方法や環境について記す．4，5 章では，実験結果について述べる．6 章にて本実験で得たデータを考察し，7 章で本論文をまとめる．

2. Prior Work

本研究では Xeon プロセッサや Intel Atom プロセッサ，ARM プロセッサを搭載した計算機を用意し，その上で稼働する HTTP サーバおよび Key-Value ストア (KVS) に対して負荷かけることで定量的に消費電力特性を示す．これにより，データセンタの構成法や新たなシステムソフトウェアの構築の足がかりとする．

Tudor らは ARM プロセッサを搭載した計算機に着目し，サーバワークロードを実行した際の CPU サイクルとメモリサイクル，ネットワーク I/O の実行時間を計測し，その実測に基づくモデル化手法を提案している [5]．本モデルを用いて，消費電力やワークロードの実行時間を見積もることができる．Tudor らは ARM のモデル化に注目しており，他のサーバ計算機との比較は行っていない．

Verma らは実環境のワークロードトレースを利用して，その資源使用率を利用して静的にサーバを集約する方式を提案している [6]．本研究で得た知見，および Verma らの提案手法を組み合わせれば，消費電力当たりのスループットがより高いサーバ集約が可能となる．

FAWN [4] では，処理能力の低いプロセッサを搭載したサーバ計算機を活用した低消費電力なクラスタの構築手法を提案している．FAWN 上で仮定している低処理能力プロセッサはシングルコアである．近年の Atom や ARM のマルチコア化により，それを搭載したマシンのボトルネックとなる箇所がサーバワークロードにおいてシングルコアとは異なることが知られている [5]．本研究ではマルチコア化した低処理能力プロセッサを利用しており，FAWN で構築されたクラスタをより効率化することができると考えられる．

サーバ計算機の構造を工夫することで消費電力を削減する手法が提案されている．KnightShift [7] では，低処理能力サーバ計算機およびハイエンドなサーバ計算機を組み合わせて，リクエストに応じて使用する計算機を切り替えるアーキテクチャを提案している．他にも，高速に active/sleep 切り替えを可能にする方式 [8] やモバイルデバイスのメモリを利用する方式 [9] などが提案されている．これらを併用することでより消費電力の低いデータセンタを構築することができる．

CPU の周波数を動的に変更する DVFS を活用した場合における消費電力特性の評価やモデル化がなされている [10], [11], [12], [13]．本研究では，DVFS ではなく異なるサーバ計算機の消費電力特性を示すことが目的である．

3. Experimental Setup

本研究ではリクエストの変化に応じたサーバ計算機の消費電力特性を定量的に示すことを目的とする．本稿では第一歩として，データセンタ内で稼働している典型的なサービスである HTTP サーバおよび KVS に着目して，そのリクエストの変動に応じて消費電力がどのように変動するかを計測する．

3.1 Configurations

本実験で利用したマシンは次の通りである．これらのマシンはギガビットイーサネットのスイッチに繋ぎ，Apache Web サーバおよび Memcached を稼働させた．具体的には，それぞれ Linux 3.11.5，Apache 2.0.17，Memcached 1.4.10 が稼働している．

- **Xeon:** Intel Xeon E3-1240 v3 3.4 GHz
- **Atom:** Intel Atom Processor S1260，2.0 GHz，Dual-Core
- **Arm:** Octa Cortex A15 1.6 GHz Quad-Core & Cortex A7Quad-Core CPUs

またこれらの消費電力特性の基準を知るために，各コアで無限ループを行うプログラム，および iperf を用いて 1Gbps 通信を行ったときの消費電力を計測した．加えてアイドル時の消費電力を計測している．それぞれ，Xeon は 109.6 W，39.4 W，36.7 W，Atom は 27.1 W，25.0 W，24.2 W，Arm は 17.3 W，6.0 W，3.7 W となっている．

ここで，Arm はギガビットイーサネットのスイッチに接続しているが，Xeon と Atom とは異なり NIC が USB に接続されている．加えて Arm のドライバはまだ洗練されていないため，ネットワークバウンドなワークロードを処理する上でも `ksoftirqd` の CPU 使用率が高くなる傾向にある．

