FUJITSU
shaping tomorrow with you

SS研 科学技術計算分科会 2015年度会合

アプリケーション性能による PRIMEHPC FX100の評価

2015年10月28日

富士通株式会社
次世代テクニカルコンピューティング開発本部
三吉 郁夫

Copyright 2015 FUJITSU LIMITED

もくじ

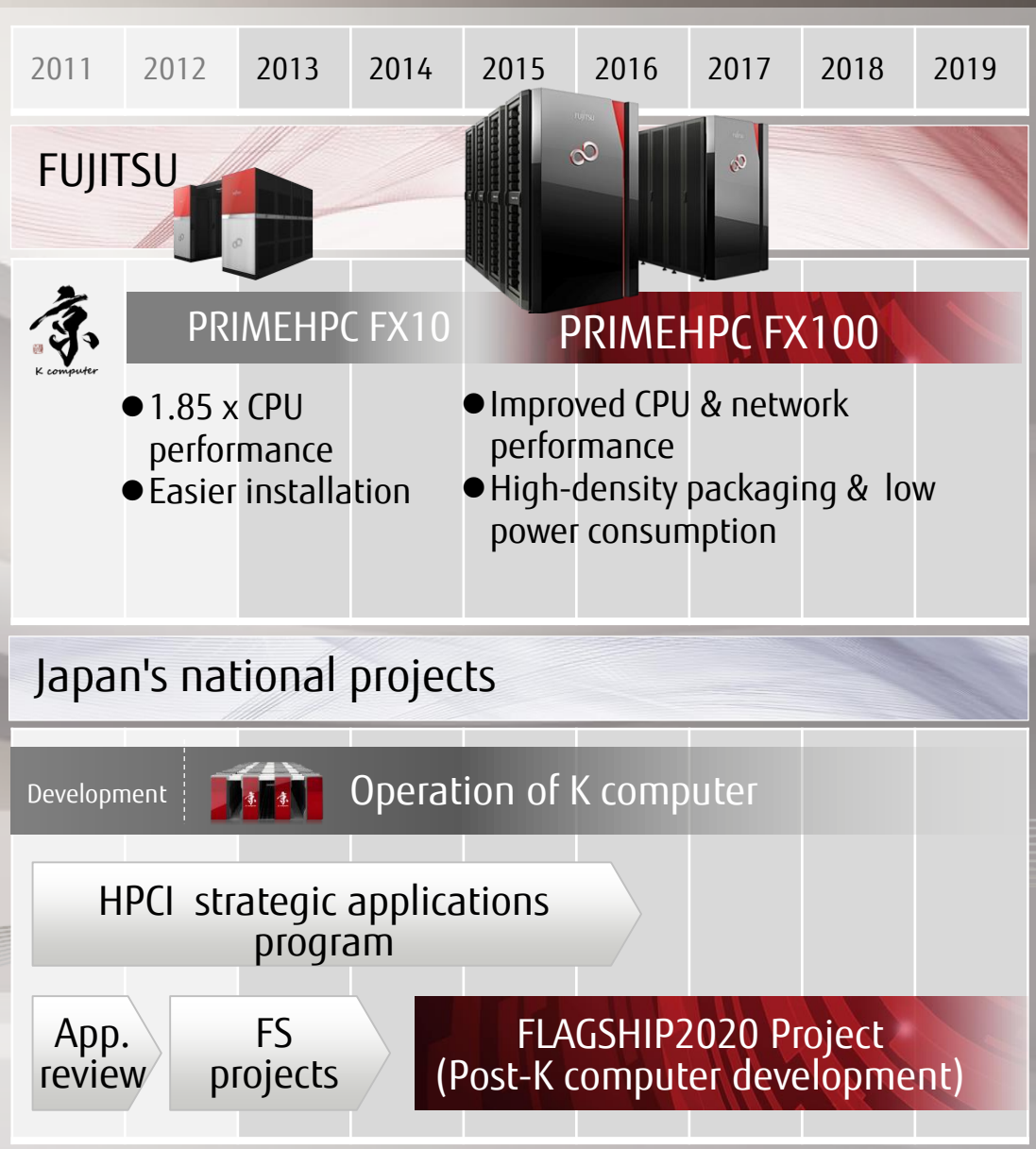
FUJITSU

- PRIMEHPC FX100の概要
- PRIMEHPC FX100の使い方
- アプリケーションカーネルの性能
- ミニアプリによる評価
- SS研ポストペタアプリ性能WG報告より
- C++アプリの性能
- まとめ

Copyright 2015 FUJITSU LIMITED

ロードマップ – エクサスケールへ

FUJITSU



「京」 ：稼働中
PRIMEHPC FX10 : 稼働中

科学/技術分野の多数のアプリが開発され稼動している

PRIMEHPC FX100 : 出荷中

「京」のアーキテクチャコンセプトを引き継ぎ、CPUとインターコネクトの性能を向上

エクサスケールに向けて

FLAGSHIP2020プロジェクトにて、ポスト「京」コンピュータの基本設計に参画

Copyright 2015 FUJITSU LIMITED

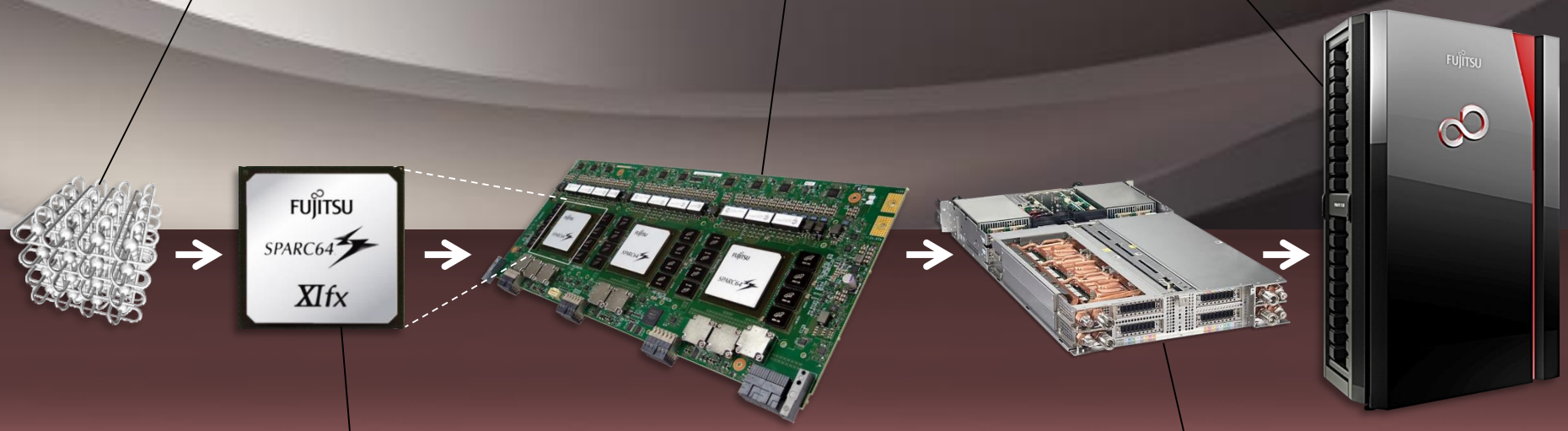
PRIMEHPC FX100の構成と機能

FUJITSU

Tofu Interconnect 2
- 12.5 GB/s × 2(in/out)/link
- 10 links/node
- Optical technology

CPU Memory Board
- Three CPUs
- 3 x 8 Micron's HMCs
- 8 opt modules, for inter-chassis connections

Cabinet
- Up to 216 nodes/cabinet High-density
- 100% water cooled with EXCU (option)



Fujitsu designed SPARC64 XIfx
- 1TF~(DP)/2TF~(SP)
- 32 + 2 core CPU
- HPC-ACE2 support
- Tofu2 integrated

