スーパーコンピュータ「富岳」における システム省電力に向けた取り組み

2023年1月20日

富士通株式会社

ミッションクリティカルシステム事業本部

UNIX＆FXシステム事業部 FX運用管理ソフトウェア部

篠原　誠

© 2023 Fujitsu Limited

FUJITSU

自己紹介

1998年入社

インターコネクト、ファイルシステム関連のHPCソフト開発に従事
現在、ジョブスケジューラを含むシステムソフト全体を担当

© 2023 Fujitsu Limited

「富岳」の目標と開発

FUJITSU

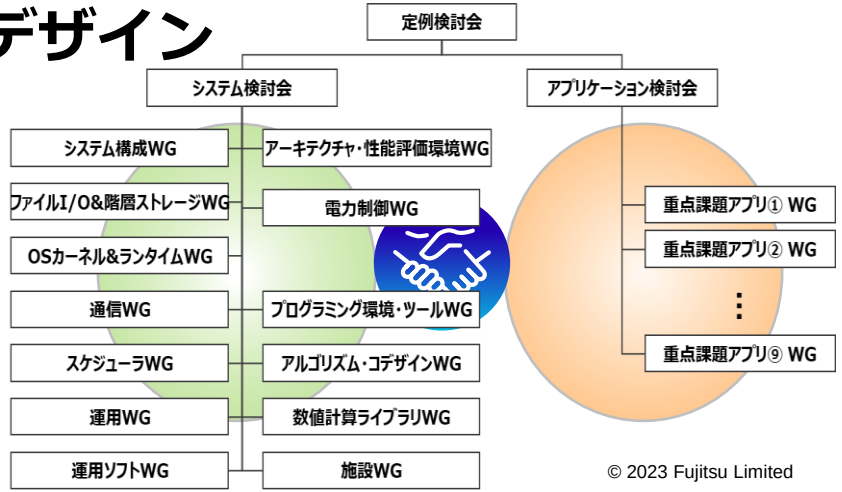
- **目標**

引用：富岳コデザイン・レポート ～フラッグシップ2020プロジェクト・テクニカルレポート～
https://www.r-ccs.riken.jp/wp/wp-content/uploads/2022/03/fs2020-report-j.pdf

  - 「京」の 100 倍のアプリケーション実効性能
  - 高い電力効率

目標	目標値	
システム電力	30～40MW	「京」の施設を再利用
電力性能	15GFlops/W	幅広いユーザの活用のため、汎用CPUのみで構成

- **アプリケーションとシステムによるコデザイン**
  - 重点課題のターゲットアプリケーションの実行効率の向上を目指して、共同設計。



消費電力を抑えるために、アプリケーションの特性に応じた電力制御を検討

© 2023 Fujitsu Limited

「富岳」の実績

- スーパーコンピュータ「富岳」は2021年3月から共用開始
  - 2019年6月にプロトタイプでGREEN500の世界一を獲得
  - 2020年6月に史上初のHPC４冠を獲得（TOP500, HPCG, HPL-AI, Graph500）
    2021年11月まで４期連続で世界一を獲得
  - 2021年11月ゴードン・ベル賞COVID-19研究特別賞を受賞

© 2023 Fujitsu Limited

「富岳」の電力性能

FUJITSU

2020/11の最大ピーク性能到達時にGREEN500でもトップ10入り
電力あたり性能は汎用CPUではトップ、「京」の約20倍を達成

ランク	システム名	アクセラレータ無し	コア数	演算性能 [TFlop/s]	システム電力 [kW]	電力当たり性能 [GFlops/W]
1	**NVIDIA DGX SuperPOD**	-	19840	2356	90	26.20
2	**MN-3**	-	1664	1652.9	65	26.04
3	**JUWELS Booster Module**	-	449280	44120	1764	25.01
4	**Spartan2**	-	23040	2566	106	
5	**Selene**	-	555520	63460	2646	
▶ 6	**A64FX prototype**	✓	36864	1999.5	118.5	**16.88**
7	**AiMOS**	-	130000	8339	512	16.28
8	**HPC5**	-	669760	35450		15.74
9	**Satori**	-	23040	1464		15.57
▶ 10	**Supercomputer Fugaku**	✓	7630848	**442010**	**29899**	**15.42**
			:			
▶ **2011/11**	**K-Computer**	✓	705024	**10510**	**12660**	**0.83**

目標達成 15GFlops/W超

目標達成 30～40MW以内

Top500 No1

×42

×2.4

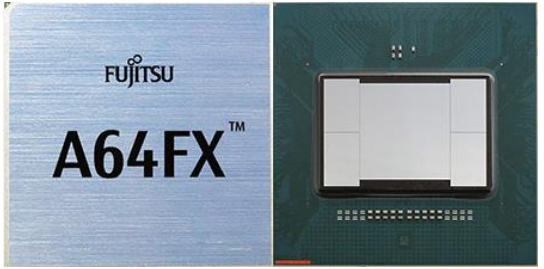
×20

×19

© 2023 Fujitsu Limited

Fujitsu Arm CPU A64FX

- **Armアーキを HPC向けに拡張**
  - Armと協業し、HPC向け拡張(SVE*)の策定に、リードパートナーとして貢献
  - SIMD*を512bitに拡張
- **省電力性能・電力制御機能を強化**
  - **SIMD関連の消費電力削減**
  - **パワーノブ**
  - **リテンション**