3.2 Workloads

上述の 3 台のマシンに Apache と Memcached へリクエストを送信するプログラムを，同じスイッチに繋がれたマシン上で稼働させた．このマシンは Xeon E3-1240 を搭載しており，16 GB のメモリを有している．

HTTP サーバや KVS を稼働させるサーバマシンの電力特性を抽出するために，人工的なワークロードを構築した．本研究ではリクエストのサイズに着目する．これらのサービスはリクエストを受理すると，その内容を解釈しデータを転送するという単純な動作を繰り返す．しかしながら，リクエストのサイズを変更することでサービスの資源使用量は大きく変わる．たとえば，小さいサイズのリクエストを多く受信すれば，それらを送り返すためにシステムコール (send() など) を多く発行する．この点に着目して，リクエストのサイズを変動させ，そのときのスループット・消費電力を計測する．ワークロードとして Apache に対しては ApacheBench を利用して 1KB，10KB，100KB のデータの要求・取得を繰り返す．Concurrency は 100 と設定している．Memcached に対しては 100 スレッドが同時に get 命令を行うワークロードを用意し，1KB，10KB，999KB のデータの要求・取得を繰り返す．

電力特性を抽出するために，それぞれのデータをリクエストする間隔を調整して，そのデータを処理するときに発揮するスループットおよび消費電力を詳細に調査した．CPU 性能が各サーバマシンで異なるため，同じワークロードでも CPU の処理時間が変わってくる．たとえば，Atom や Arm では処理に時間がかかるものが，Xeon ではすぐに処理できる．この際，Xeon の遊休時間が長くなり，電力面で有利になる可能性がある．この点を網羅するために，リクエストする間隔を調整しながらスループットおよび電力を測定した．具体的には 5 秒間隔を基準に，5 秒間リクエストを送り続ける (100%)，3.75 秒間リクエストを送り 1.25 秒スリープする (75%)，2.5 秒間リクエストを送り 2.5 秒スリープする (50%)，1.25 秒間リクエストを送り 3.75 秒スリープする (25%) 動作を行った．これらのパーセンテージを Density と呼ぶ．

4. Experimenta Results (Apache)

4.1 Power Consumption

実験結果を図 1 に示す．横軸が消費電力，縦軸がスループットを表す．図より，リクエストサイズを変動させることでプロセッサごとに最大スループットが異なることが分かる．1 KB のとき，Xeon が 50.1 MB/s という最も高いスループットを示している．特に 1 KB のときは Density を低下させていっても常に Xeon のスループットが他のマシンよりも高いことが分かる．一方で Xeon の消費電力は 45 W から 76 W 前後消費しており，どのマシンよりも高い結果となっている．Atom のスループットは Arm と変わらない．しかし消費電力が高いことから，電力当たりのスループットは Arm よりも低いといえる．この点については詳しく後述する．

10 KB のときにおいても，Xeon が他のマシンに比べて高いスループットを保っている．Xeon は最大で 110.5 MB/s のスループットを記録している．1 KB のときに比べると，Atom や Arm のスループットが Xeon のそれに近づきつつある．Atom では最大で 40 MB/s，Arm では 20 MB/s という値を記録している．電力面からみると，このスループットを達成したいときには，これらのマシンを使った方が電力面で有利であるということができる．

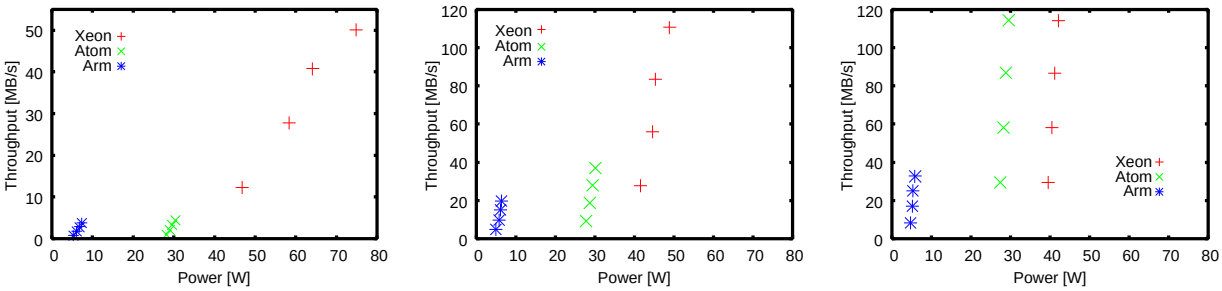
100 KB のとき，Xeon と Atom のスループットがほぼ同一となる．各 Density でほぼ同一のスループットを記録しており，Atom の消費電力が低いことから，Atom の方が電力当たりのスループットが高いを言える．Arm も 40 MB/s に近いスループットを発揮している．