Chassis
- 1 CPU/1 node
- 12 nodes/2U Chassis
- Water cooled

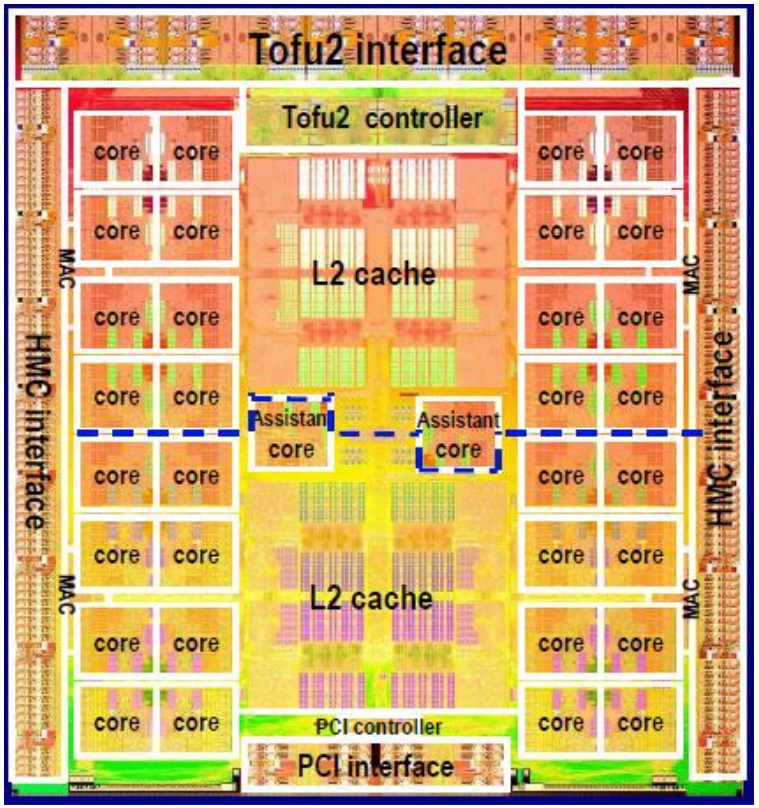
Copyright 2015 FUJITSU LIMITED

独自開発CPU SPARC64™ XIfx

FUJITSU

- 256ビット幅SIMD
  - 単精度倍幅演算、整数演算にも対応
- L1Dキャッシュ容量/ウェイ数を倍増
- アウトオブオーダ資源、分岐予測機構などを強化
- アシスタントコア×2
  - OSノイズ低減、通信と演算のオーバラップ
- HMCサポート
- Tofuインターコネクト2内蔵

アーキテクチャ	SPARC V9 + HPC-ACE2
コア数	32 コア + 2 アシスタントコア
演算器	FMA x 2 (256ビット幅SIMD)
キャッシュ	L1命令: 64 KB / コア L1データ: 64 KB / コア L2: 24 MB / ノード
メモリ容量	32 GB / ノード
メモリバンド幅	240 GB/秒 (read) + 240 GB/秒 (write)



Copyright 2015 FUJITSU LIMITED

PRIMEHPC FX100の使い方

- FX10からの機能強化ポイント
- 倍精度4way/単精度8way SIMD
- 32スレッド実行
- XFILL&セクタキャッシュ機能
- 通信と演算のオーバラップ実行
- COARRAY機能

Copyright 2015 FUJITSU LIMITED

FX10からの機能強化ポイント

FUJITSU

項目	FX10	FX100
演算コア	16コア	32コア
アシスタントコア	なし	2コア
SIMD幅	128ビット	256ビット
SIMD命令	浮動小数点演算、 連続ロード/ストア	左記に加え、整数演算、 ストライド&間接ロード/ストア
その他の拡張命令	逆数近似、平方根逆数近似、 三角関数補助 など	左記に加え、指数関数補助、 丸め処理 など
L1Dキャッシュ/コア	32KB、2ウェイ	64KB、4ウェイ
L2キャッシュ/ノード	12MB	24MB
メモリバンド幅	85GB/秒	480GB/秒
XFILL機能	あり	あり
セクタキャッシュ機能	あり	あり
コア間HWバリア	あり	あり
Tofuバリア	あり	あり

