科学技術計算分科会 2020年度会合
富岳スペシャル2.0　～より深く詳しく～

FUJITSU
shaping tomorrow with you

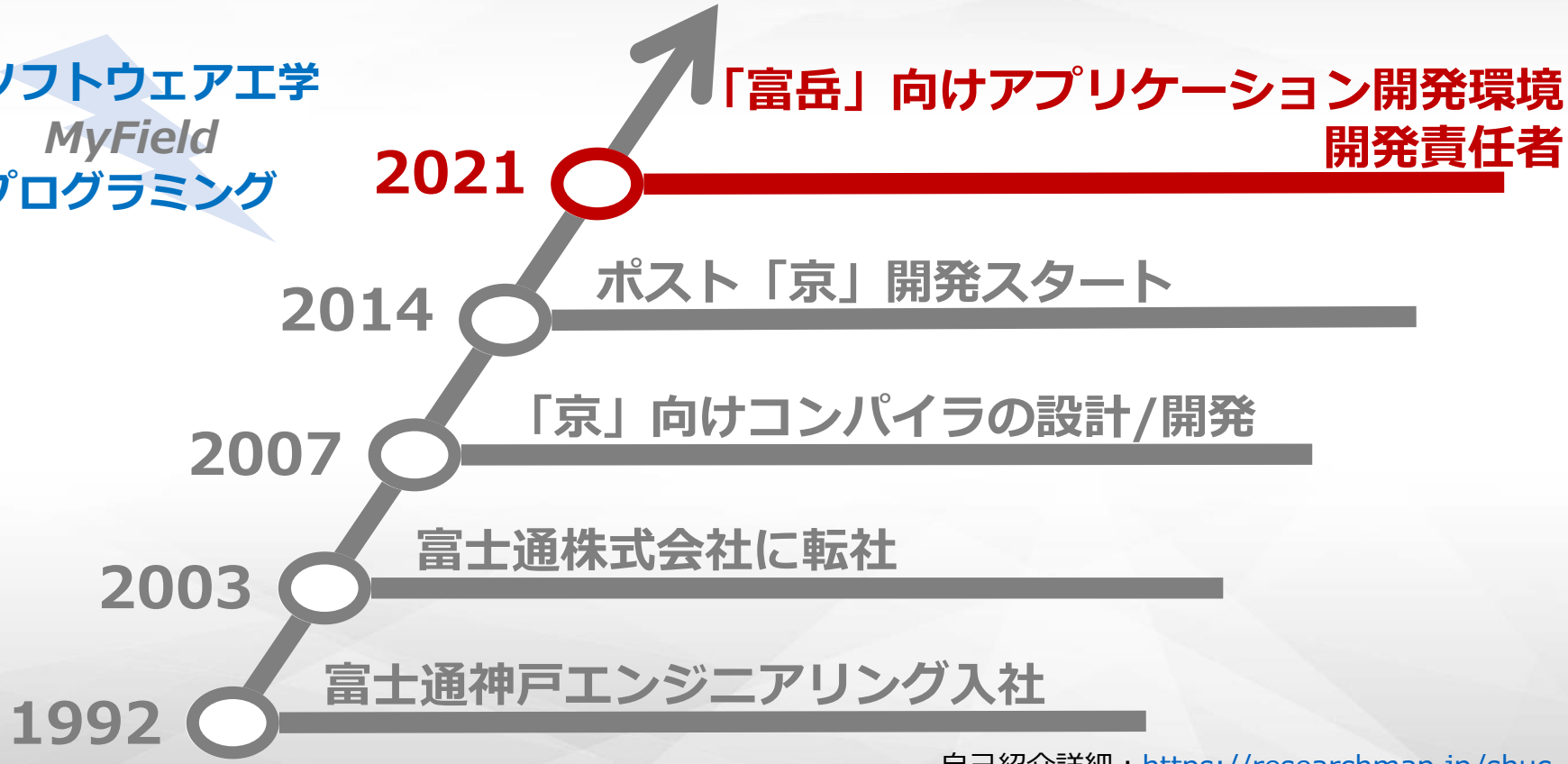
スーパーコンピュータ「富岳」を支えるコンパイラの技術

2021年 1月 21日

富士通株式会社
プラットフォームソフトウェア事業本部
千葉　修一

Profile

FUJITSU



自己紹介詳細：https://researchmap.jp/shuc

Outline

FUJITSU

- “コンパイラ”をより深く
  ～コンパイラの役割～
  - ✓コンパイラとは
  - ✓最適化の必要性
- “コンパイラ”と”富岳”をより詳しく
  ～コンパイラの作り方～
  - ✓コデザイン
  - ✓実用化・商品化・事業化