電力当たりのスループットを表 1 にまとめる．表より，リクエストサイズ毎に電力当たりのスループットが異なることが分かる．1 KB のとき，Xeon がどの Density でも最も高い値となった．Density が低くなっていくと，電力当たりのスループットも低下していき，その電力効率は Arm を下回る．これは，サーバ計算機はアイドル状態でも一定の電力を消費することに起因する．アイドル時における消費電力の低下は急務な課題であると言える．

4.2 Resource Usage

Density が 100 % のときの資源使用率を図 2 に示す．図より，データサイズが小さいほど CPU 使用率が高いことが分かる．これは，データ転送量が小さいため，要求から取得までの時間が他のリクエストより短く，単位時間当たりに処理リクエスト量が多いことに起因する．結果として，Apache がシステムコールをより多く発行するため，CPU 使用率が高くなっている．そのため，1 KB のときには CPU バウンドなワークロードとなり，Xeon が高いスループットを記録している．

データサイズが大きくなっていくと，CPU 使用率が低くなっていき，ネットワーク使用量が高くなっていく．これは，データ転送に要する時間がより長くなるため，単位時間当たりのリクエストの処理数が低下していく．結果として，Apache が発行するシステムコール量が少なくなるため，ワークロードはネットワークバウンドとなる．結果



(a) 1 KB (b) 10 KB (c) 100 KB

図 1 Throughput and Power Consumption (Apache).

表 1 Throughput per Power Consumption (Apache)

	1 KB [KB/s / W]				10 KB [KB/s / W]				100 KB [KB/s / W]			
Density	100 %	75 %	50 %	25 %	100 %	75 %	50 %	25 %	100 %	75 %	50 %	25 %
Xeon	668.74	634.47	475.97	263.07	2258.29	1843.50	1256.27	665.90	2711.17	2104.74	1437.80	746.10
Atom	156.43	120.62	82.64	42.63	1230.70	950.17	654.62	334.43	3880.46	3007.03	2062.78	1076.05
Arm	563.52	458.12	336.47	188.30	3073.64	2455.28	1697.09	981.07	5567.51	4787.71	3260.49	1744.36

として，Atom や Arm が，高いスループット値を記録する結果となっている．

5. Experimental Results (Memcached)

5.1 Power Consumption

Memcached を用いた実験の結果を図 3 に示す．横軸が消費電力，縦軸がスループット消費電力を表す．図より，いずれの場合も Xeon と Atom のスループットが近い値を示していることがわかる．Apache の場合と比べると，1 KB のデータを取得する際にも Atom のスループットが Xeon のそれに近い．最大スループットは Xeon の 115.2 MB/s であるが，Atom も 88.3 MB/s と近い値を記録している．これは次の章で述べるが，Memcached はメモリ内のデータを検索して転送するのみなので，Apache と比べると CPU 時間をあまり要さない．結果として，ワークロードがネットワークバウンドになりがちになるため，Atom のスループットが高くなっている．Arm も 24.0 MB/s のスループットを出している．