仕様	
命令セットアーキテクチャ	Armv8.2-A SVE (512bit SIMD)
演算コア数	48 コア (12コア x 4 CMG)
アシスタントコア数	計算ノード: 2コア IO兼計算ノード: 4コア
倍精度演算性能	3.3792 TFLOPS
メモリバンド幅	1,024 GB/s
スレッド並列サポート	コア間HWバリア
スケーラビリティ	TofuインターコネクトD内蔵

* SVE: Scalable Vector Extension
* SIMD: Single Instruction Multiple Data

© 2023 Fujitsu Limited

FUJITSU

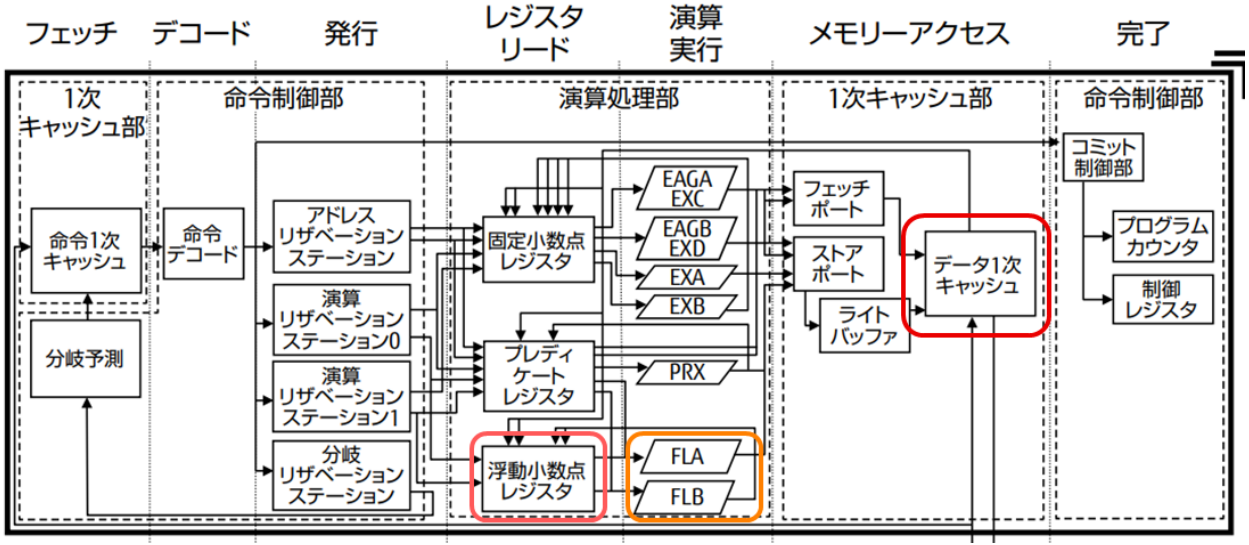
A64FXの省電力

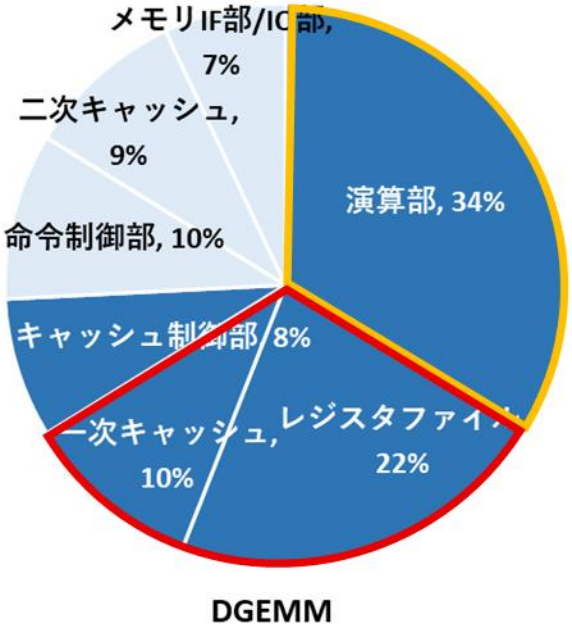
・SIMD関連の消費電力削減

© 2023 Fujitsu Limited

A64FXの消費電力割合

- 高頻度で動作するSIMD関連部の割合が多い(74%)
  - 性能のために、演算部は常に動く必要があり、電力削減はできない
- 演算部へのデータ供給に着目
  - レジスタファイル、一次キャッシュの電力削減を検討





