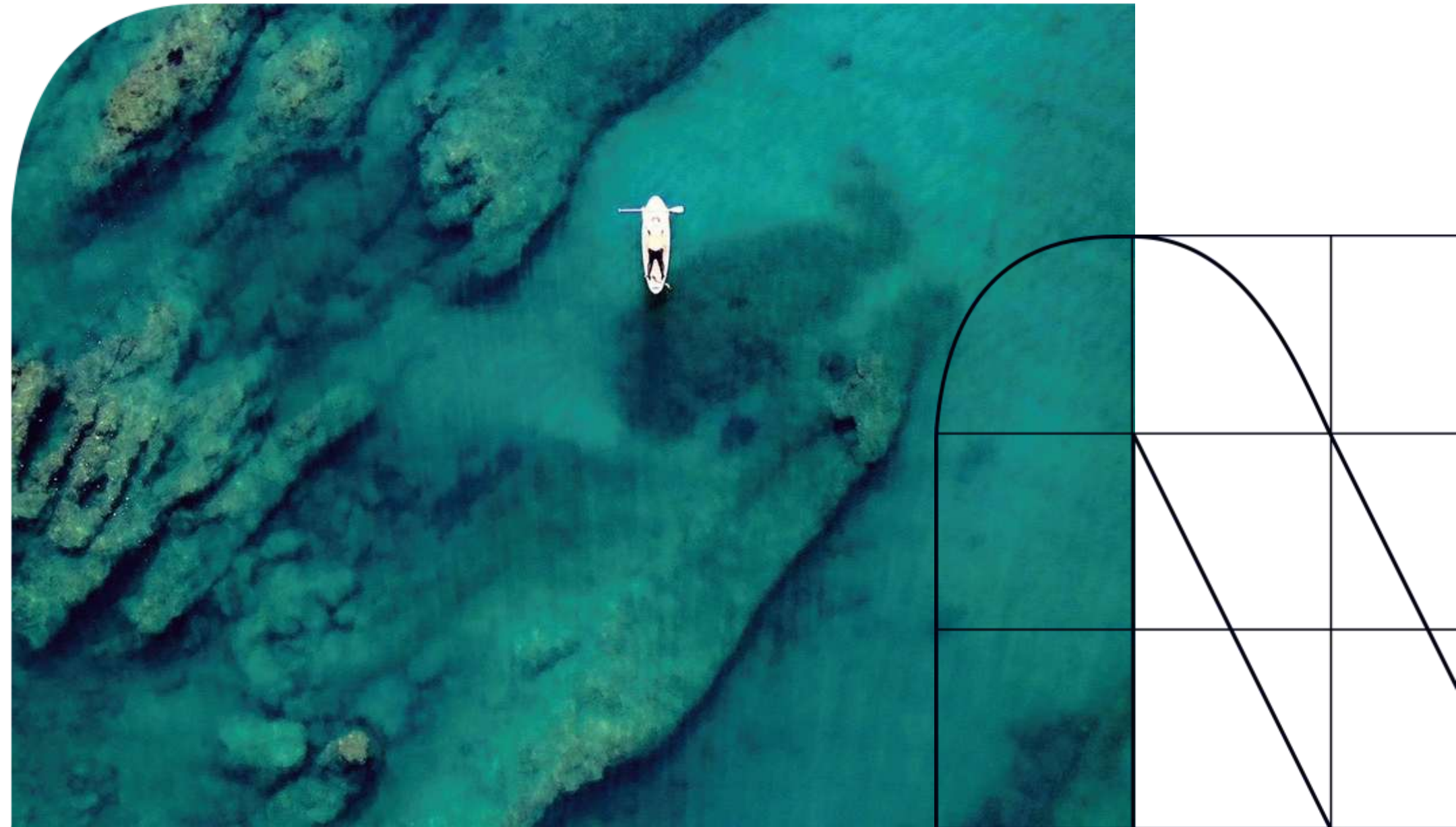


Armサーバにおける Javaアプリケーションの電力効率

～Javaアプリケーションの消費電力から見えるArmサーバ活用の利点～



背景

サステナビリティ（持続可能性）は現代の組織・企業にとって主要な優先事項である。その取り組みや成果について、利害関係者や株主向けの情報公開も増えつつある。特に企業活動による環境負荷は重要な指標の1つとなっており、脱炭素化に向けた取り組みが各所で加速している。今日「グリーン化」は意思決定のキーポイントとなっている。