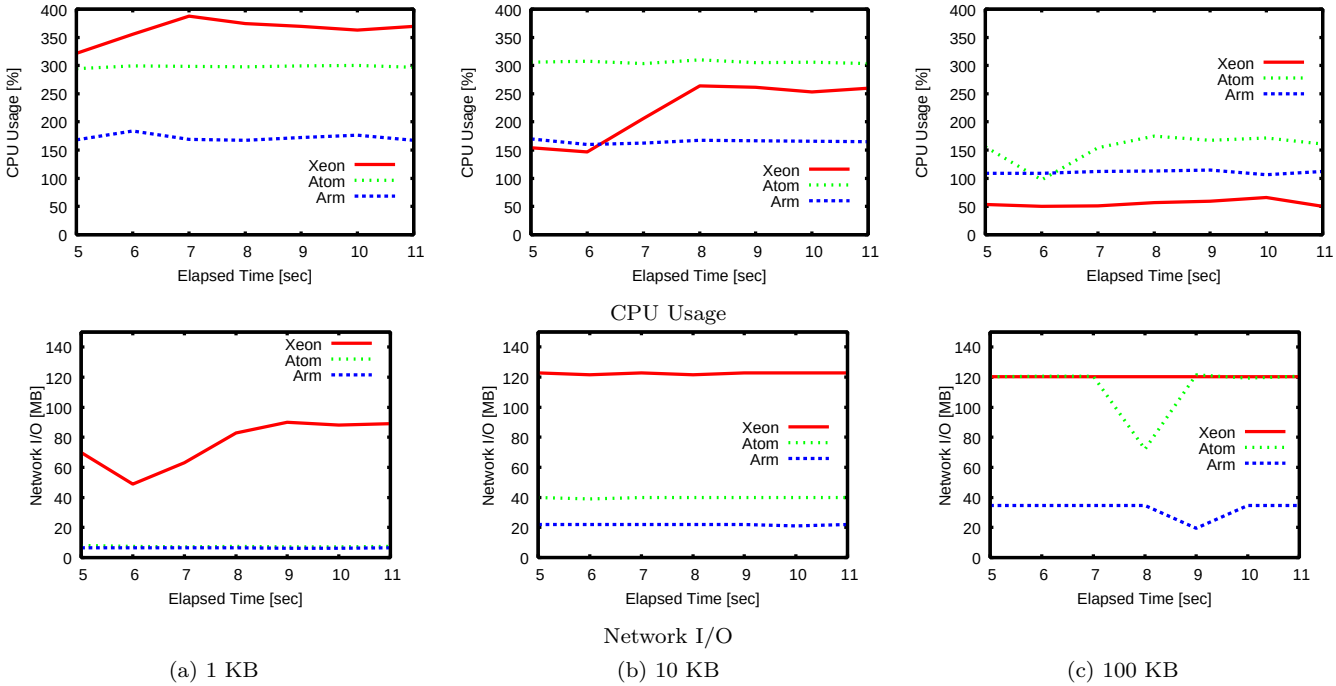
データサイズが大きくなると，Apache のときと同様，Xeon と Atom がほぼ同じスループットとなる．Density 100 % において，両者とも 110.4 MB/s のスループットを記録しておる．消費電力は Atom が低いため，電力面で見ると Xeon よりも Atom の方が有利となる．一方，Arm も 40.3 MB/s のスループットを発揮しており，本ワークロードで 40.4 MB/s を達成したいときには Arm を使用することが電力面では効果的である．

消費電力当たりのスループットを表 2 にまとめる．表より，Atom と Arm の値が総じて高いことがわかる．1 KB のときは Atom が 2967.67 と最も高い値を示している．Xeon は Atom よりも高いスループットを示していたが電力当たりのスループットは Arm よりも低いことがわかる．また，サイズを大きくすると，Arm の値が最も高くなる．10 KB のときが 6545.94，999 KB のときが 6003.86 である．ここで，Xeon や Atom はサイズが大きくなるにつれて電力効率が良くなっているにもかかわらず，Arm は効率が悪くなっていることに注意されたい．電力を要する CPU の処理が少なくなっていくため，電力効率は良くなるはずである．これは Arm のデバイスドライバが洗練されていないことに起因する．大量のデータ転送を行うと，`ksoftirqd` が過剰に動作するため，10 KB のときよりも 999 KB のときの方がスループットが低くなっていると考えられる．