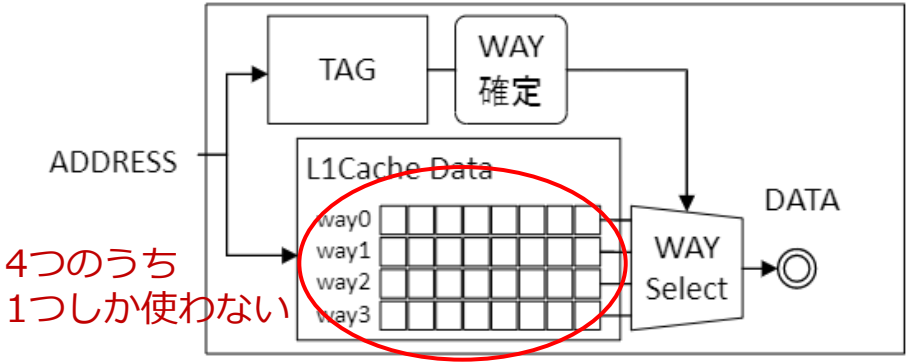
© 2023 Fujitsu Limited

SIMD関連の消費電力削減

FUJITSU

データの読み出しを減らすための2つの手段を検討

一次キャッシュの課題



・データ候補の4つのWAYを読み出し
・TAGの一致を確認してからWAYを選択

使うのは1WAYのみなので残りは無駄

Long Latency L1 Cache Access

レジスタファイルの課題

```
fmla   z10.d, p1/m, z0.d, z22.d
fmla   z11.d, p1/m, z1.d, z22.d
fmla   z12.d, p1/m, z2.d, z22.d
fmla   z13.d, p1/m, z3.d, z22.d
```

同じデータを
何度も参照

・何度も参照されるレジスタが存在

同じデータを何度も読み出すのは無駄

Operand Register Bypass

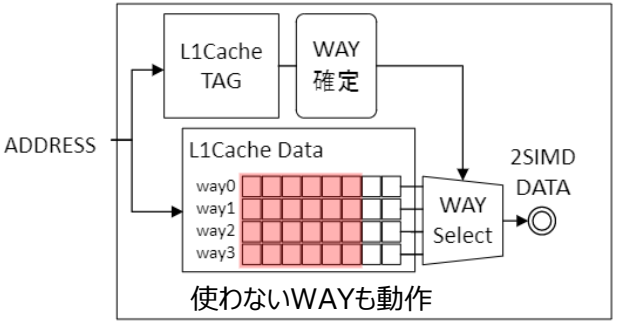
© 2023 Fujitsu Limited

Long Latency L1 Cache Access

FUJITSU

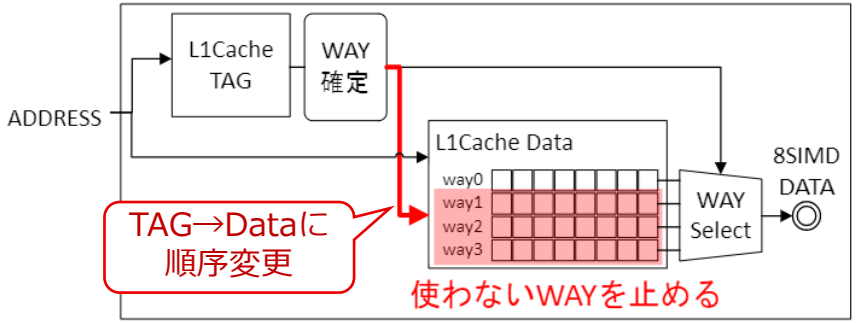
データ幅とパイプラインの状態で2種類のレイテンシで動作

従来：Short Latency



- 128bit以下の狭いデータ幅
- WAY確定とデータ読み出しが並列
  →データ読み出し後にWAY選択
- レイテンシ重視のアプリを想定

新手法：Long Latency



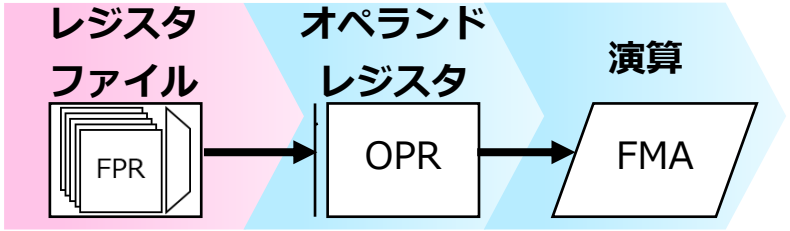
- 256bit以上の広いデータ幅
- WAY確定後にデータ読み出し
  →不要WAYの読み出しをストップ
- スループット維持
  →大規模データの並列処理に性能影響小