ITシステムは業界を問わず、企業活動においてますます重要になってきている。IT機器やインフラ自体が二酸化炭素（CO₂）を直接排出することはないが、それらを動作させるために必要な電力の発電時にCO₂が排出されることが多い。CO₂排出量削減のためには利用する電力の由来（再生可能エネルギー、化石燃料、原子力など）を意識することはもちろんだが、最も重要なことは「電力消費量を減らす」ことである。Ampere® Altra® プロセッサを搭載するHPE ProLiant RL300はそれを目的に設計されたサーバである。

Ampere® Altra® プロセッサはArm64（AArch64）アーキテクチャを採用している。このアーキテクチャはスマートフォンや組み込み機器のように省電力が必須となる分野では以前から採用実績がある。最近ではHPC（High Performance Computing）分野*1をはじめ、オンプレ・クラウドを問わず適用領域が広がっている。

Armアーキテクチャの電力性能比の高さはかねてより評価されていたが、特にJavaベースのものを含むエンタープライズアプリケーションやベンチマークにおいてもその優位性がアピールされている。

NTT DATAではJavaで開発したシステムを多くのお客様にお使いいただいていることから、JavaアプリケーションをサステナブルなITインフラに移行することはNTT DATAの環境目標の達成のみならず、お客様のサステナビリティ実現にも貢献する。そこで、ArmベースのサーバであるRL300のJavaで開発されたエンタープライズアプリケーションにおける省電力効果を検証することにした。

*1 <https://www.arm.com/markets/computing-infrastructure>
*2 <https://github.com/Macchinetta/atrs>

検証

Armサーバの省電力効果を検証するため、プロセッサの処理性能を同等としたRL300とx86サーバ（x86-64アーキテクチャのIntelプロセッサ）（表1）それぞれでアプリケーションを実行し、消費電力に着目した。x86ベースのシステムは2 CPU 28コア（ハイパースレッディングにより56スレッド）、Armベースのシステムは1 CPU 128コア（128スレッド）のサーバを使用した。実行したものは以下の2つである。

- SysBench CPUベンチマーク
- Javaアプリケーション (RESTful Webサービス: ATRS *2)

表1. 各サーバに搭載するプロセッサのベンチマークスコア

サーバ	プロセッサ	アーキテクチャ	ベンチマークスコア (SPECrate2017 int_base)
x86	Intel Xeon Gold 6330 x2 sockets	x86-64	359
Arm	Ampere M128-30 x1 sockets	Arm64	366

省電力設定は表2および表3に記載のとおり実施した。

SysBenchのCPUベンチマークは10,000以下の素数をマルチスレッドで探索するものである。各サーバの消費電力はこの探索を60秒間実行している最中に計測した。ただし、前述のとおりArmサーバとx86サーバで論理コア数（スレッド数）が異なるため、各サーバで2つの並列度で計測を実施した。1つはx86サーバに合わせた112スレッド、もう1つはArmサーバに合わせた128スレッドである。

各サーバはHPE製で、HPE Integrated Lights-Out (iLO) によって管理することができるが、iLOはサーバ管理の業界標準であるRedfishに準拠したAPIも提供する。Redfish APIを呼び出すことで消費電力を含むサーバのさまざまな情報を取得できる。消費電力はRedfish APIを呼び出してPowerリソース情報を取得し、レスポンスに含まれるPowerControl項目のPowerConsumedWattsの値から確認できる。そのためシステム全体の消費電力計測のための測定機器類は一切不要であった。