5.2 Resource Usage

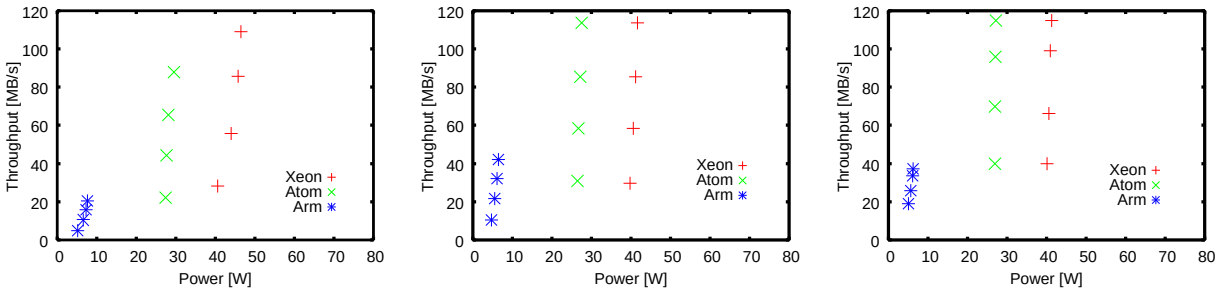
Density が 100 % のときの資源使用率を図 4 に示す．図より，データサイズが小さいほど CPU 使用率が高いが，すべての場合においてネットワークバウンドな処理を実行していることがわかる．Apache の場合と同様，データ転送量が小さいため，要求から取得までの時間が他のリクエストより短く，単位時間当たりに処理リクエスト量が多いことに起因すると考えられる．Apache の場合と比べると，ファイルを取得するためのシステムコールがないため，CPU 使用率はそれほど高くはならない．そのため，Xeon や Atom が高いスループットを記録している．

データサイズが大きくなっていくと，CPU 使用率が低くなっていき，ネットワーク使用量が高くなっていく．これも Apache の場合と同様，データ転送に要する時間がより長くなるため，単位時間当たりのリクエストの処理数が低下していくためである．Memcached では，10 KB のデータのやりとりにおいてすでにネットワークバウンドとなる．結果として，Xeon と Atom のスループットがほぼ



(a) 1 KB (b) 10 KB (c) 100 KB

図 2 Resource Usage (Apache).



(a) 1 KB (b) 10 KB (c) 999 KB

図 3 Throughput and Power Consumption (Memcached).

同一となっている．

6. Discussion

実験を通して，リクエストするデータサイズによって適切なサーバマシンを選択することで，消費電力当たりのスループットを向上できる．データが小さいものは CPU バウンドとなりやすいため，Xeon といったサーバに任せた方がよい．特に Apache の結果は顕著である．データが大きいものはネットワークバウンドとなりやすいため，Arm や Atom といったサーバに任せた方がよい．これにより，展開するサービスのファイルサイズに応じてサーバ計算機を使い分けることで，消費電力当たりのスループットを高くすることができる．

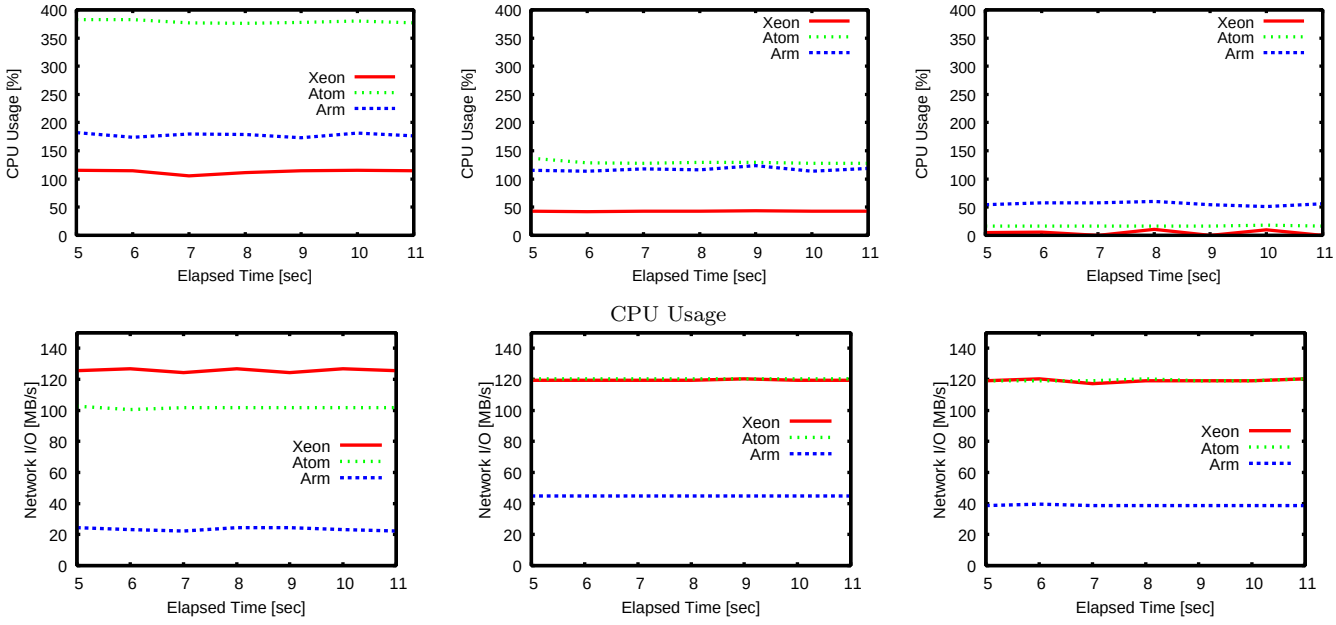
KVS を用いた実験では，KVS 自身がネットワークバウンドになりやすいことがわかった．これは KVS がキーを受信したら該当する値を返すというシンプルな動作をすることに起因する．そのため，KVS を稼働させる際には，Arm や Atom といったサーバマシンを利用することで大幅な消費電力削減が見込める．KVS は広く展開されており，Facebook でも大規模に展開されているため [14]，低消費電力サーバを活用することで，飛躍的に消費電力を削減することが期待できる．