© 2023 Fujitsu Limited

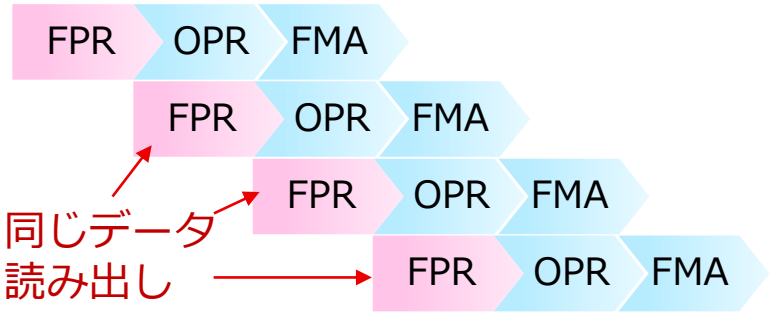
Operand Register Bypass

オペランドレジスタへのデータ経路を追加し、読み出しを削減

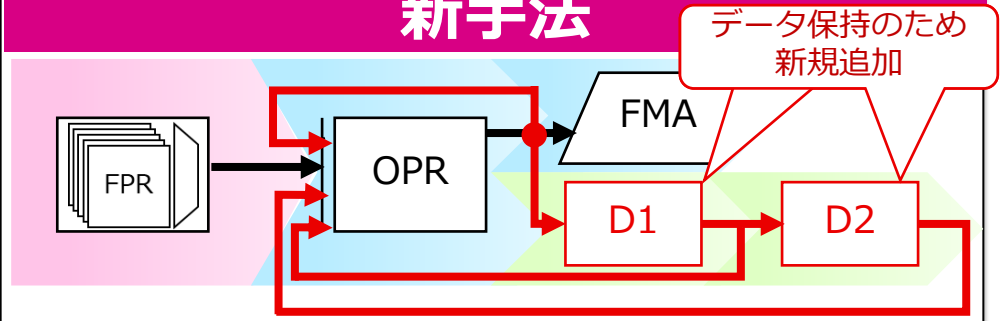
従来



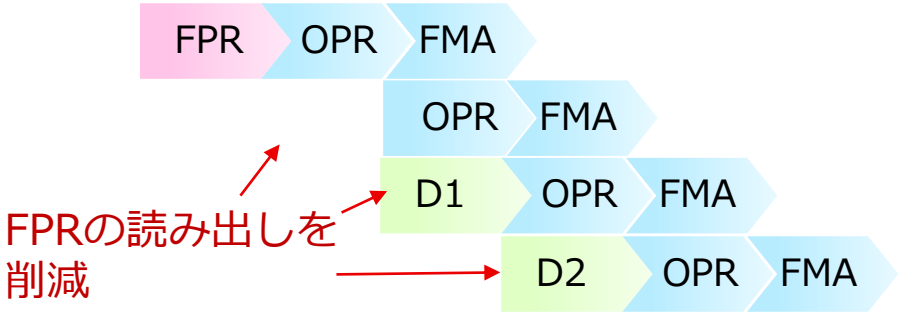
・同じデータを連続して使う場合も、
FPRからの読み出しが毎回発生



新手法



・小規模回路の追加で読み出しを3回削減し、
性能を維持したまま省電力化



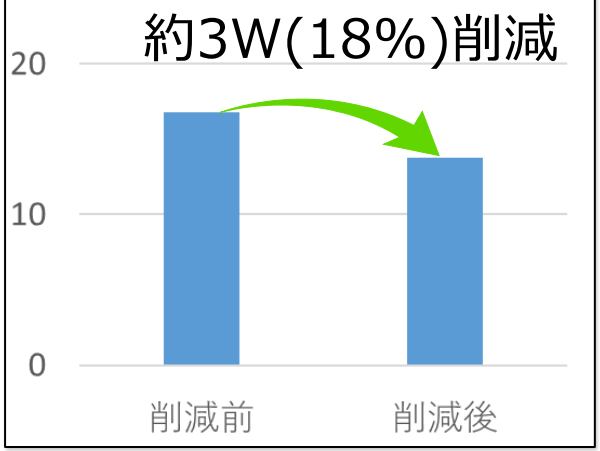
© 2023 Fujitsu Limited

消費電力削減の効果

期待通りに消費電力を削減、電力性能目標15GFlops/W超えを達成

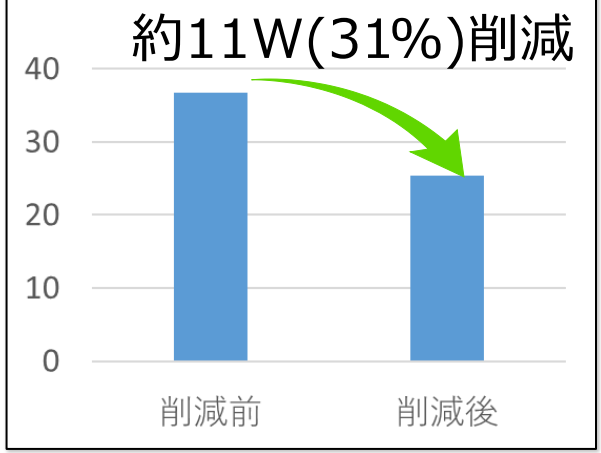
Long Latency
L1 Cache Access

一次キャッシュ電力削減



Operand Register
Bypass

レジスタファイル電力削減



電力対性能

16.88 GFlops/W達成

© 2023 Fujitsu Limited

FUJITSU

A64FXの電力制御機能

・パワーノブ

© 2023 Fujitsu Limited

A64FXの電力制御：パワーノブ

FUJITSU

性能を下げずに余分な電力を削減するため、パワーノブを採用

アプリケーション特性に応じて、使用していない部分を停止し、電力効率を向上

パワーノブ	概要	設定可能な値
周波数変更	**コア全体のCPU周波数を制御**	**・2.2GHz（ブースト） ・2.0GHz（ノーマル） ・1.6GHz**
浮動小数点演算ユニット制御	**使用可能な浮動小数点演算器の数を制御 エコモード(※)の制御 ※電力安定化のための電力消費を抑えるモード**	**・FLA/FLB、エコモード無効** ・FLA Only、エコモード無効 **・FLA Only、エコモード有効**
固定小数点演算ユニット数	使用可能な固定小数点演算器の数を制御	・EXA/EXB ・EXA Only
命令発行数	コアが同時に処理する命令数を制御	・4命令 ・2命令
メモリスロットリング	メモリアクセスコントローラーとメモリ間のバス使用率を制御	・バス使用率 100%～10% 10%単位