表2. x86サーバの省電力設定

オプション	設定値
Power Regulator	Dynamic Power Savings Mode
Minimum Processor Idle Power Core C-State	C6 State
Minimum Processor Idle Power Package C-State	Package C6 (non-retention) State
Intel® Turbo Boost Technology	Enabled
Energy Performance Preference	Disabled
Energy Performance Bias	Balanced Performance
Collaborative Power Control	Enabled
Intel DMI Link Frequency	Auto
NUMA Group Size Optimization	Flat
Intel Performance Monitoring Support	Disabled
Uncore Frequency Scaling	Auto
Sub-NUMA Clustering	Disabled
Energy Efficient Turbo	Enabled
Local/Remote Threshold	Auto
LLC Dead Line Allocation	Enabled
State A to S	Disabled

表3. Armサーバの省電力設定

オプション	設定値
APEI Support	Enabled
CPPC Support	Enabled
LPI Support	Enabled

検証結果

SysBench CPU ベンチマーク

図1は素数探索における1秒あたりの計算処理数を比較したものである。Armサーバはx86サーバと比べて3倍以上の計算処理（SysBenchではeventsと表現される処理単位での比較）を実行した。図2では試験パターンごとの消費電力を示しているが、スレッド数に関係なくArmサーバの消費電力が低かった。128スレッド（Armサーバの論理コア数と同数）のパターンにおける消費電力を見ると、Armサーバでは350 Wだったのに対し、x86サーバでは490 Wの電力を消費していた。これは140 W、パーセンテージにして40%ほどArmサーバの消費電力がx86サーバよりも低かったことを示している。さらに重要なのは、1 Wあたりのパフォーマンス（Performance per Watts）で見たとき、Armサーバが今回比較したx86サーバに比べ4.8倍近い性能を示したことだ。Armサーバは 1305 events/s per W（456,806 events/s を 350 W で処理した）、x86サーバは 267 events/s per W（130,842 events/s を 490 W で処理した）といった結果となった。これらのことからNTT DATAはArmサーバが高い処理性能を持ちながらも少ない消費電力で動作することを実際に確認できた。

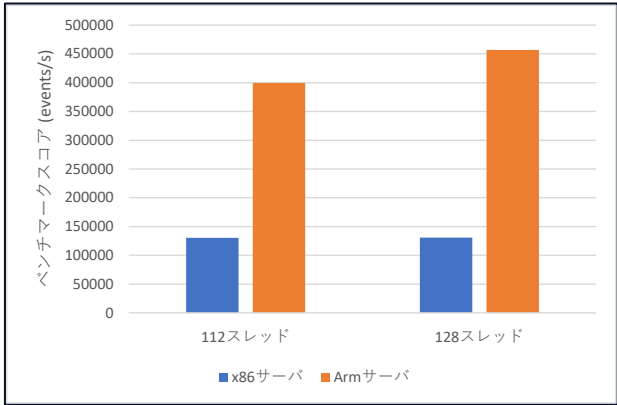


図1. SysBench測定結果

Javaアプリケーション

同じJavaアプリケーションを各サーバ上で実行し、同量の負荷を同一時間かけ続けたときの筐体消費電力を測定した。使用したアプリケーションはエンタープライズ向けの典型的なWebアプリケーションを想定している。HTTPリクエストに含まれるパラメータをもとにリレーショナルデータベースから必要なデータを取得し、処理結果をJSON形式でクライアントへ送り返すものである。OSにはRocky Linux 8.7を、Javaはバージョン19.0.2を使用した。なお、アプリケーション実行時にスレッド数の調整等 Java 仮想マシン（JVM）のチューニングを行うようなオプションは指定していない。JVMは論理プロセッサ数に基づいてスレッド数を自動調整するようになっている。