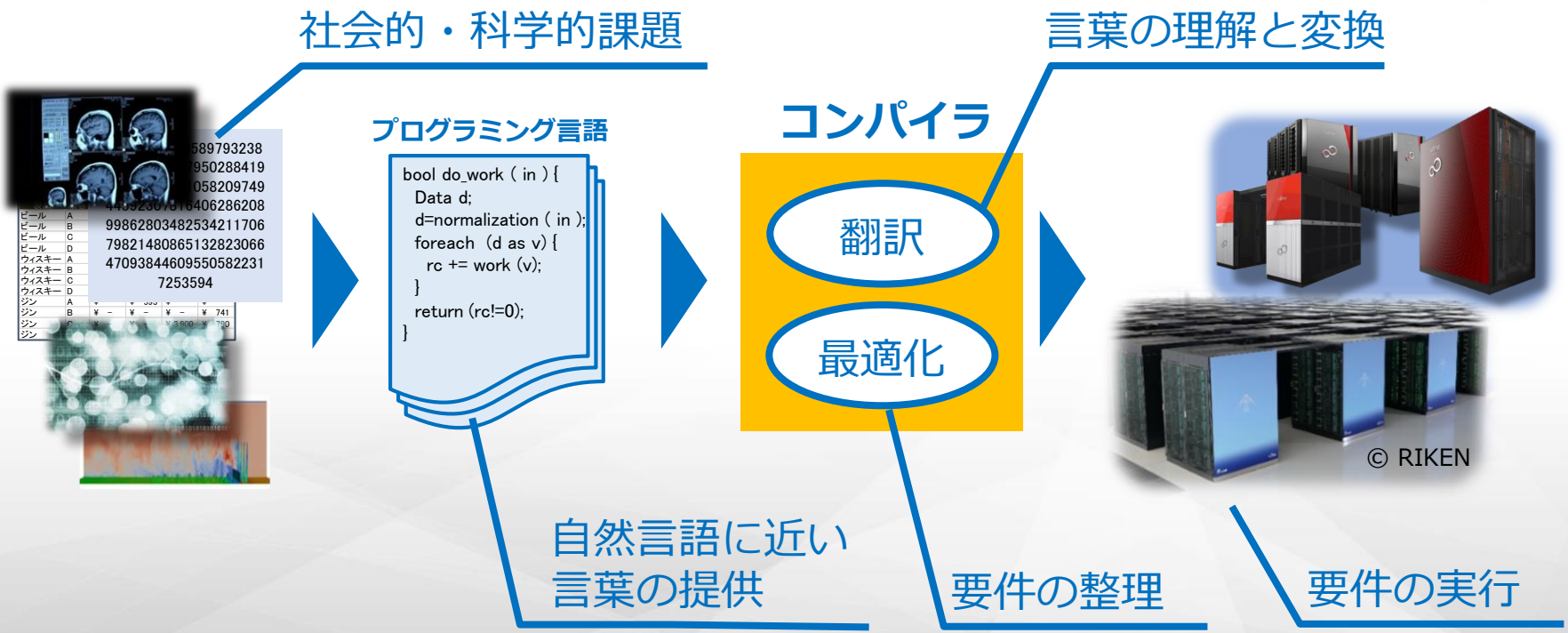
FUJITSU

“コンパイラ”をより深く

コンパイラとは

FUJITSU

プログラミング言語を介しコンピュータと容易に会話するためのツール

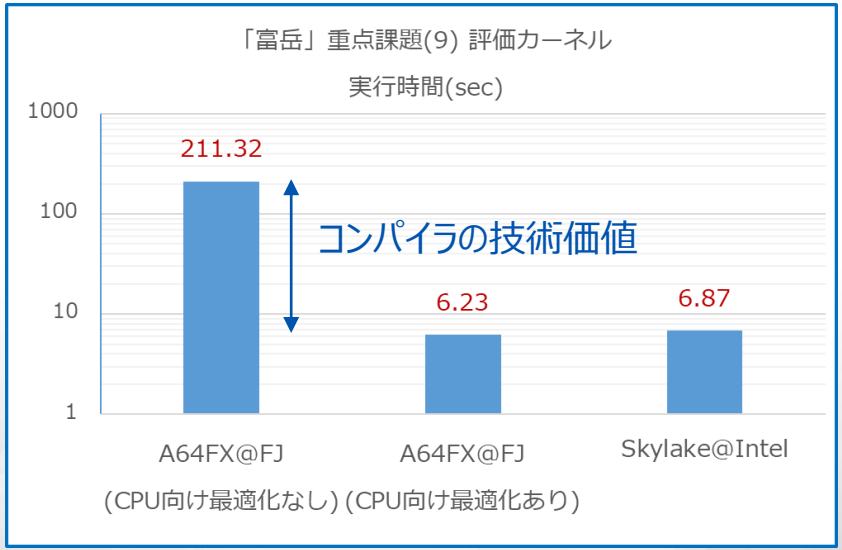


最適化の必要性

FUJITSU

コンパイラの最適化によりRISCアーキテクチャの最高性能を引き出す

- ✓ 基本命令を組み合わせることで複雑な命令を実現
- ✓ パイプライン処理を意識し効率的な命令順序にスケジューリング



FUJITSU

“コンパイラ”と”富岳”をより詳しく

コデザイン

FUJITSU

システムとアプリケーションの協調設計による汎用スパコンの開発

- ✓ プロセッサアーキテクチャやアプリケーションのアルゴリズムなど全体を紐づけた設計
- ✓ システム・ハードウェアが利用できない状態でのソフトウェアの設計／開発

ポスト「京」（富岳）

	京	富岳
ISA	SPARC64	**Armv8-A**
extension	HPC-ACE	**SVE**
FP registers	256	**32**

ハードウェアの設計中に開発スタート

コンパイラ

- 翻訳
- 最適化

富士通の技術

コンパイラの開発技術

最適化技術だけでなく、ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ言語の規格委員などを保有し、業界に深く精通

「京」の開発経験

CPU、ネットワーク、ソフトウェア、開発環境など大規模計算を実現、提供するノウハウ

コデザイン

FUJITSU

システムとアプリケーションの協調設計による汎用スパコンの開発

- ✓ プロセッサアーキテクチャやアプリケーションのアルゴリズムなど全体を紐づけた設計
- ✓ システム・ハードウェアが利用できない状態でのソフトウェアの設計／開発

SPARC-V9	京 SPARC64™VIIIfx	富岳 A64FX
32	256	32

～「京」の開発当初のコメント～　★大容量オシ★
コンパイラはこの大容量レジスタを用いてソフトウェアパイプライニングなどの最適化を行い、アプリケーションが持つ命令レベルの並列性を最大限に引き出す

extension	[illegible]ACE	SVE
FP registers	256	32

最適化

「富岳」での課題

- 最適化は可能か？
- 性能を引き出せるか？

アプリケーションの性能を引き出せるか

コンパイラの挑戦

コンパイラ内で仮想「富岳」を実装し最適化の効果を検証

- ✓ 「京」、FX100向けのコンパイラを改造し、レジスタ数の変動による影響を検証
- ✓ X-Gene 2向けコンパイラを作成し、Armアーキテクチャによる影響を検証

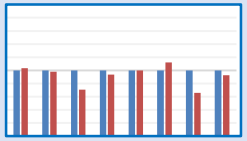
評価対象

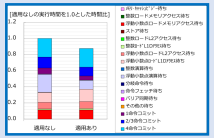
1. 基礎性能コードによる評価
   - 命令セットアーキテクチャ、レジスタ数の違いによる効果の変化
2. ベンチマークおよび数値シミュレーションによる評価
   - レジスタ数の変化による実用性の評価
   - アプリケーションの移行性を検証

評価方法

① 最適化の有無による実効性能の変化を比較

② プロファイラによる実行時のコスト分布を分析





コンパイラの挑戦

FUJITSU

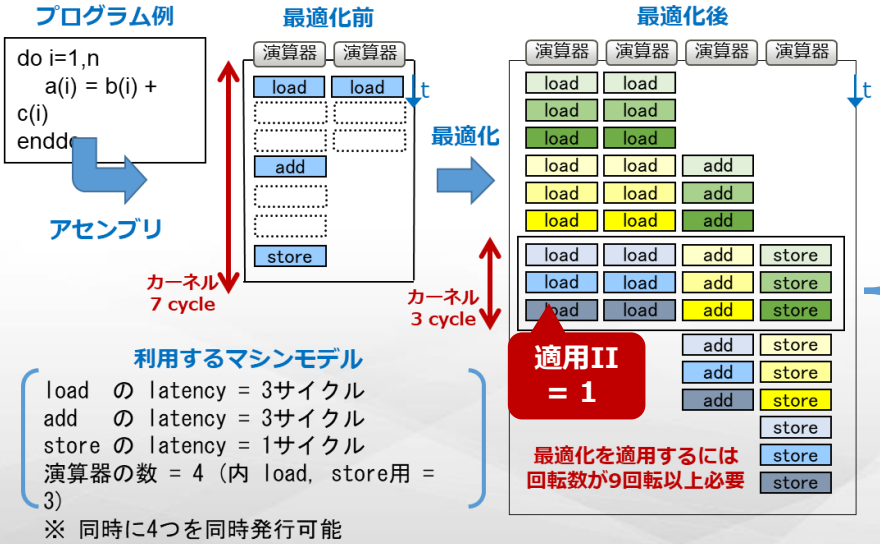
評価環境

	SPARC64™ XIfx (128レジスタモード)	SPARC64™ XIfx (32レジスタモード)	X-Gene 2
Architechure	**SPARC64**	SPARC64	**Armv8-A**
Peak performance	1TFLOPS 以上(倍精度)	1TFLOPS 以上(倍精度)	Unknown
CPU clock	2.2 GHz	2.2 GHz	2.8 GHz
Core	32 +2 (assistant core)	32 +2 (assistant core)	8
SIMD length	256 bit	256 bit	128 bit
Registers (Core spec)	**SIMD : 128** General purpose: 188	**SIMD : 32** General purpose: 188	SIMD : 32 General purpose: 31
Cache	L1I$/L1D$: (4way) separate 64KiB L2$: (24way) shared 24MiB	L1I$/L1D$: (4way) separate 64KiB L2$: (24way) shared 24MiB	L1I$/L1D$: Separate L2$: Shared L3$: Globally shared 8MB
Memory throughput	240GiB/s x2 (R/W)	240GiB/s x2 (R/W)	Unknown

ソフトウェアパイプライニング

「富岳」でも重要となるソフトウェアパイプライニングを中心に評価

- ✓ ループブロック内の命令列をスケジューリングし、演算器を効率的に利用する最適化技術
- ✓ 一回転分の命令を処理できるサイクル数 (II: Initiation Interval)が効果指標