© 2023 Fujitsu Limited

パワーノブの効果

FUJITSU

アプリケーション特性により、効果はそれぞれに異なる

電力モード	周波数	浮動小数点演算
ノーマル	2.0	FLA/FLB, エコモード無効
ブースト	2.2	FLA/FLB, エコモード無効
エコ	2.0	FLA Only, エコモード有効
ブーストエコ	2.2	FLA Only, エコモード有効

ノーマルの性能,電力を1.0とした場合の相対値

アプリ	ブースト		エコ		ブーストエコ	
	性能	電力	性能	電力	性能	電力
GENESIS	1.10	1.10	0.97	0.70	1.06	0.80
Genomon	1.05	1.13	1.00	0.75	1.05	0.81
GAMERA	1.09	1.10	0.90	0.80	0.97	0.85
NICAM+LETKF	1.09	1.09	0.99	0.82	1.04	0.86
NTChem	1.04	1.14	0.67	0.76	0.73	0.86
ADVENTURE	1.05	1.09	0.95	0.91	1.00	0.77
RSDFT	1.09	1.14	0.71	0.77	0.80	0.91
FFB	1.00	1.08	0.98	0.81	1.02	0.88
LQCD	1.06	1.11	0.92	0.79	1.00	0.84

引用：コデザインによるスーパーコンピュータ「富岳」プロセッサの開発
https://www.jstage.jst.go.jp/article/essfr/15/4/15_300/_pdf/-char/en

© 2023 Fujitsu Limited

FUJITSU

A64FXの電力制御機能

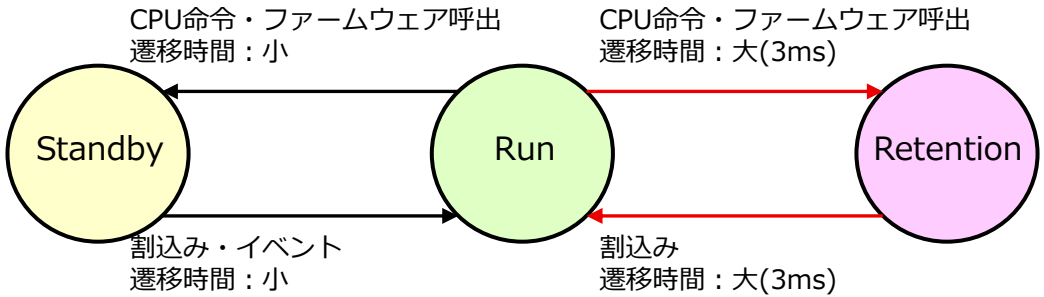
・リテンション

© 2023 Fujitsu Limited

A64FXの電力制御：リテンション

使用しないアイドル状態のCPUコアを停止し、消費電力を削減

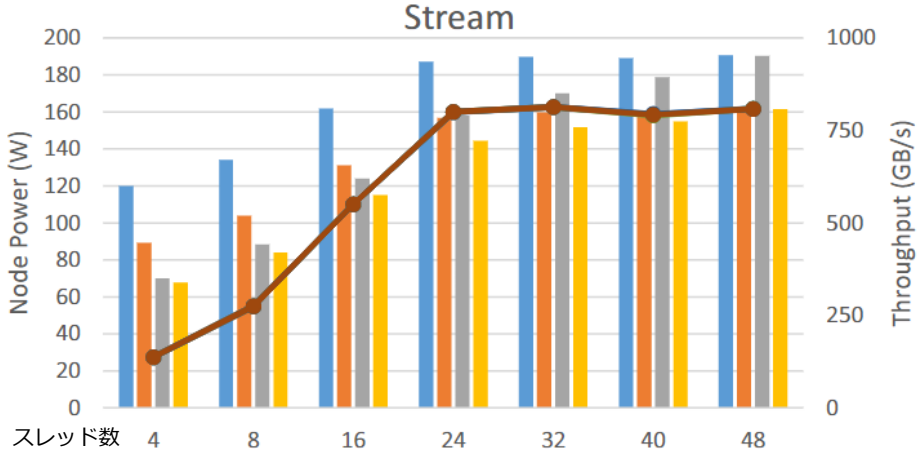
- コア単位に有効／無効の制御が可能。
  演算コアはジョブから制御可、アシスタントコアはシステム管理者からのみ。
  ノード全体をリテンション状態(ノードリテンション)にすることも可能。
- パワーノブと組み合わせた運用が可能。
  例：周波数2.2GHz（ブースト） ＆ リテンション有効
- 遷移契機はWFI(Wait For Interrupt)命令による割り込み待ち。