アプリケーションには別のサーバから負荷試験ツールGatlingによって負荷がかけられた。負荷は15分間かけているが、最初の5分間は徐々にユーザー数を増やし、5分後に最大ユーザー数に到達してから10分間同じユーザー数のまま負荷をかけ続けた。

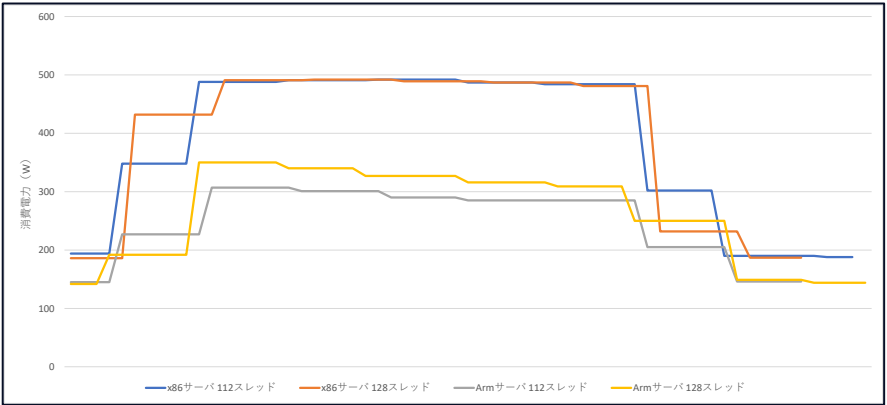


図2. SysBench実行中の消費電力

試験の最中は毎秒Redfish APIを呼び出し消費電力を記録した。JavaはプログラムをJVMで動作させるアーキテクチャであることから、対応するJDK（Java SE Development Kit）さえあればアプリケーションは変更や再コンパイルすることなく、どの環境でも動作させることができる。JDKは以前からx86-64だけでなく64ビットArmプロセッサ（AArch64）もサポートしていた。そのため、追加の作業なくアプリケーションをArm上で実行することができた。測定結果を図3に示す。

グラフで消費電力が安定した期間に着目すると、x86サーバは 380 W の電力を消費したのに対し、Armサーバは 240 W の消費にとどまった。CPUベンチマーク同様、Armサーバがx86サーバに比べて消費電力が140 W 少ない結果となった。Armサーバ、x86サーバともに主に動作していたのは検証用のJavaアプリケーションのみであったため、Armサーバはx86サーバに比べて約37%少ない消費電力でアプリケーションを動作させたとみなすことができる。この結果はArmアーキテクチャの電力性能比の高さを証明している。さらに印象的だったのは、アイドル状態（145 W）から高負荷状態（240 W）まで、Armサーバはわずか 95 W しか増加しなかったのに対し、x86サーバは 195 W から 380 W と、185 W の消費電力増で、負荷による消費電力増はArmサーバがx86サーバの半分だったということだ。

多くのWebアプリケーションが昼夜問わず動作し続けることを考慮すると、この検証で確認した省電力効果はさらに重要なものになる。ハードウェアの消費電力はWebアプリケーションにおける消費電力の大きな要因の1つであることから、サーバをArmサーバに変更するだけでこれほど大きな省電力効果が得られることを確認できたのは今後省電力化を進めていく上で実りある一歩であると考えている。これはサーバのアーキテクチャを変更する際にアプリケーションの変更や再コンパイルが基本的に不要であるJavaでは特にメリットが大きくなる。

NTT DATAでは、それぞれのサーバにおいてさまざまなJavaの実行時オプションを指定して検証を実施したが、いずれのパターンでも本検証結果と消費電力に大きな違いが見られなかった。この理由としてJavaの最適化が十分に行われればハードウェア利用の面で大きな差は生まれず、それゆえ消費電力に影響がなかったと見ている。このような結果はJavaアプリケーションだけでなく他のプログラミング言語で開発されたアプリケーションにも当てはまり得るものだが、正確なパフォーマンスと電力効率の向上はソフトウェアとツール（ランタイムやコンパイラを含む）のArm64に対する実装成熟度に強く依存する。近年、Javaはオープンソース実装であるOpenJDKにおいてArm64への対応や改善が積極的に行われているため、Armサーバの省電力効果を、懸念なく最大限に享受することが可能だ。他のプログラミング言語やソフトウェアの場合はArmサーバへの移行プロセスや結果（効果）がJavaとは異なる場合も考えられるため、注意が必要だ。