基礎性能コードによる評価

FUJITSU

「京」で効果が高かった多項式のコードを評価

- ✓ 命令セットアーキテクチャ、レジスタ数の変化に応じて最適化の適用状況が変化
- ✓ 特にソフトウェアパイプライニングの効果指標となるIIの数値が大きく変化

9th degree polynomial　expression

```
  Do i = 1, n
        y(i) = c0 + x1(i)*(c1 + x1(i)*
&              (c2 + x1(i)*(c3 + x1(i)*
&              (c4 + x1(i)*(c5 + x1(i)*
&              (c6 + x1(i)*(c7 + x1(i)*
&              (c8 + x1(i)*c9)))))))
  End do
```

最適化情報

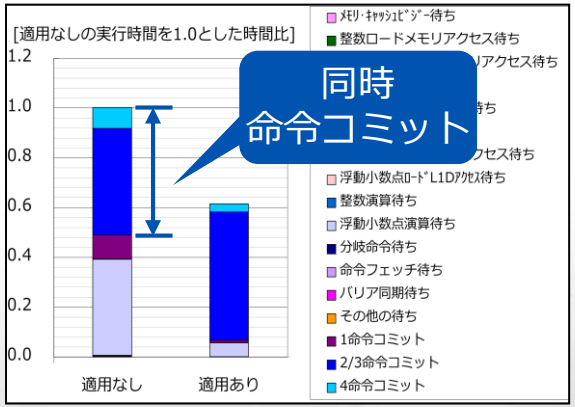
	FX100		X-Gene 2
	128レジスタモード	32レジスタモード	
ループアンローリング	6 展開	3 展開	1 展開
SIMD幅	4	4	2
II	15	22	9