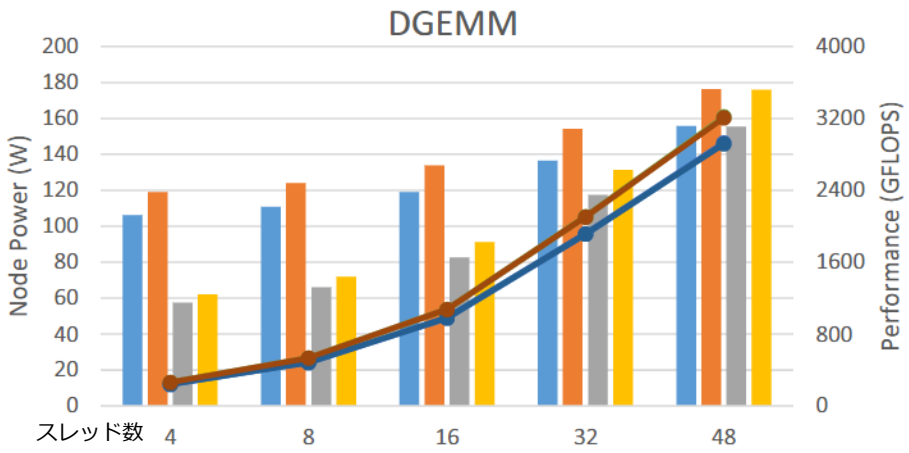


© 2023 Fujitsu Limited

リテンションの効果

使用しないコアがあるほど、リテンション効果は大きい





ノーマル: リテンション無効、エコモード無効	**リテンション：** **リテンション有効**、エコモード無効
エコ： リテンション無効、**エコモード有効**	**エコリテンション：** **リテンション有効、エコモード有効**

- 性能はほぼ同一、リテンションでの差はなし
- スレッド数が小さいほど、リテンション効果大

ノーマル: リテンション無効、周波数2.0GHz	**リテンション：** **リテンション有効**、周波数2.0GHz
ブースト： リテンション無効、**周波数2.2GHz**	**ブーストリテンション：** **リテンション有効、周波数2.2GHz**

- 性能は周波数による違い、リテンションでの差はなし
- スレッド数が小さいほど、リテンション効果大

引用：富岳コデザイン・レポート ～フラッグシップ2020プロジェクト・テクニカルレポート～
https://www.r-ccs.riken.jp/wp/wp-content/uploads/2022/03/fs2020-report-j.pdf

© 2023 Fujitsu Limited

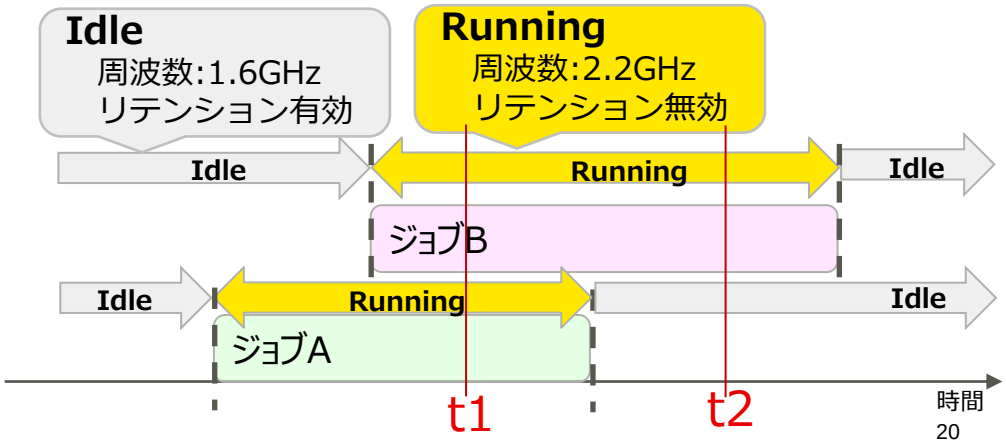
FUJITSU

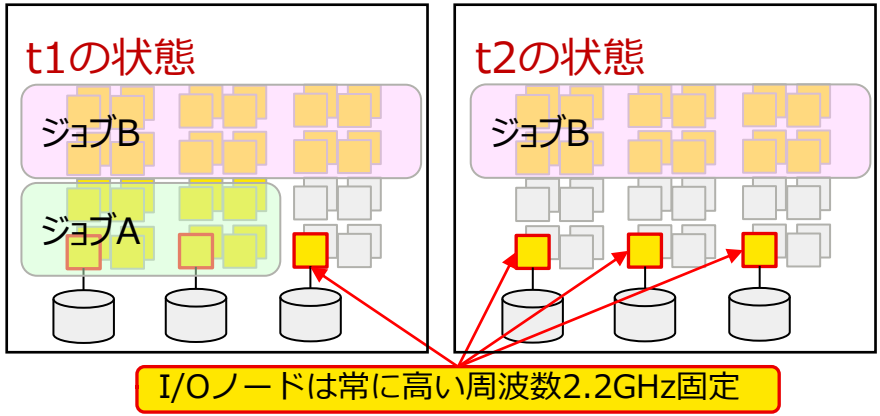
システムソフトウェアによる電力制御の活用

© 2023 Fujitsu Limited

ジョブ運用ソフトウェアによる活用

- **ジョブ実行に合わせた省電力モード切替**
  - ジョブの開始・終了に応じてモード（Idle/Running）を自動的に切り替え
  - 運用に合わせて、システム管理者はそれぞれのモードの設定を定義可能
- **ノード種別を意識した電力制御**
  - 計算ノードはディスクレスのため、ファイルI/OはI/Oノードを経由
  - I/Oノードにファイル I/O性能影響を考慮した設定が可能（例：2.2GHz固定）