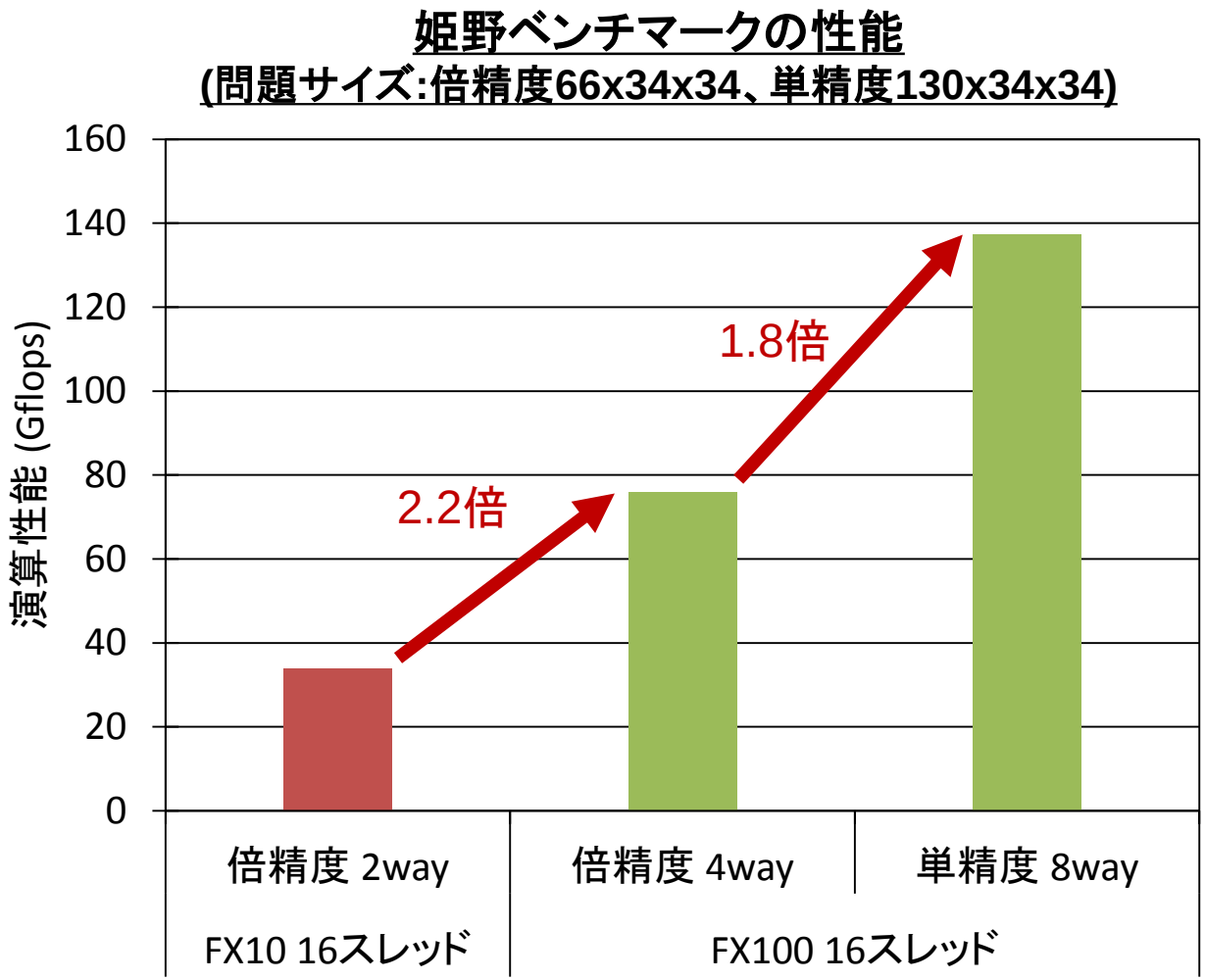
- 以降のスライドでは、主なポイントを姫野ベンチマークで説明

Copyright 2015 FUJITSU LIMITED

倍精度4way/単精度8way SIMD

FUJITSU

- FX10に対し、倍精度で2倍、単精度で4倍の理論演算性能
  - 単純ケースは、再翻訳するだけで性能向上

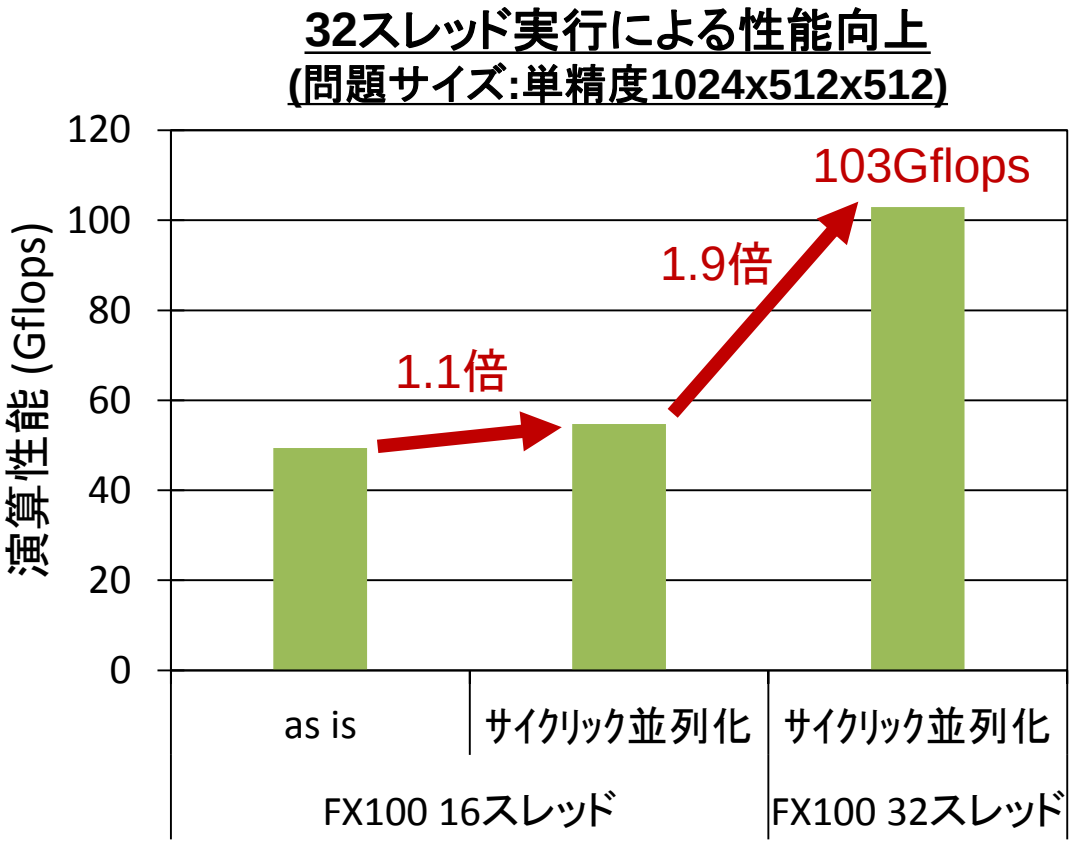


Copyright 2015 FUJITSU LIMITED

32スレッド実行

- L2キャッシュおよびメモリを共有し、32スレッド実行が可能
  - 実行コマンド例: numactl –i all lpgparm –l demand ... himenoBMTxp
- サイクリック並列化により、スレッド間でデータを再利用し性能向上
  - コーディング例:

```
!OCL PARALLEL_CYCLIC
do k=2,kmax-1
  do j=2,jmax-1
    do i=2,imax-1
      s0=...p(i,j,k)... &
            ...p(i,j,k+1)... &
            ...p(i,j,k-1)...
      ...
    enddo
  enddo
enddo
```



Copyright 2015 FUJITSU LIMITED

XFILL&セクタキャッシュ機能

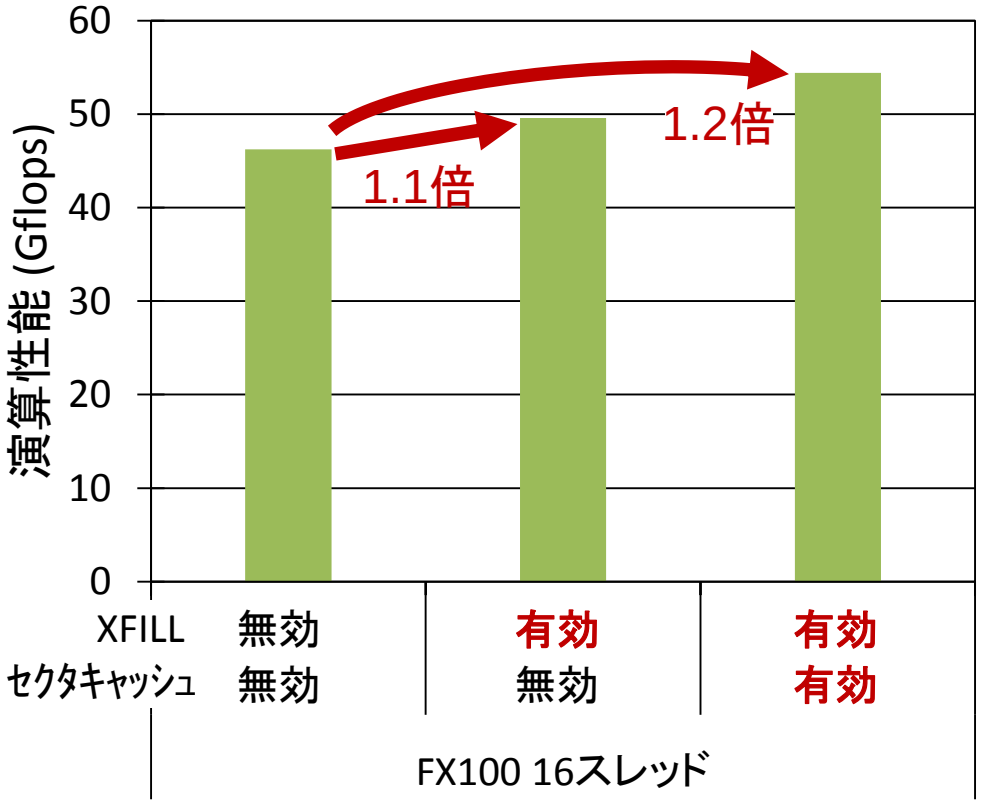
FUJITSU

- XFILL機能でメモリバンド幅を節約可能
  - 翻訳コマンド例: frtpx ... -KXFILL ... himenoBMTxp.f90
- セクタキャッシュ機能で再利用可能データをキャッシュに残すことで性能向上
  - コーディング例:

```
!OCL CACHE_SECTOR_SIZE(18,6)
!OCL CACHE_SUBSECTOR_ASSIGN(…)
do k=2,kmax-1
  do j=2,jmax-1
    do i=2,imax-1
      s0=...p(i,j,k)... &
           ...p(i,j+1,k)... &
           ...p(i,j-1,k)...
      …
      wrk2(i,j,k)=…
    enddo
  enddo
enddo
!OCL END_CACHE_SECTOR_SIZE
!OCL END_CACHE_SUBSECTOR
```

P以外の配列を指定

XFILL&セクタキャッシュ機能の効果
(問題サイズ:単精度4098x130x130、jループで並列化)