基礎性能コードによる評価

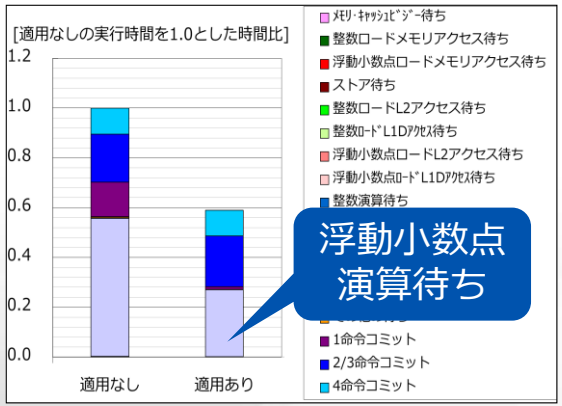
命令のスケジューリングに改善の余地あり

- ✓ 命令セットアーキテクチャに関係なくX-Gene 2でも十分な効果を観測
- ✓ 32レジスタモードでは浮動小数点演算待ちの改善効果が弱い

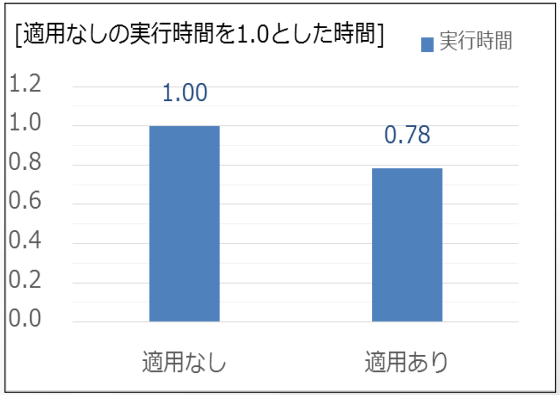
128レジスタモード@FX100



32レジスタモード@FX100



X-Gene 2

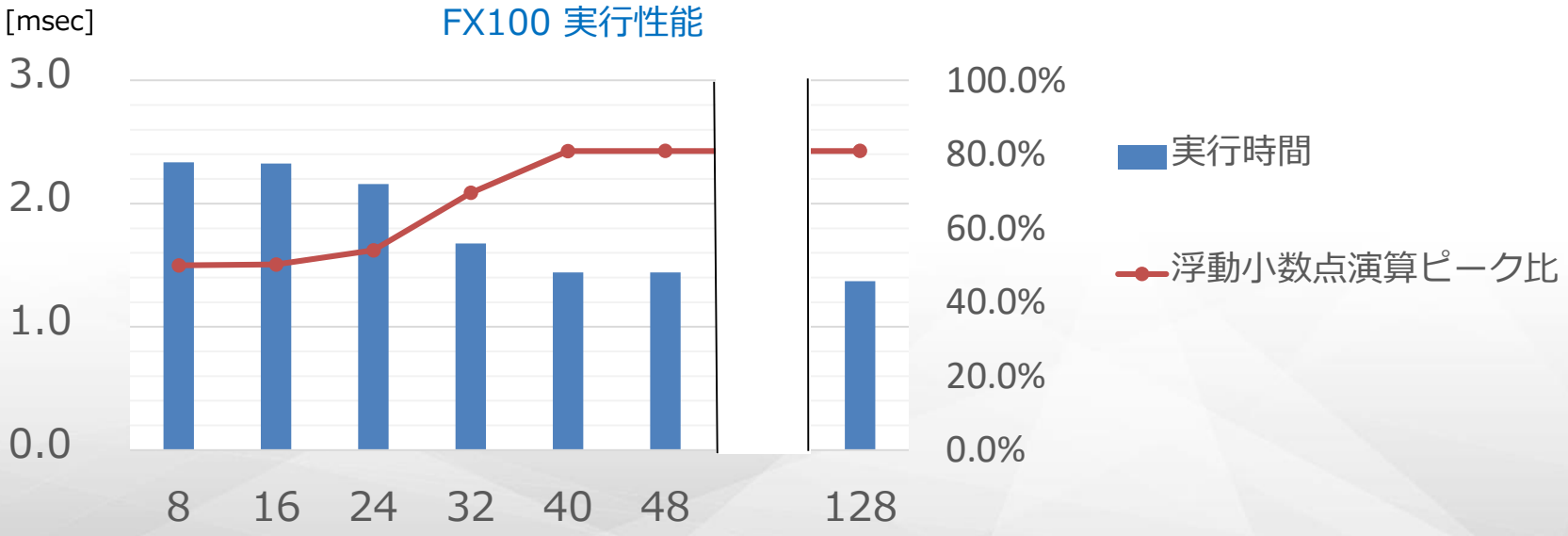


基礎性能コードによる評価

FUJITSU

ソフトウェアパイプライニングで要求されるレジスタ数はコードに依存

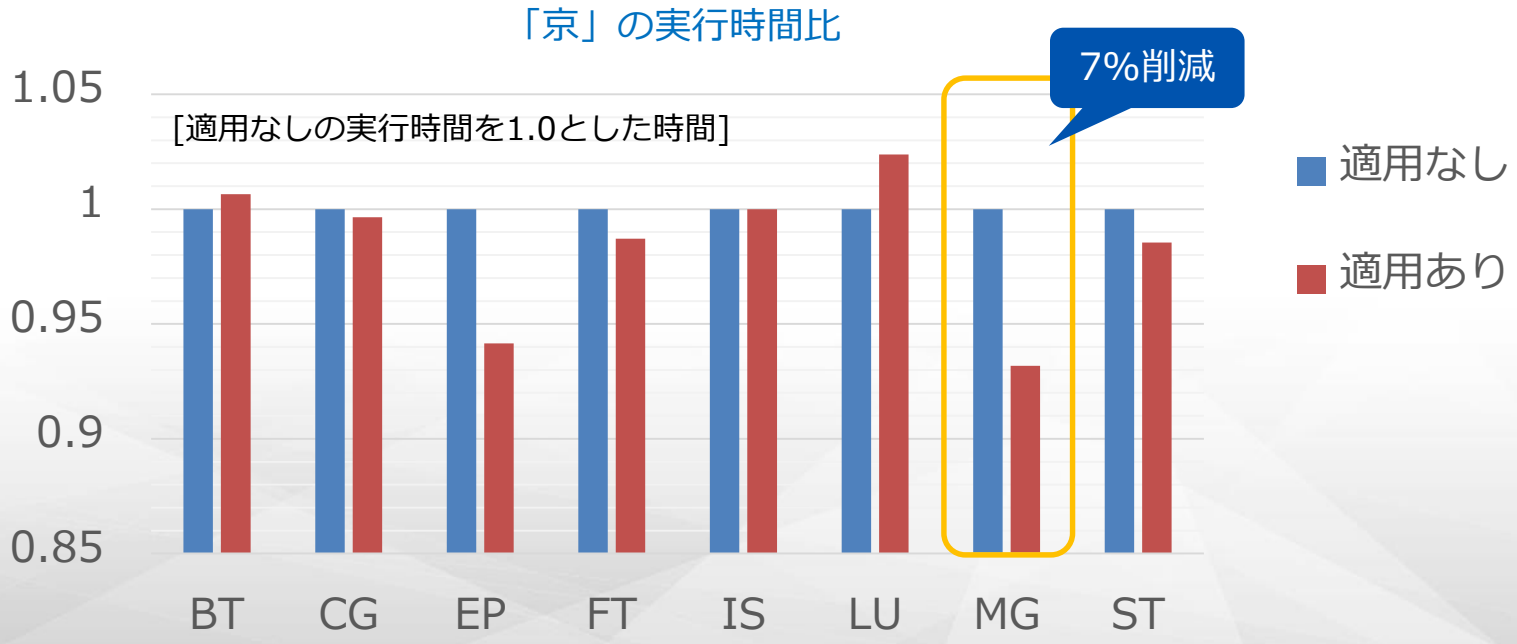
- ✓ 一般的には浮動小数点レジスタ数が多い方が有利
- ✓ 本コードについては、”４０”で十分な性能を観測



ベンチマークによる評価

NPB: Nas Parallel Benchmark class Aによる評価

✓「京」でもっとも効果が高かったMGを評価対象として利用



ベンチマークによる評価

MGベンチマークは命令の並列性が高く評価対象として最適

- ✓ 三次元ポアソン方程式をマルチグリッド法で解くコード
- ✓ 残差計算が高コスト部、命令の並列性が高い演算列

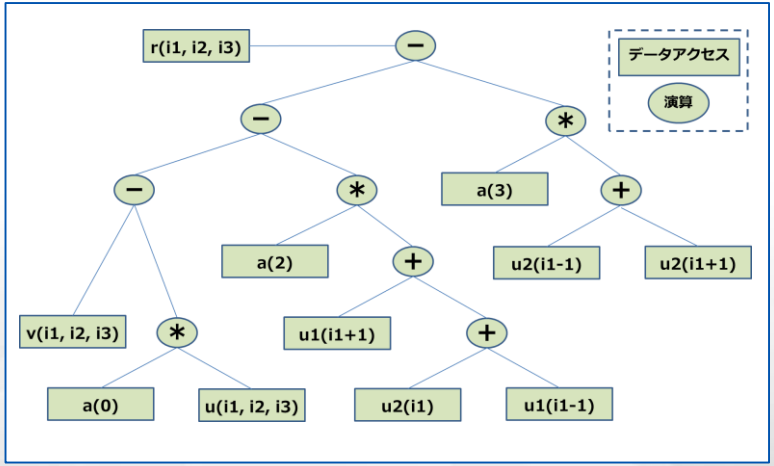
コスト分布

関数 / ループ部	実行コスト割合(%)
resid / 2nd loop	41.1765
zran3 / 3rd loop	23.5294
resid / 1st loop	11.7647

resid 2nd loopのループボディ

```
  r(i1,i2,i3) =
>    v(i1,i2,i3)
>    - a(0) * u(i1,i2,i3)
>    - a(2) * ( u2(i1) + u1(i1-1) + u1(i1+1) )
>    - a(3) * ( u2(i1-1) + u2(i1+1) )
```

演算の依存関係

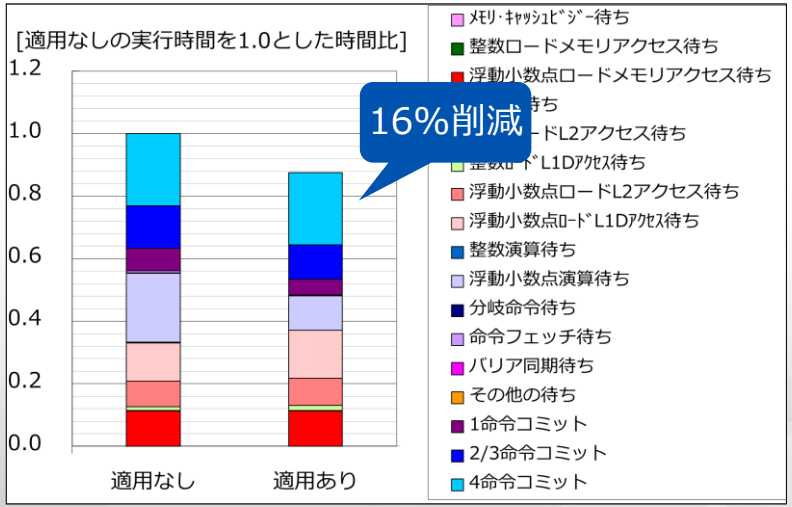


ベンチマークによる評価

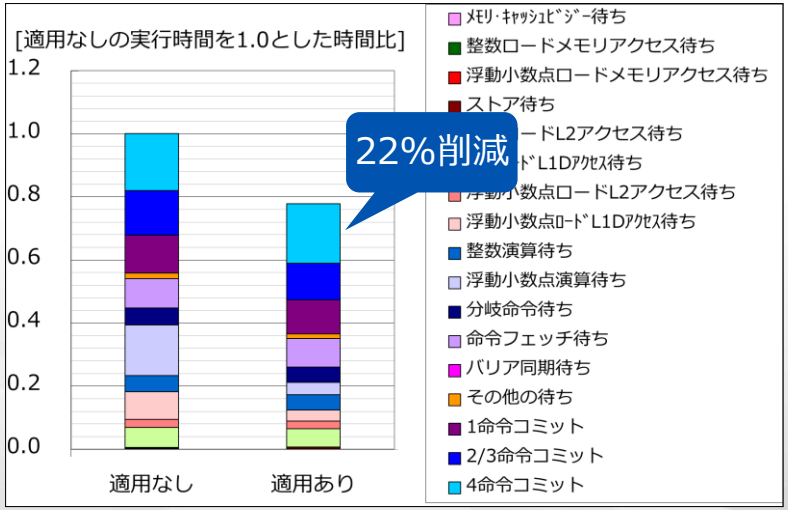
「京」と同等レベルとなるソフトウェアパイプライニングの効果を観測

- ✓ 浮動小数点演算待ちの改善度合いが大きい
- ✓ 全区間で16%、高コスト部で22%の削減効果
- ✓ 全区間のL1Dアクセス待ちの増加はデータ供給待ちが顕在化