© 2023 Fujitsu Limited

Power APIの提供

- **Power API** (Power Application Programming Interface)
  - Sandia National Laboratoriesが提唱した電力計測・制御をするためのライブラリIF
    参考：https://powerapi.sandia.gov/
  - システム管理者だけでなく、エンドユーザも利用可能
  - エンドユーザがAPIを利用して、ジョブ内で電力計測や電力制御を実現可能

電力計測

Attribute	計測項目
PWR_ATTR_ENERGY	推定電力量
PWR_ATTR_MEASURED_ENERGY	実測電力量

電力制御

Attribute	パワーノブ・リテンション
PWR_ATTR_FREQ	周波数変更
PWR_ATTR_THROTTLING_STATE	メモリアクセス制限
PWR_ATTR_ISSUE_STATE	命令発行制限
PWR_ATTR_EX_PIPE_STATE	固定小数点演算ユニット
PWR_ATTR_ECO_STATE	浮動小数点演算ユニット、エコモード
PWR_ATTR_RETENTION_STATE	リテンション制御

統計情報

Attribute	項目
PWR_ATTR_POWER	電力

© 2023 Fujitsu Limited

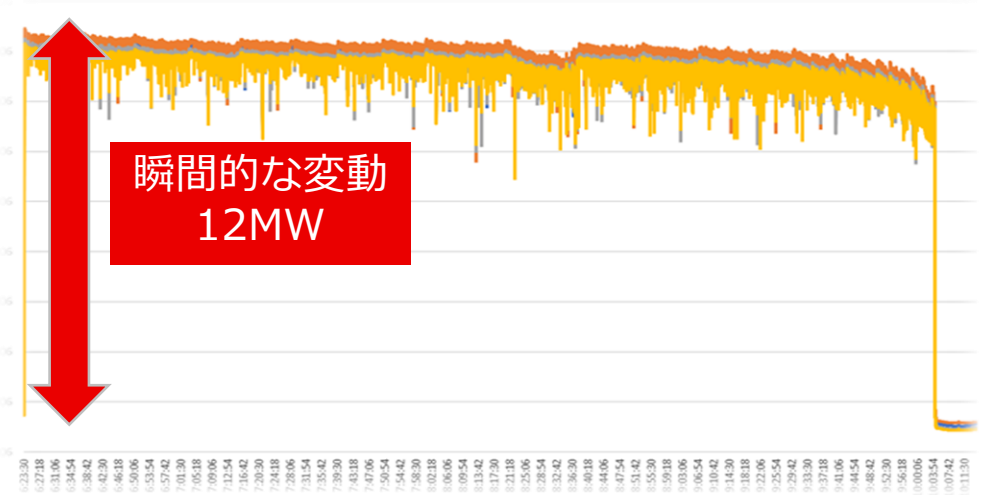
システム電力超過への取り組み

© 2023 Fujitsu Limited

システム消費電力の変動によるリスク

- 「富岳」のシステム消費電力は30～40 MWに設定
- アプリによっては、全系実行時の消費電力を考慮すると設定電力を超過する可能性もあると判明。
- 2020/9のTop500測定時、施設の最大電力は32MWを記録。消費電力変動は12MWと膨大。