Copyright 2015 FUJITSU LIMITED

通信と演算のオーバラップ実行

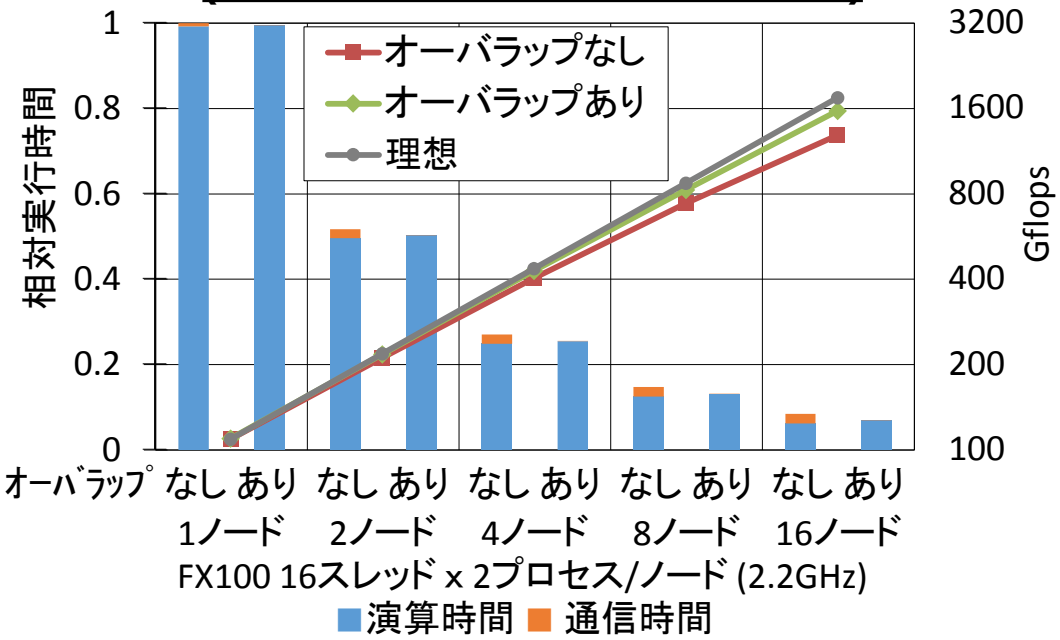
FUJITSU

- 通信データに依存しない演算処理をMPI_Isend/Irecv～MPI_Waitall間に置き、アシスタントコア利用を実行時に指示
  - 実行コマンド例:
    mpiexec ... --mca opal_progress_thread_mode 3 ... himenoBMTxpr
  - コーディング例:

```
do k=2,kmax-1,kmax-3
  do j=2,jmax-1; do i=2,imax-1
    ...
  enddo; enddo
enddo
call MPI_Irecv(...); call MPI_Irecv(...)
call MPI_Isend(...); call MPI_Isend(...)
do k=3,kmax-2
  do j=2,jmax-1; do i=2,imax-1
    ...
  enddo; enddo
enddo
call MPI_Waitall(...)
```

通信データに依存しない演算処理

通信/演算オーバラップの性能効果
(問題サイズ:単精度1024x512x512)



通信/演算オーバラップによって性能スケーラビリティを改善

Copyright 2015 FUJITSU LIMITED

COARRAY機能 (Fortran 2008規格より)

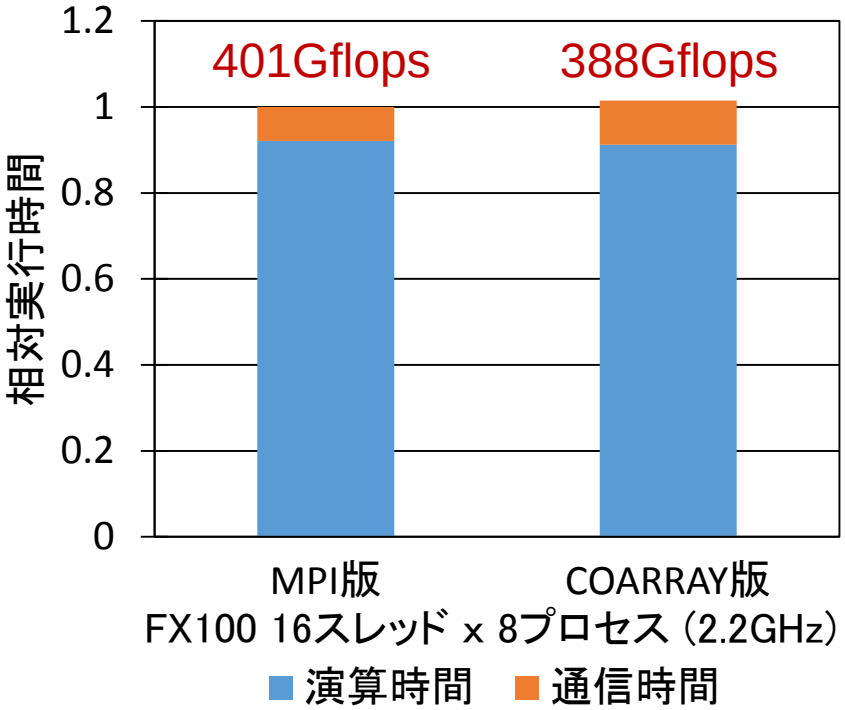
FUJITSU

- COARRAY(共配列)…プロセス間で定義/参照が可能な配列
  - 翻訳コマンド例: frtpx ... -Ncoarray ... co_himeno.f90
  - コーディング例:

```
do k=2,kmax-1
  do j=2,jmax-1
    do i=2,imax-1
      ...
    enddo
  enddo
enddo
...
sync all
if (myx < ndx) then
  p(1,2:jmax-1,2:kmax-1)[myx+1,myy,myz] &
    = p(imax-1,2:jmax-1,2:kmax-1)
endif
...
call co_sum(gosa)
```

他プロセスのデータを更新

MPI版との性能比較
(問題サイズ:単精度1024x512x512、4ノード)



COARRAY機能によって
プロセス並列も容易かつ効率的に

(参考: https://github.com/srinathv/Coding-Examples/tree/master/nagle1/Fortran-III/02-Himeno)

Copyright 2015 FUJITSU LIMITED

アプリケーションカーネルの性能

- 測定環境
- NAS Parallel Benchmarks

Copyright 2015 FUJITSU LIMITED

測定環境

FUJITSU

				PRIMEHPC FX10	PRIMEHPC FX100	PRIMERGY RX2530 M1
CPU	モデル			SPARC64 IXfx	SPARC64 XIfx	Xeon E5-2698v3 (Haswell)
CPU	動作周波数(GHz)			1.848	**1.975**	2.3※1
CPU	CPUあたりコア数			16	**32**	16※2
CPU	キャッシュメモリ	容量	L1I(B/コア)	32K	64K	32K
CPU	キャッシュメモリ	容量	L1D(B/コア)	32K	64K	32K
CPU	キャッシュメモリ	容量	L2(B/コア)	—	—	256K
CPU	キャッシュメモリ	容量	LL(B/CPU)	12M	24M	40M
CPU	キャッシュメモリ	連想度	L1I,D(ウェイ)	2	4	8
CPU	理論ピーク性能(Gflops/CPU)		倍精度	236.5	1,011	589
CPU	理論ピーク性能(Gflops/CPU)		単精度		2,022	1,178
CPU	主記憶-CPU間帯域幅(GB/秒/CPU)			85	480	68
CPU	B/F比(B/flop)		L1D	4		
CPU	B/F比(B/flop)		主記憶	0.36	0.47	0.12
ノード	ノードあたりCPU数			1		2
ノード	ノードあたりコア数			16	32	
ノード	理論ピーク性能(Gflops/ノード)		倍精度	236.5	**1,011**	1,178
ノード	理論ピーク性能(Gflops/ノード)		単精度		2,022	2,355
ノード	主記憶容量(GB/ノード)			64	32	128
ノード	主記憶-CPU間帯域(GB/秒/ノード)			85	**480**	137
ノード	ノード間通信帯域(GB/秒/ノード/双方向)			40	100	—
ソフト	OS名称と版数			XTC OS 1.3.5	XTC OS 2.1.1	RHEL 6.5
ソフト	コンパイラ版数			TC Suite V2.0L20		Intel Fortran/C/C++ 15.0.3
ソフト	MPIライブラリ版数			TC Suite V2.0L20		Intel MPI 5.0.3