全区間（32レジスタモード@FX100）



残差計算2nd loop（32レジスタモード@FX100）

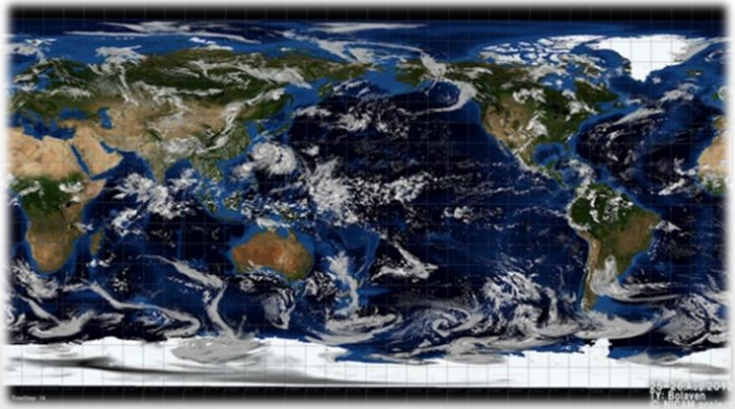


数値シミュレーションによる評価

NICAMを利用した実際のアプリケーションの評価

- ✓ Non-hydrostatic ICosahedral Atmospheric Model（非静力学正20面体格子大気モデル）
- ✓ 「富岳」の重点課題（4）に採用

NICAM



コードの特徴

1. 力学過程：格子上の風速・気温などを解く
   - 構造格子ステンシル計算
   - メモリへの要求が高い
2. 物理過程：雲の相変化など物理現象を解く
   - 計算量が多い
   - 分岐処理が多い

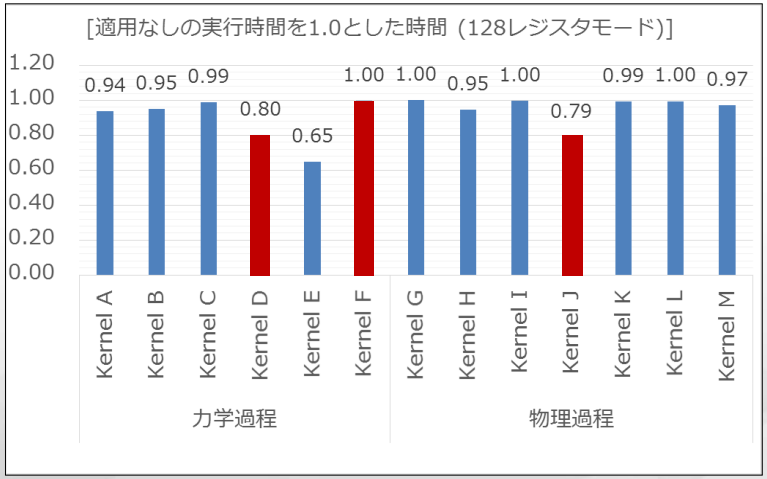
画像引用2021/1/08時点：東京大学:NICAM:非静力学正20面体格子大気モデル ”https://cesd.aori.u-tokyo.ac.jp/nicam/index.html”

数値シミュレーションによる評価

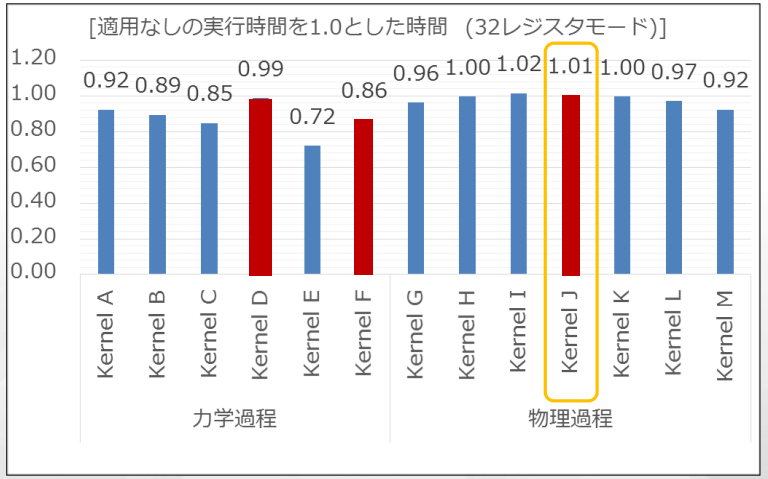
各過程ごとに特徴コードを切り出し細かい単位で性能の変化を評価

- ✓ 最適化の適用有無によるスピードアップ度合いを比較
- ✓ 力学過程 カーネルD, Fと物理過程 カーネルJで効果差
- ✓ カーネル Jについて分析した結果、レジスタ不足により最適化が適用できていないことが判明

実行時間比（128レジスタモード@FX100）



実行時間比（32レジスタモード@FX100）



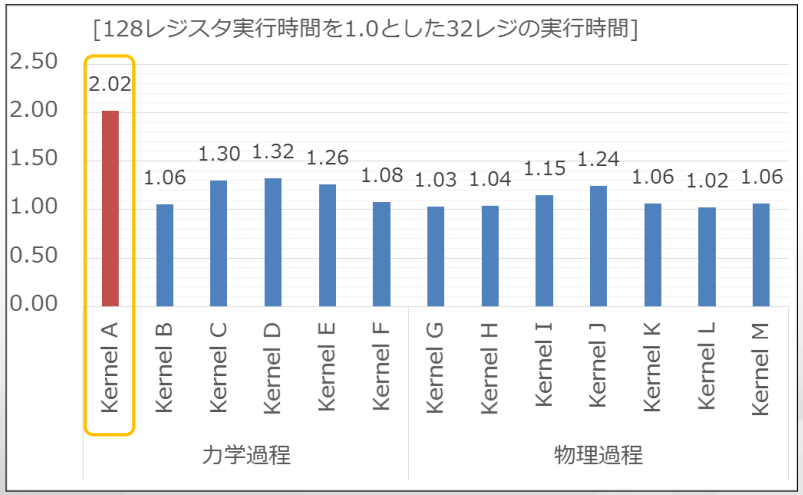
数値シミュレーションによる評価

FUJITSU

性能効果だけでなくレジスタ数の違いによる実行時間の変化も評価

- ✓ 128レジスタモードと32レジスタモードの実行時間比
- ✓ 力学過程 Kernel Aに大きな差あり
- ✓ レジスタ不足によるSPILL/FILLが原因

実行時間比（FX100）



カーネルA 命令情報

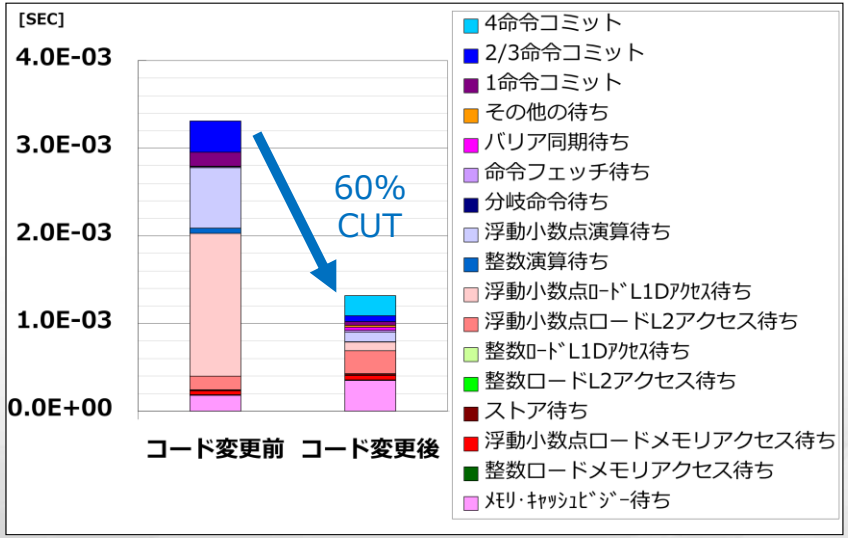
	浮動小数点演算命令数	整数演算命令数	メモリロード/ストア命令数
128レジスタ	1.83E+07	6.34E+06	**1.66E+07**
32レジスタ	1.85E+07	7.86E+06	**3.67E+07**