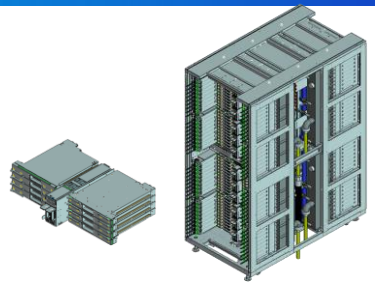
設定電力超過は施設・設備に深刻なダメージを与えるため、対策が必要



© 2023 Fujitsu Limited

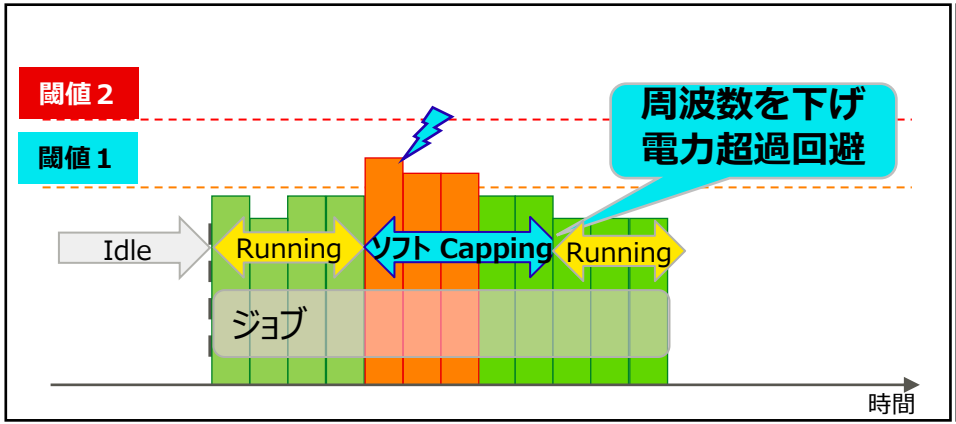
BoBの消費電力キャッピング

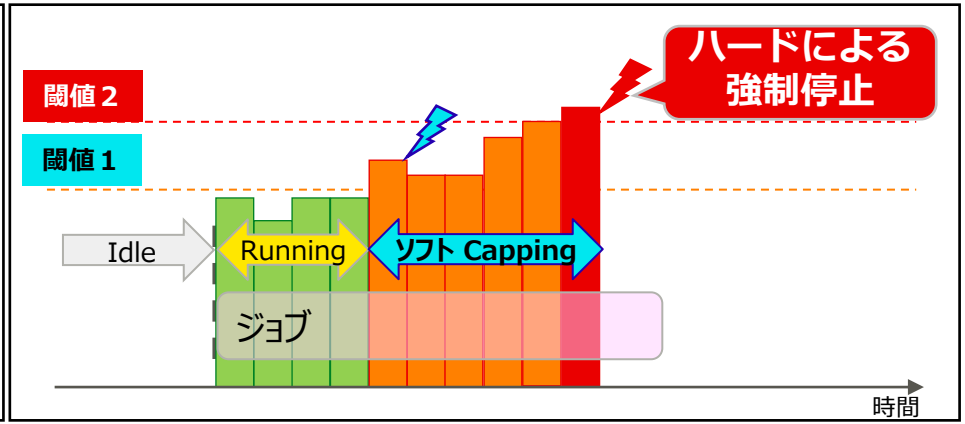
施設保護のため、許容電力超過を短時間で抑制



24BoB = ラック

- キャッピングはBoB単位を選択
  ラックよりも短時間での監視・制御が可能
- ２つの閾値を設けて、段階的な消費電力キャッピングを実現
  第１段階：ソフトがBoB内のジョブ実行中の計算ノードのCPU周波数を低下
  第２段階：ハードがBoB内のすべての計算ノードを強制停止





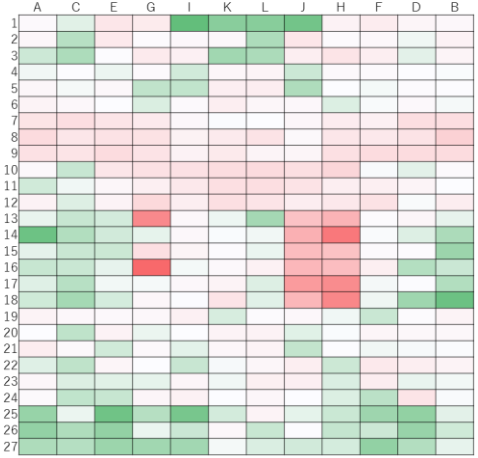
© 2023 Fujitsu Limited

現地で発生した問題と対策

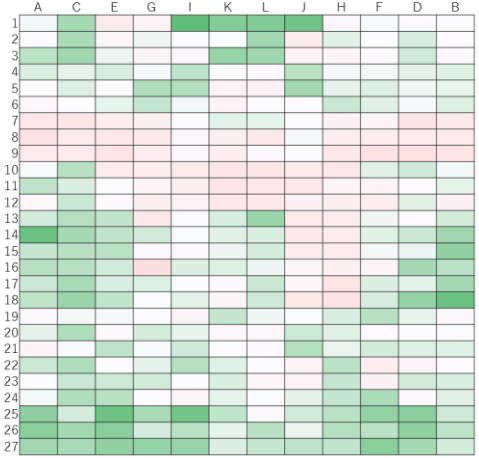
不要なキャッピング発生を抑制し、ジョブ運用の安定化に貢献

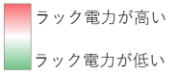
問題	“意図しない”消費電力キャッピングが発生
原因	ノード内電力のバラつきによる、BoB単位の消費電力差
対策	ラック単位にノード内電力を測定し、消費電力の高低を確認。 消費電力の高いノードと低いノードを入替ることで消費電力を平準化

BoB電力平準化前



BoB電力平準化後





ノード交換したBoB数：全体の6%
ノード交換作業　　：1500回

© 2023 Fujitsu Limited

まとめ

- **高い電力性能を実現したA64FXの省電力・電力制御**
  - SIMD関連の消費電力削減
  - パワーノブ
  - リテンション
- **システムソフトウェアによる電力制御の活用**
- **現場一丸で取り組んだキャッピング安定化の事例**
- **今後に向けて**

  ますます重要性を増す電力に対して、
  電力を意識した運用はこれから本格化。
  幅広く安心して利用いただけるよう取り組みます。

© 2023 Fujitsu Limited

Thank you

© 2023 Fujitsu Limited