Arm の消費電力効率は Web サーバや KVS いずれの場合も顕著である．そのため，Arm を複数台用意して Xeon よりも消費電力当たりのスループットを高められるかを検証する必要がある．KVS はもともとスケールアウトが可能なアプリケーションなので，Arm を複数台並べてその上で KVS を稼働させて，その消費電力特性を計測する必要がある．複数の KVS を調停するための機構が動作するため，それによって CPU 使用率やネットワーク I/O 量が増えるためである．

仮想マシン技術を考慮する必要がある．仮想マシン技術を用いて，複数サーバを同一マシンに集約し，必要となるマシンを削減できることが知られている．これにより，Xeon 1 台で複数の仮想マシンを稼働させた方がよいのか，Arm を複数台用意してサービスを稼働させた方がよいの

表 2 Throughput per Power Consumption (Memcached)

	1 KB [KB/s / W]				10 KB [KB/s / W]				999 KB [KB/s / W]			
Density	100 %	75 %	50 %	25 %	100 %	75 %	50 %	25 %	100 %	75 %	50 %	25 %
Xeon	2342.19	1875.03	1268.27	696.25	2730.09	2079.58	1440.89	774.58	2779.01	2419.80	1627.17	994.56
Atom	2967.67	2327.75	1601.97	807.99	4132.12	3141.17	2141.41	1092.76	4228.29	3538.76	2590.48	1480.97
Arm	2678.50	2159.01	1738.49	969.94	6545.94	5205.19	3887.85	2220.94	6003.86	5696.33	4603.17	3738.86



(a) 1 KB (b) 10 KB (c) 999 KB

図 4 Resource Usage (Memcached).

かは今後検証すべき課題である．その際，本研究で作成したベンチマークを利用することで，リクエストサイズに応じたサーバの配置方法が明確になる．

受信するリクエストの種類が変動したら，その特性に合わせて利用するサーバ計算機を切り替えることが理想的である．そのため，KnightShift [7] といった手法や，適切なサーバマシンへとサービスを移送する技術が必要となる．移送に関して，たとえば，リクエストされるデータサイズが小さいときには Xeon に配置し，サイズが大きくなったら Arm に配置することが可能な技術が消費電力を削減する上では重要となる．仮想化技術が用いられているデータセンタでは，Arm から Xeon，またその逆は CPU アーキテクチャが異なるマシン間での移送になるため，現在の技術では移送は不可能である．そのため，CPU アーキテクチャが異なるマシン間の移送を実現する技術が必要となる．