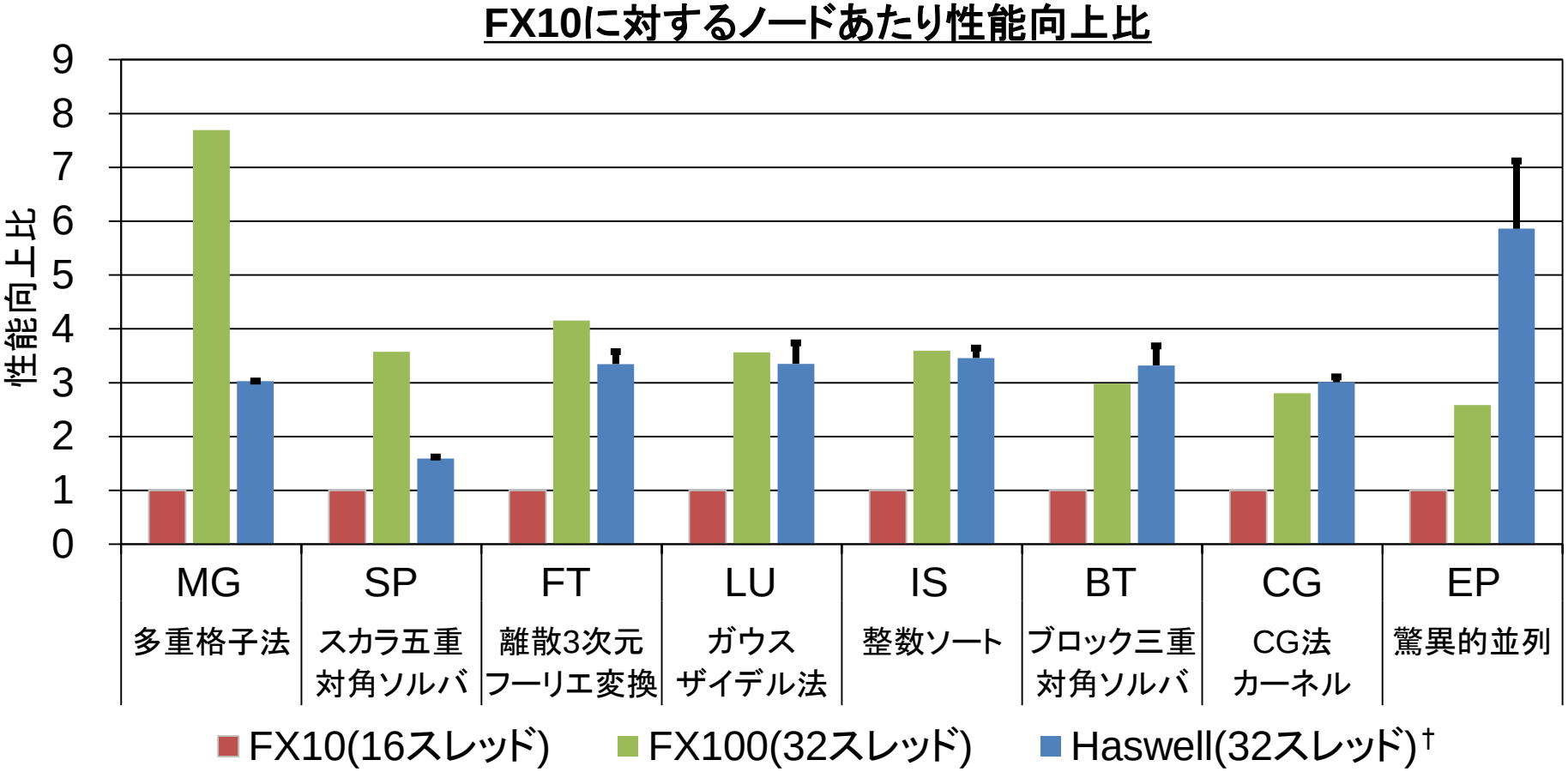
※1: Turbo Boost時最大3.6GHz、特記する場合を除き1.9GHz固定　※2: Hyper-Threading無効

Copyright 2015 FUJITSU LIMITED

NAS Parallel Benchmarksによる評価

FUJITSU

■ OpenMP版を用いてノードあたり演算性能を評価



† エラーバーはCPU周波数を1.9GHzに固定しない場合の性能を表す

FX100は、FX10比平均3.9倍、Haswell比平均1.3倍のノードあたり演算性能

使用コード: NAS Parallel Benchmarks Ver. 3.3.1 OpenMP版 クラスC

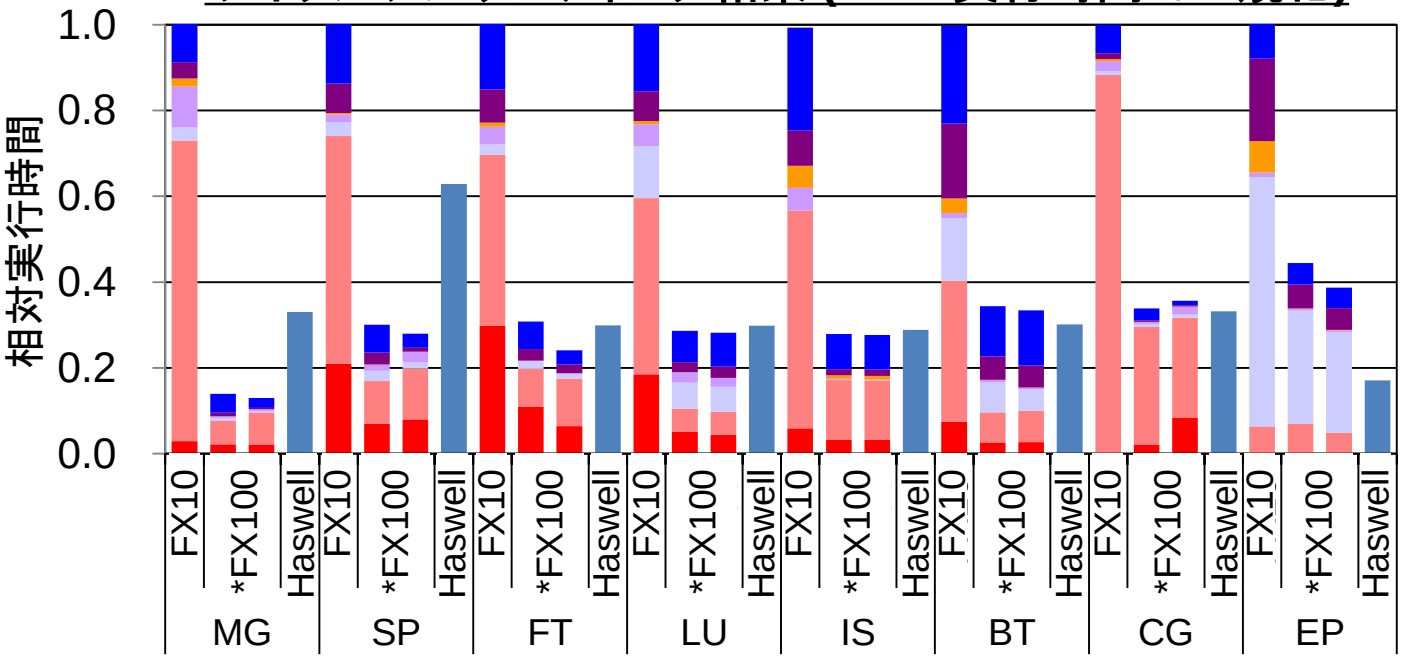
Copyright 2015 FUJITSU LIMITED

NAS Parallel Benchmarksの性能分析 (1/2)