数値シミュレーションによる評価

手修正のコード変形により開発が必要な最適化を抽出

- ✓ ループ分割、データ依存期間の短縮、ループ融合の最適化を適用することで性能が向上
- ✓ 実行時間を大幅に改善（約60%短縮）

コード変更後の実行時間比



コデザインから
コンパイラ開発へフィードバック

カーネルA 命令情報

	浮動小数点演算命令数	整数演算命令数	メモリロード/ストア命令数
128レジスタ	1.83E+07	6.34E+06	**1.66E+07**
32レジスタ	1.85E+07	6.86E+06	**3.67E+07**
			0.98E+07

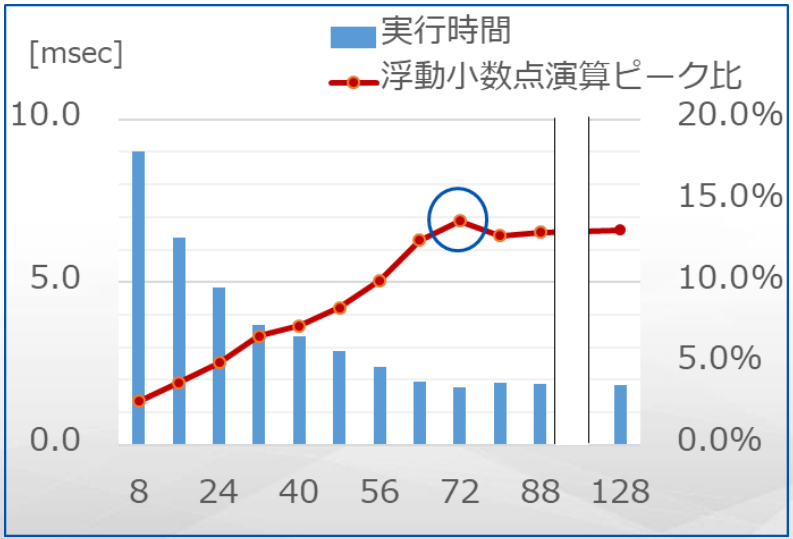
数値シミュレーションによる評価

FUJITSU

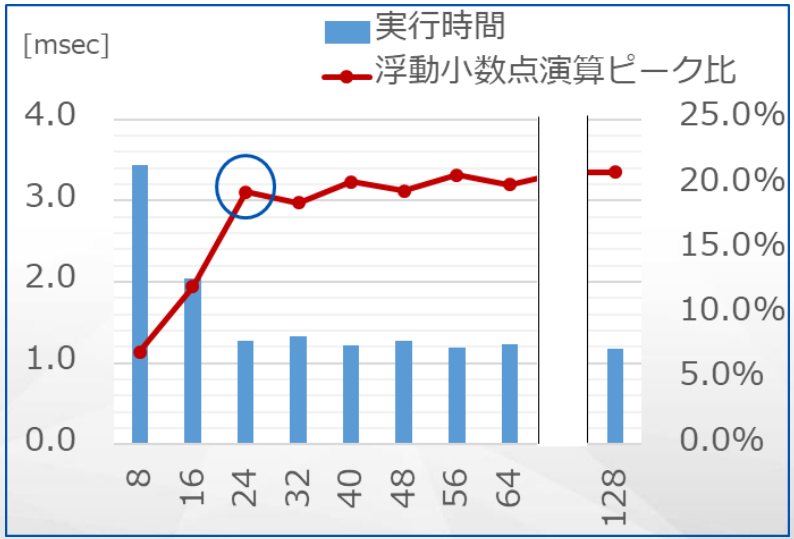
最適化を追加することで要求されるレジスタ数を改善し高速化を実現

- ✓ コデザインの結果をコンパイラ開発へフィードバックすること最適化を強化
- ✓ 追加された最適化により要求されるレジスタ数が改善

初回評価（32レジスタモード@FX100）



最適化追加後（32レジスタモード@FX100）

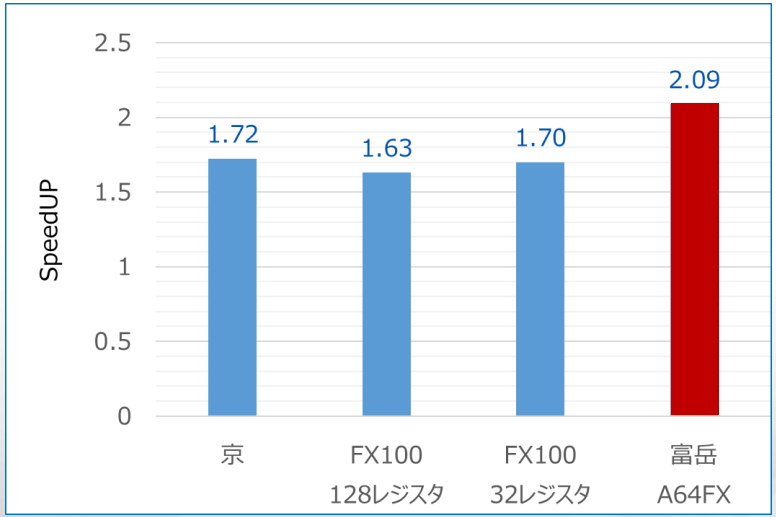


コデザインの成果

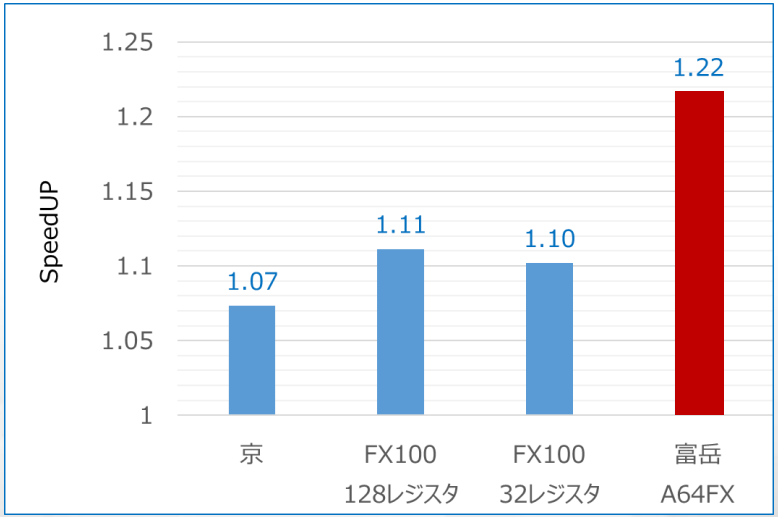
評価環境における予測を上回る効果をA64FXで観測

- ✓ コデザインによる最適化開発によりArmv8-Aアーキテクチャに対応ができている
- ✓ 最適化の効果が「京」を上回っている

基礎性能コードの最適化効果



NPB MGの最適化効果

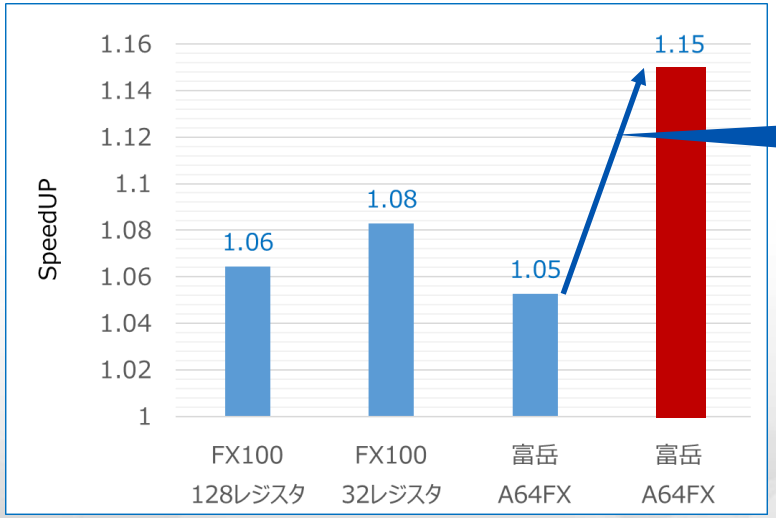


コデザインの成果

評価環境における予測を上回る効果をA64FXで観測

- ✓ コデザインによる最適化開発によりArmv8-Aアーキテクチャに対応ができている
- ✓ 最適化を追加することでソフトウェアパイプライニングの効果を引き出している

NICAMの最適化効果



コデザインのフィードバックにより
最適化を強化したことで性能向上

ソフトウェア
パイプライニング
＋
ループ
分割
＋
ストリップ
マイニング

コデザインの成果

FUJITSU

A64FXの完成前に「富岳」向けコンパイラ技術で論文賞を受賞

論文

省レジスタアーキテクチャ向けソフトウェアパイプライニングの評価

千葉 修一（富士通）
青木 正樹（富士通）
鎌塚 俊　（富士通）
松井 雅人（メトロ）
八代 尚（理化学研究所）

FIT2018 船井ベストペーパー賞

- ✓情報処理学会と電子情報通信学会情報・システムソサイエティ合同の会議「情報科学技術フォーラム（FIT）」において、その開催趣旨に賛同頂いた船井情報科学振興財団から贈呈される賞。
- ✓選奨セッションの中から選ばれたFIT論文賞候補論文の中から3件の受賞論文を決める。