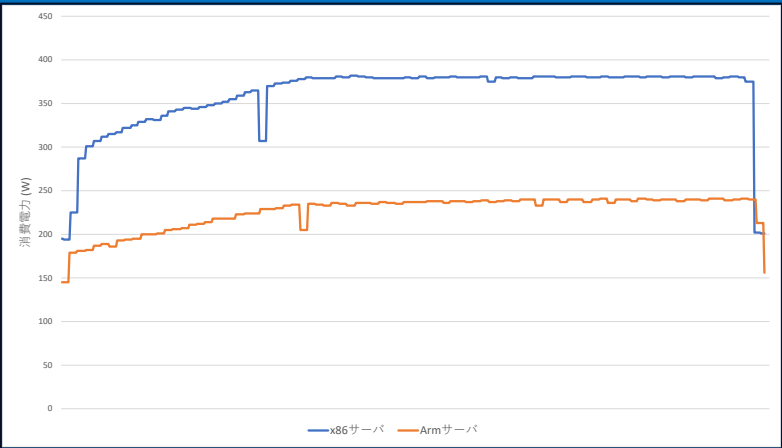


図3. Javaアプリケーション実行時の消費電力

まとめ

検証 – CO₂ 排出量削減の影響

x86サーバをArmサーバへ移行することによって見込まれる3年間のCO₂ 排出削減量を試算した。

想定条件

- 1ラックに21台のx86サーバが搭載されており、それらすべてをArmサーバへ移行する。
- サーバは24時間365日の運用とする。
- 電力料金とCO₂排出係数は以下のとおりとする。
 - 電力料金（円／kWh）：
季節や時間帯による料金差を考慮し、東京電力の年間平均値を参照する。（参考リンク #2）
 - CO₂排出係数（t-CO₂/kWh）：
東京電力がWebサイトで公表するCO₂排出係数を参照する。（参考リンク #3）

算出結果

x86サーバをArmサーバに移行することで年間のCO₂排出量を約10t削減でき、3年間では約30tの削減につながることが判明した。

ラックごとのCO ₂ 排出量 (kg-CO ₂ /Rack)			
年	x86サーバ	Armサーバ	削減量
1年	26,350	16,601	9,750
2年	52,700	33,201	19,499
3年	79,050	49,802	29,249

まとめ

「グリーンソフトウェア」という言葉を聞くと、ソフトウェアの文脈ではアルゴリズムのように動作効率の向上による省電力化がまず最初に思い浮かぶ。しかし、プログラムは最終的にはハードウェア上で実行されるものである。そのため、省電力化はソフトウェアとハードウェアの両方で考慮する必要がある。Armサーバはハードウェア省電力化への明確な寄与が期待できる。NTT DATAにおける検証の結果より、Armサーバの適用領域はHPCにとどまらず、エンタープライズ領域にも広げられることがわかった。RL300に採用されているAmpere Altra®とAltra® Maxプロセッサはクラウドのワークロードにおけるパフォーマンスと動作効率向上にフォーカスして設計されているが、これらの検証結果より、IaaS・PaaS・SaaSにおけるパフォーマンス向上とカーボンフットプリント削減も重要な要素となるだろう。

パフォーマンスと省電力の両立において、Armサーバはより広い領域で活用されていくだろう。

参考リンク

- [HPE ProLiant RL300 Gen11](#)
- [東京電力 電気料金単価表](#)
- [東京電力 CO₂排出係数](#)