FUJITSU

■ PAデータによる傾向分析

サイクルアカウンティング結果 (FX10実行時間で正規化)



* FX100の左側は2way SIMD、右側は4way SIMD

SIMD命令率、L1D$/L2$ミス率およびメモリスループット (FX100)

	MG	SP	FT	LU	IS	BT	CG	EP
SIMD命令率(%)	77	56	59	13	0	6	74	**0**
L1D$ミス率(%)	1.5	3.2	4.5	3.6	5.0	1.7	32.9	**0.8**
L2$ミス率(%)	1.0	1.8	0.6	0.8	0.5	0.3	1.6	0.5
メモリスループット(GB/秒)	**236**	**243**	**132**	104	44	37	50	14

メモリ律速に強いが、on L1D$かつSIMD化難の場合に課題あり

Copyright 2015 FUJITSU LIMITED

NAS Parallel Benchmarksの性能分析 (2/2)

FUJITSU

■ EPベンチマークの性能改善課題

EPベンチマークの高コストループ

```
do 140 i = 1, nk
  x1 = 2.d0 * x(2*i-1) - 1.d0
  x2 = 2.d0 * x(2*i) - 1.d0
  t1 = x1 ** 2 + x2 ** 2
  if (t1 .le. 1.d0) then
    t2   = sqrt(-2.d0 * log(t1) / t1)
    t3   = (x1 * t2)
    t4   = (x2 * t2)
    l    = max(abs(t3), abs(t4))
    qq(l) = qq(l) + 1.d0
    sx   = sx + t3
    sy   = sy + t4
  endif
continue
```

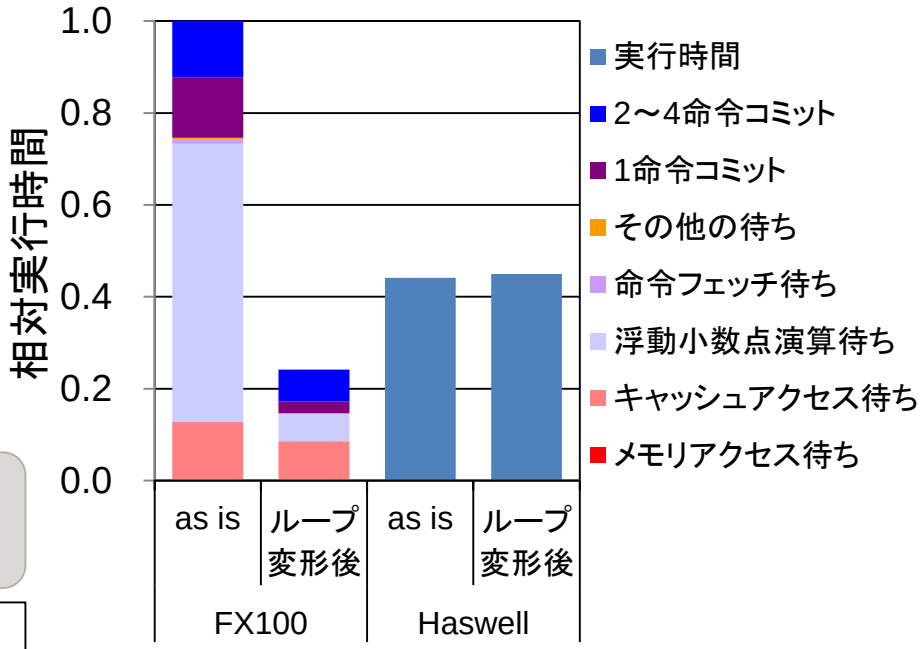
ifブロックの中で数え上げ処理

SIMD化促進のためのループ変形

```
    ...
    llc = llc + 1
    ll(llc) = l
    ...
  endif
continue
do 141 i = 1, llc
  qq(ll(i)) = qq(ll(i)) + 1.d0
continue
```

数え上げ処理を別ループに

SIMD化促進の性能改善効果



	as is	ループ変形後
SIMD命令率(%)	0	**39**
L1D$ミス率(%)	0.8	1.6
L2$ミス率(%)	0.5	1.3
メモリスループット(GB/秒)	14	**165**

SIMD化促進のためのループ分割が必要

Copyright 2015 FUJITSU LIMITED

ミニアプリによる評価

- Fiberミニアプリ

Copyright 2015 FUJITSU LIMITED

Fiberミニアプリ†とは

FUJITSU

- 理化学研究所 計算科学研究機構(AICS)がアプリ開発者と協力して開発・整備している日本発のミニアプリ集
- アプリケーションの種類

アプリ名	概要
CCS QCD ★	高エネルギー物理学で用いられる格子量子色力学プログラム
FFVC-MINI ★	三次元非定常非圧縮性熱流体シミュレーションプログラム
NICAM-DC-MINI ★	全球雲解像モデルNICAMの力学コアを計算するプログラム
mVMC-MINI	多変数変分モンテカルロ法
NGS Analyzer-MINI	ヒト個人間の遺伝的差異やがんゲノムの突然変異を同定するプログラム
MODYLAS-MINI	高並列汎用分子動力学シミュレーション
NTChem-MINI ★	分子科学計算ソフトウェア
FFB-MINI ★	非圧縮流体の非定常流動を高精度に予測可能な Large Eddy Simulation(LES)に基づいた汎用流体解析コード

- 以降のスライドでは、★印の5本を用いてFX10からの性能向上を評価

† http://fiber-miniapp.github.io/

Copyright 2015 FUJITSU LIMITED

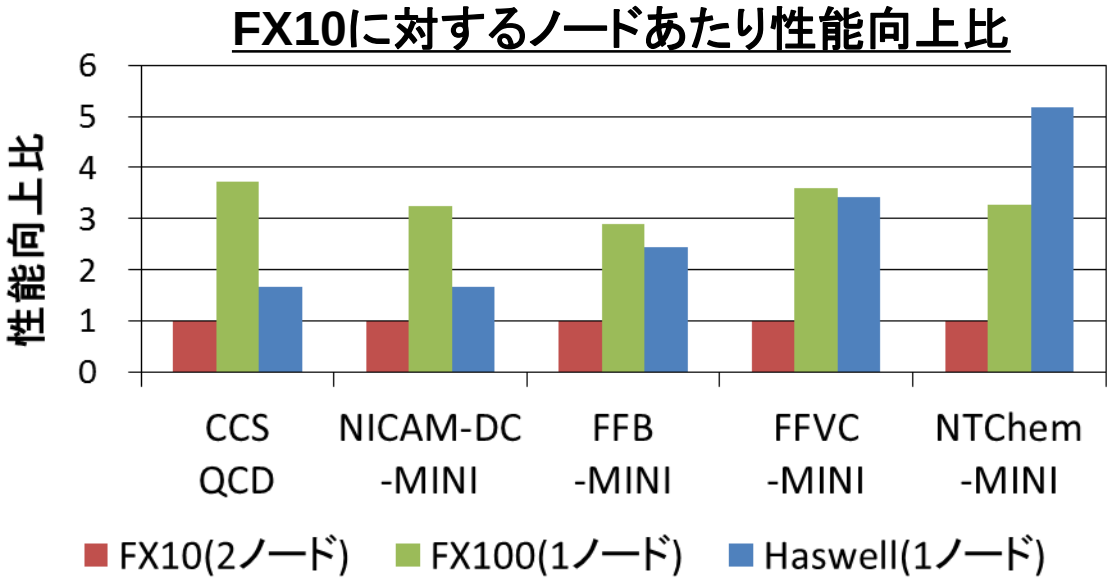
Fiberミニアプリによる評価

FUJITSU

■ 評価条件

アプリ名	問題サイズ	評価区間	スレッド数×プロセス数 (左からFX10、FX100、Haswell)		
CCS QCD	32×32×32×32	BiCGStab	16t×2p	32t×1p	16t×2p
NICAM-DC-MINI	gl05rl00z80pe10	Dynamics	3t×10p		
FFB-MINI	1,048,576要素	MAIN_LOOP	1t×32p	8t×4p	1t×32p
FFVC-MINI	256×256×256	Total	4t×8p		16t×2p
NTChem-MINI	taxol	RIMP2_Driver	1t×32p	16t×2p	2t×16p