引用2021/1/08時点：情報処理学会: FIT船井ベストペーパー賞「 https://www.ipsj.or.jp/award/funai_best-paper.html 」

実用化・商品化・事業化

FUJITSU

事業化に向けては商品化までの「品質」が重要

- ✓ コデザインの成果は「実用化」と一部の「商品化」が該当
- ✓ 企業として事業継続するためには「事業化」が必要

実用化

- ✓ 最適化技術の研究／開発を行い、効果を確認
- ✓ 基本的な品質の確保
- ✓ 社会的利用に向けてコンパイラを提供する

商品化

- ✓ 互換性や網羅的テスト品質を確保しパッケージング
- ✓ 富士通が定める品質基準に基づく検査をクリア
- ✓ 利用を望むすべての人が利用可能になる

事業化

- ✓ 富士通としてビジネスモデルを成立
- ✓ 後継製品の研究／開発を実施する

品質 → 4P/4C

品質評価

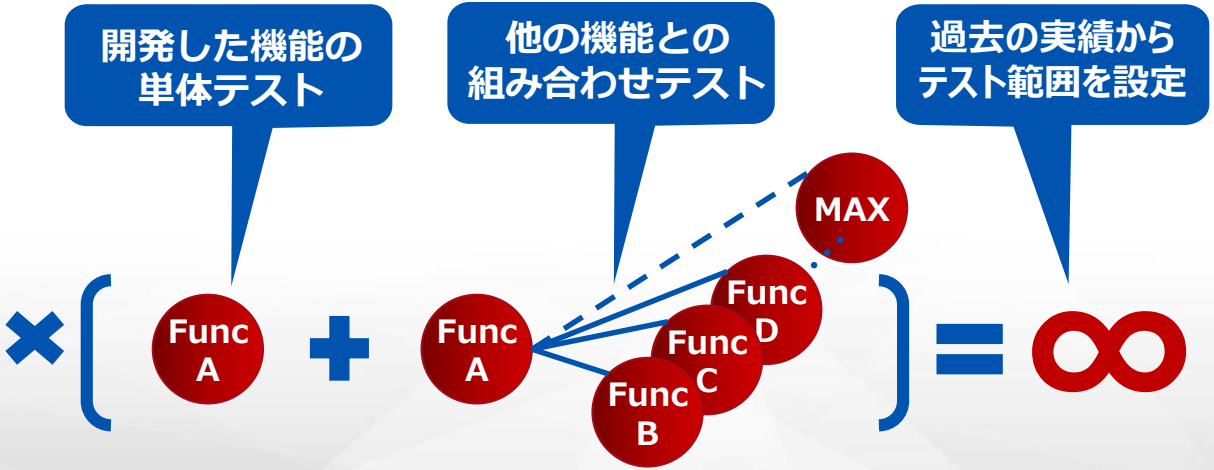
開発した最適化機能ごとに翻訳性能・実行性能を検証

- ✓ プログラミング言語規格が更新されるごとに拡充し続ける基本テストセットが中心
- ✓ 最適化の組み合わせテストをすべて実施するとテスト量は無限大に近い

基本テストセット

Fortran	約 165,000 本
C	約 32,000本
C++	約 15,000本

＋
ベンチマーク
OSSアプリケーション
「富岳」重点課題など



基本テストセットの１ラウンドの走行に約２日

性能保証

FUJITSU

ベンチマークなどの性能値から実アプリケーションの影響を判断

A) 実行時間の劣化度合いは同じになっているが、体感時間が違うケース

B) 大きな性能変化が見られるが、体感時間がわずかなケース

SPEC CPU2017	最適化A 無効	最適化A 有効	実行時間比
600.perlbench_s	0:00:50	0:00:51	1.02
602.gcc_s	0:03:54	0:03:46	0.97
605.mcf_s	0:00:03	0:00:03	1.00
620.omnetpp_s	0:01:32	0:01:32	1.00
625.x264_s	0:00:37	0:00:36	0.97
657.xz_s	0:00:12	0:00:12	1.00
631.deepsjeng_s	0:00:06	0:00:05	0.83
619.lbm_s	0:00:03	0:00:03	1.00
638.imagick_s	0:01:02	0:00:58	0.94
644.nab_s	0:00:07	0:00:08	1.14

ケースごと判定

ケースA

ケースB

性能変化の影響

FUJITSU

テスト時の数字だけでは判断できない利用者への影響

- ✓ 同じ10%の性能低下でも利用シーンごとに影響が異なる、また利用者の価値観も影響
- ✓ 最適化機能の利用頻度、適用範囲、利用者などを踏まえ経験則に基づき許容値を判断