7. Conclusion and Future Work

本稿では Xeon，Atom，そして Arm という 3 台のサーバ計算機をケーススタディとして，消費電力特性を定量的に測定した．具体的にはデータセンタの典型的なワークロードである HTTP サーバや KVS において，そのリクエストサイズを変えただけでも電力当たりのスループットは変化し，リクエストに応じて利用するマシンを切り替えることで，電力当たりのスループットを最大化できることがわかった．

今後は仮想化によるサーバ統合を用いたときの消費電力特性，および複数台の上で Web サーバや KVS を稼働させたときの消費電力特性を定量的に示し，データセンタの構築法やサーバ計算機の構成法，およびその上で稼働するシステムソフトウェアの構成法への足がかりとしたい．

参考文献

[1] HP Thermal Logic technolog: Control power and cooling for data center efficiency (2006).
[2] Qureshi, A., Weber, R., Balakrishnan, H., Guttag, J. and Maggs, B.: Cutting the Electric Bill for Internet-Scale Systems, *Proc. of the 2009 ACM Conference on Data Communication (SIGCOMM '09)*, pp. 123–134 (2009).
[3] Ahmad, F., Chakradhar, S., Raghunathan, A. and Vijaykumar, T. N.: Tarazu: Optimizing MapReduce On Heterogeneous Clusters, *Proc. of the 17th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '12)*, pp. 61–74 (2012).
[4] David G. Andersen and Jason Franklin and Michael Kaminsky and Amar Phanishayee and Lawrence Tan

and Vijay Vasudevan: FAWN: A Fast Array of Wimpty Nodes, *Proc. of the 22nd ACM Symposium on Operating Systems Principles (SOSP '09)*, pp. 1–14 (2009).

[5] Tudor, B. M. and Teo, Y. M.: On Understanding the Energy Consumption of ARM-based Multicore Servers, *Proc. of the 2013 ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '13)*, pp. 267–278 (2013).

[6] Verma, A., Dasgupta, G., Nayak, T. K., De, P. and Kothari, R.: Server Workload Analysis for Power Minimization using Consolidation, *In Proc. of the 2009 USENIX Annual Technical Conference (ATC '09)*, pp. 355–368 (2009).

[7] Wong, D. and Annavaram, M.: KnightShift: Scaling the Energy Proportionality Wall Through Server-Level Heterogeneity, *Proc. of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO '12)*, pp. 119–130 (2012).

[8] Meisner, D., Gold, B. T. and Wenisch, T. F.: PowerNap: Eliminating Server Idle Power, *Proc. of the 14th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '09)*, pp. 205–216 (2009).

[9] Malladi, K. T., Nothaft, F. A., Perivathambi, K., Lee, B. C., Kozyrakis, C. and Horowitz, M.: Towards Energy-Proportional Datacenter Memory with Mobile DRAM, *In Proc. of the 39th Annual International Symposium on Computer Architecture (ISCA '12)*, pp. 37–48 (2012).

[10] Horvath, T., Abdelzaher, T., Skadron, K. and Liu, X.: "Dynamic Voltage Scaling in Multitier Web Servers with End-to-End Delay Control", *IEEE Transactions on Computers*, Vol. 56, No. 4, pp. 444–458 (2007).

[11] Gandhi, A., Harchol-Balter, M., Das, R. and Lefurgy, C.: Optimal Power Allocation in Server Farms, *Proc. of the 11st ACM International Joint Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '09)*, pp. 157–168 (2009).

[12] Meisner, D., Sadler, C. M., Barroso, L. A., Weber, W.-D. and Wenisch, T. F.: "Power Management of Online Data-Internsive Services", *In Proc. of the 38th Annual International Symposium on Computer Architecture (ISCA '11)*, pp. 319–330 (2011).

[13] Isci, C. and Martonosi, M.: "Runtime Power Monitoring in High-End Processors: Methodology and Empirical Data", *Proc. of the 36th ACM/IEEE Annual International Symposium on Microarchitecture (MICRO '03)*, pp. 93–104 (2003).

[14] Nishtala, R., Fugal, H., Grimm, S., Kwiatkowski, M., Lee, H., Li, H. C., McElroy, R., Paleczny, M., Peek, D., Saab, P., Stafford, D., Tung, T. and Venkataramani, V.: Scaling Memcache at Facebook, *Proc. of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI '13)* (2013).