■ 測定結果



FX100は、FX10比平均3.3倍、Haswell比平均1.4倍のノードあたり性能

注) FFVCには開発版コンパイラ、NTChemには開発版数学ライブラリを使用。QCDではセクタキャッシュ利用、FFBではループリローリングのコード変更を実施

Copyright 2015 FUJITSU LIMITED

Fiberミニアプリの性能分析 (1/2)

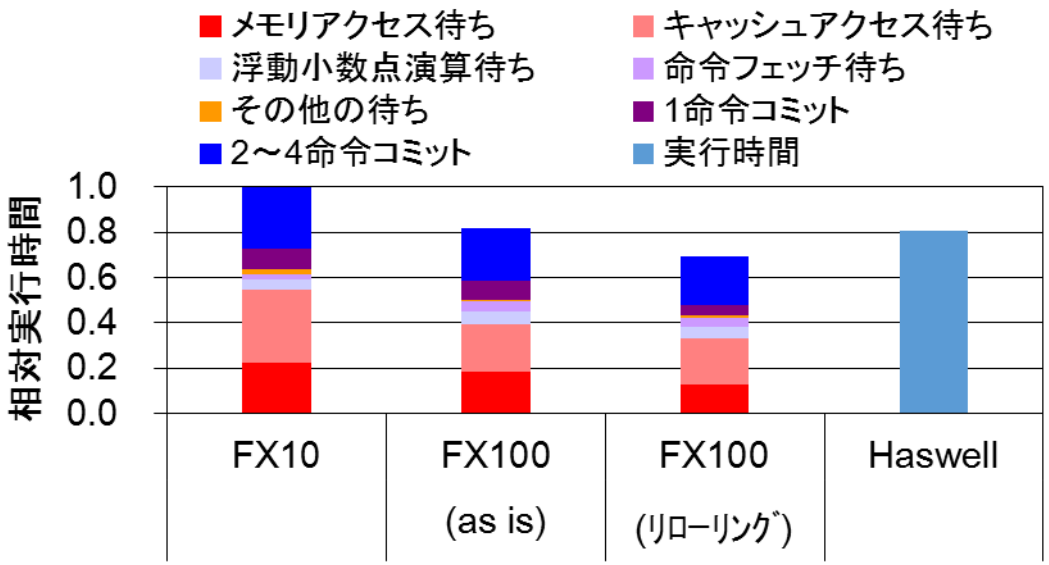
FUJITSU

FFB-MINI

- メモリ/キャッシュアクセス待ちが多く、メモリ律速傾向 (リローリング後169GB/秒)
- as isではアンローリングされているが、ループの形に戻す(ループリローリング)ことで連続SIMD化され、性能向上

連続SIMD化したほうが性能を出しやすい

FFB-MINIのサイクルアカウンティング結果



ループリローリングの例 (grad3x)

```
DO IE=IES,IEE
  IP1=NODE(1,IE)
  …
  IP8=NODE(8,IE)
  WXYZ(1,IP1)=WXYZ(1,IP1)-…
  …
  WXYZ(3,IP1)=WXYZ(3,IP1)-…
  WXYZ(1,IP2)=WXYZ(1,IP2)-…
  …
  WXYZ(3,IP8)=WXYZ(3,IP8)-…
CONTINUE
```

```
DO IE=IES,IEE
  DO J=1,8
    IP=NODE(J,IE)
    DO I=1,4
      WXYZ(I,IP)=WXYZ(I,IP)-…
CONTINUE
```


Copyright 2015 FUJITSU LIMITED

Fiberミニアプリの性能分析 (2/2)

FUJITSU

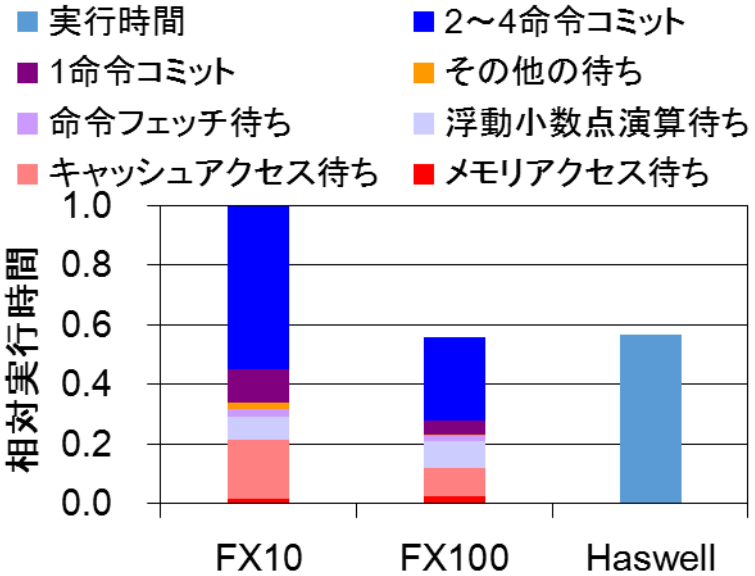
FFVC-MINI

- 命令コミット時間が多く、演算律速傾向
- 論理演算のSIMD化により性能向上を実現

SIMD化される論理演算の例

```
do i=1,ix
  …
  ndag_w=real(ibits(idx,bc_ndag_W,1))
  …
end do
```