<利用シーンA> 実行時間が10分、いつもすぐに受け取る

ジョブ実行

wait

＋1分

待ち時間が数分延びる程度

<利用シーンB> 実行時間が12時間、夜に実行して翌朝に受け取る

wait

＋72分

翌朝、確認してもジョブが終わってない

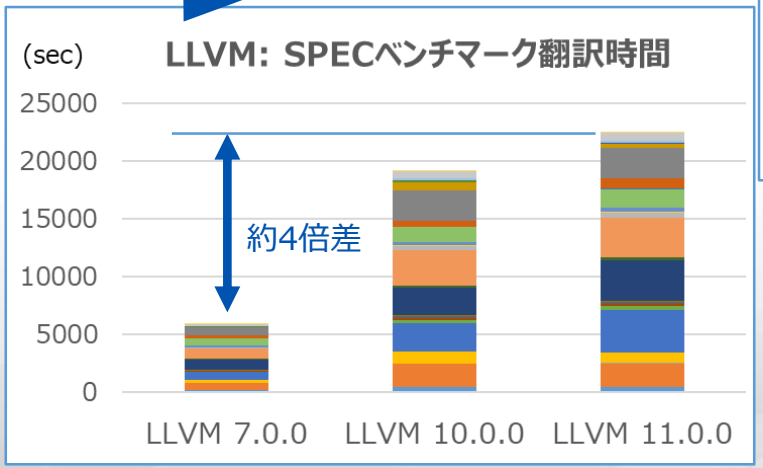
Clangモードの提供に向けて

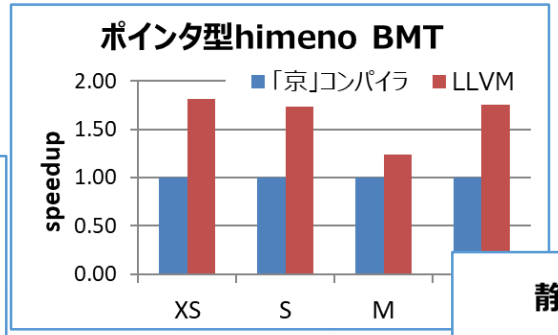
FUJITSU

「富岳」で採用したClangモードは多くの課題をクリアして搭載

- ✓ OSSアプリケーションへのサポート力強化に向けて追加したC/C++向け翻訳モード
- ✓ OSSコンパイラであるLLVMのエンジンを採用、基本品質が課題

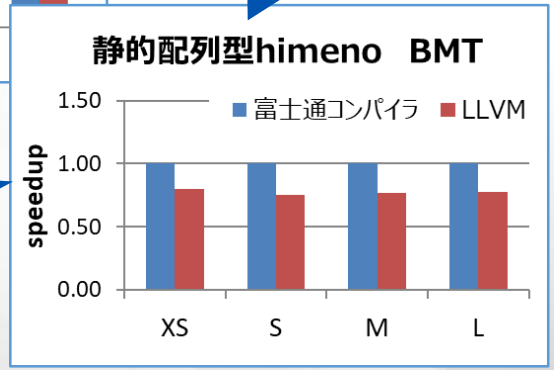
「富岳」は7.0.0を採用





LLVMの性能改善が必要

コンパイラの特徴差

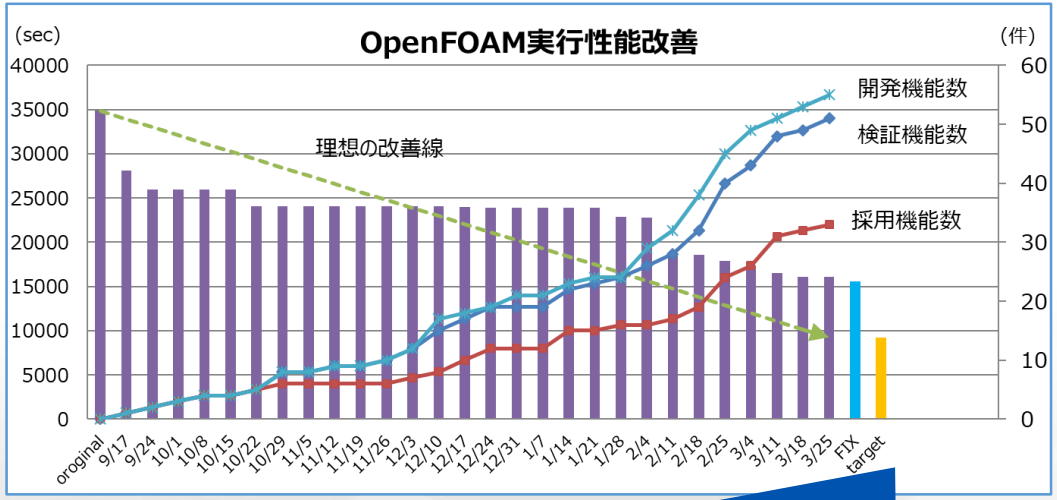


生みの苦しみ

FUJITSU

開発された最適化は全てが世に出ていくわけではない

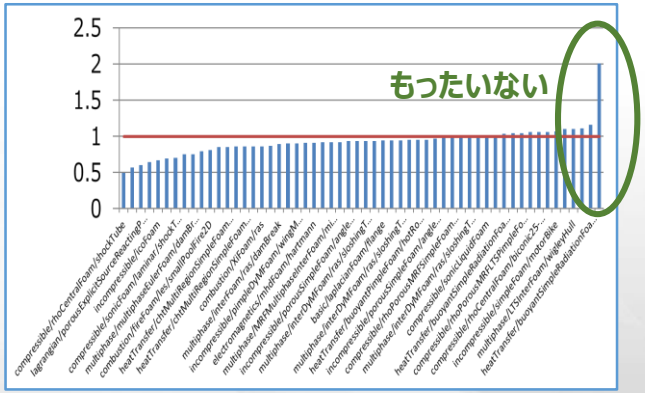
- ✓ 「京」の時代、流体解析OpenFOAM向けの最適化を開発したデータ
- ✓ 採用される機能は約半分



商品化されるのは開発した機能の約半分

不採用例
1本だけ速くてもNG

チュートリアル56本の性能UP率



利用者との共存

FUJITSU

利用者とのギャップを埋めて「富岳」の本格稼働へ

- ✓ 機能提供時の品質をレベル分けするなどサービス性向上に向けた品質基準の見直し
- ✓ 開発戦略や開発機能をオープン化し、利用者とのディスカッションを持つ

利用者の要望

＜サービス性重視＞

- ✓ 障害修正など提供のスピードUP
- ✓ 細かい最適化オプションの提供
- ✓ シミュレーションの結果は検算するため、計算結果が変化してもいいので、アグレッシブな最適化が欲しい

開発元

＜品質重視＞

- ✓ 社会基盤ソフトと同等の富士通品質
- ✓ マニュアルで説明できない機能は提供できない
- ✓ 実行性能に効果があっても翻訳時間が大きく増大するものは提供できない

「富岳」の成果

FUJITSU

４つのランキングで２期連続世界１位を獲得

TOP500 The List. HPCG HPL-AI GRAPH 500

スーパーコンピュータ「富岳」

© RIKEN

スーパーコンピュータ「京」

TOP500 No.1(2011) HPCG No.1(~2017) GRAPH 500 No.1(~2019)

AWARDS Gordon Bell Prize Finalist(2016)

TOP500 The List. 効率No.1（91%, 2008）

FX1

© RIKEN

PRIMEHPC FX10

2012

PRIMEHPC FX100

2015

arm

PRIMEHPC FX1000/FX700

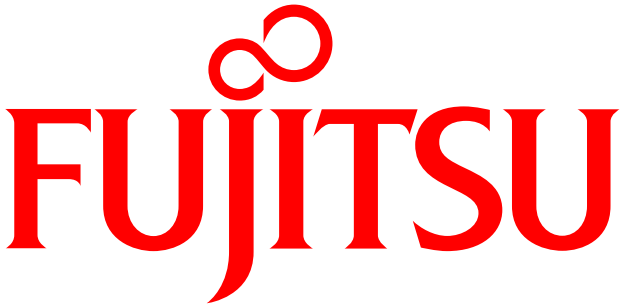
進化を続けるコンパイラ技術

ストレスフリーな利用環境へ

まとめ

FUJITSU

- ✓「京」に続き「富岳」でもコンパイラの技術が性能を支えている
- ✓「富岳」ではコデザインの成果がしっかりと取りこまれている
- ✓今後も利用者と一緒になって成長する必要がある



shaping tomorrow with you