FFVC-MINIのサイクルアカウンティング結果



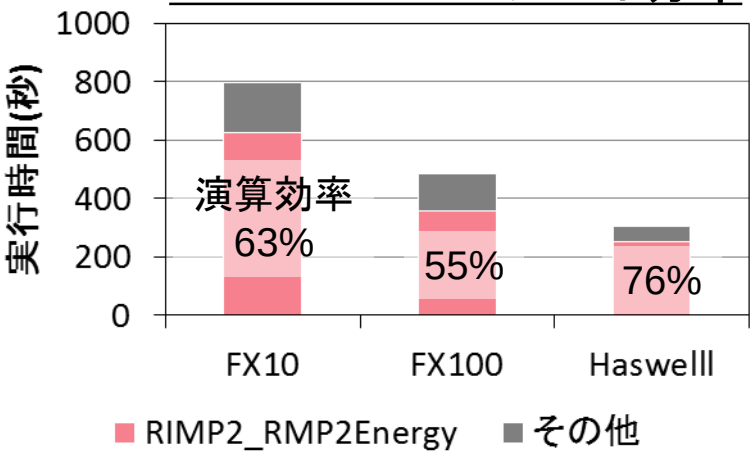
NTChem-MINI

- DGEMM演算がコストの大半を占める (RIMP2_RMP2Energy区間)
- 行列Aが横長、行例Bが縦長で、演算効率が低い⇒今後の改善課題

NTChemのDGEMM呼び出し例

```
CALL DGEMM('T','N',226,226,4186,...)
```

NTChem-MINIのコスト分布



Copyright 2015 FUJITSU LIMITED

SS研ポストペタアプリ性能WG報告より

- GT5D

Copyright 2015 FUJITSU LIMITED

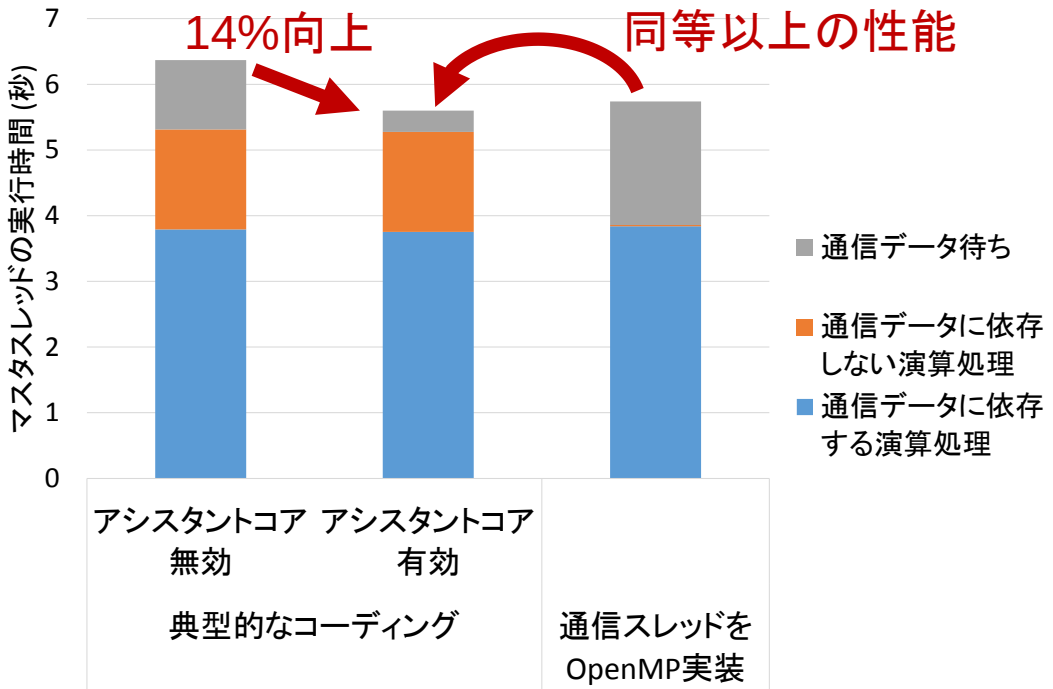
GT5D ～アシスタントコアによる通信/演算オーバラップ～

FUJITSU

- 通信データに依存しない演算処理を、MPI_Waitallの前で実行
  ⇒演算中にアシスタントコアが通信を制御
- 性能向上と可搬性/保守性を両立
  - 従来はOpenMPで通信/演算オーバラップを実装し強スケーリングを達成していたが、アシスタントコアの利用により特殊な実装なしでオーバラップ可能に

通信/演算オーバラップの典型的コード

```
   MPI_Isend
   MPI_Irecv
!$OMP PARALLEL DO
   do i=…
      通信データに依存しない演算処理
   enddo
!$OMP END PARALLEL DO
   MPI_Waitall
   do i=…
      通信データに依存する演算処理
   enddo
```



問題サイズ:256×256×64×128、並列数:16スレッド×64プロセス、評価対象区間:l4dx、通信促進:自動区間モード

Copyright 2015 FUJITSU LIMITED

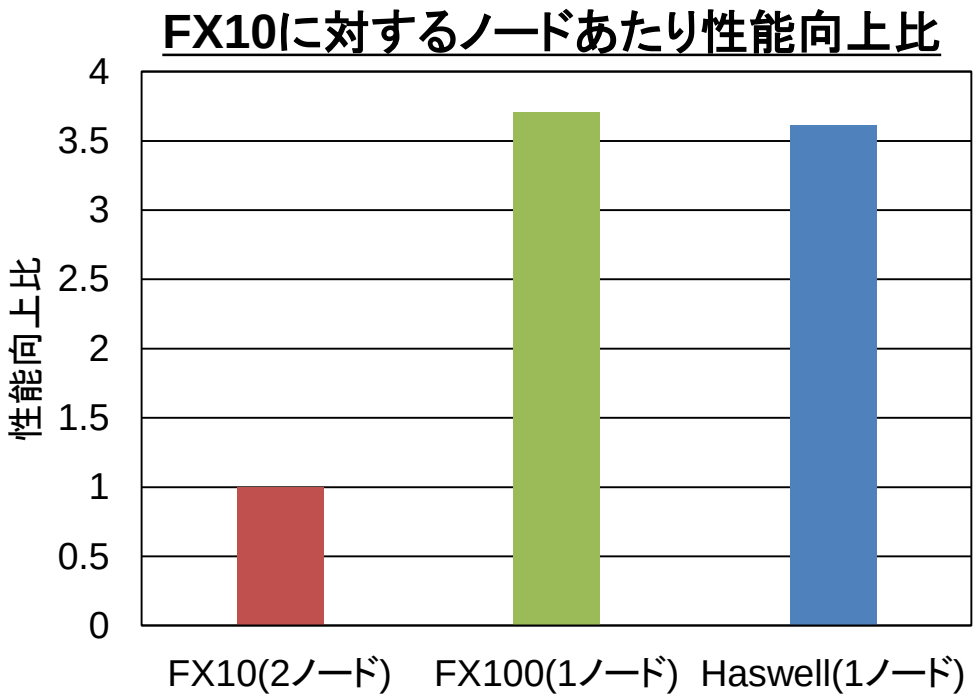
C++アプリの性能

- LULESH
- OpenFOAM

Copyright 2015 FUJITSU LIMITED

LULESH†による評価

- C++の流体力学アプリ。CORALプロジェクトのBMTコード‡の1つ
  - CORALプロジェクト…2017年頃にLLNL、ORNL、ANLへ次期スパコンを導入
- 評価条件: 4スレッド×8プロセス、49^3要素/プロセス
- 測定結果



  - as isコードの性能は、FX10比3.7倍、Haswell比1.03倍

† https://codesign.llnl.gov/lulesh.php
‡ https://asc.llnl.gov/CORAL-benchmarks/

Copyright 2015 FUJITSU LIMITED

LULESHの性能分析

FUJITSU

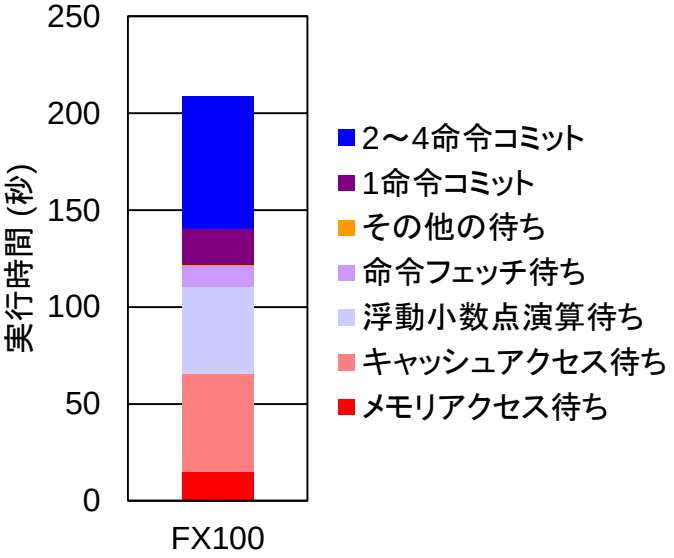
■ コード例とサイクルアカウンティング結果

コード例 (関数CalcKinematicsForElems)

```
#pragma omp parallel ...
for(k=0; k<numElem; ++k) {
  elemToNode = domain.nodelist(k);
  CollectDomainNodesToElemNodes(...);
  volume = CalcElemVolume(...);
  ...
  domain.arealg(k) = CalcElemCharacteristicLength(...);
  for(lnode=0; lnode<8; ++lnode) {...}
  ...
  for(j=0; j<8; ++j) {...}
  CalcElemShapeFunctionDerivatives(...);
  CalcElemVelocityGradient(...);
  domain.dxx(k) = D[0]; ...; domain.dzz(k) = D[2];
}
```

ループボディが大きい
関数呼び出しをインライン展開

サイクルアカウンティング結果



	FX100
SIMD命令率(%)	**12**
L1D$ミス率(%)	1.4
L2$ミス率(%)	0.7
メモリスループット(GB/秒)	**79**

■ メモリバンド幅を活かすための性能向上課題

- ■ プリフェッチ命令生成の高度化
- ■ SIMD化促進のためのループリローリング
- ■ 命令スケジューリング改善のためのループ分割

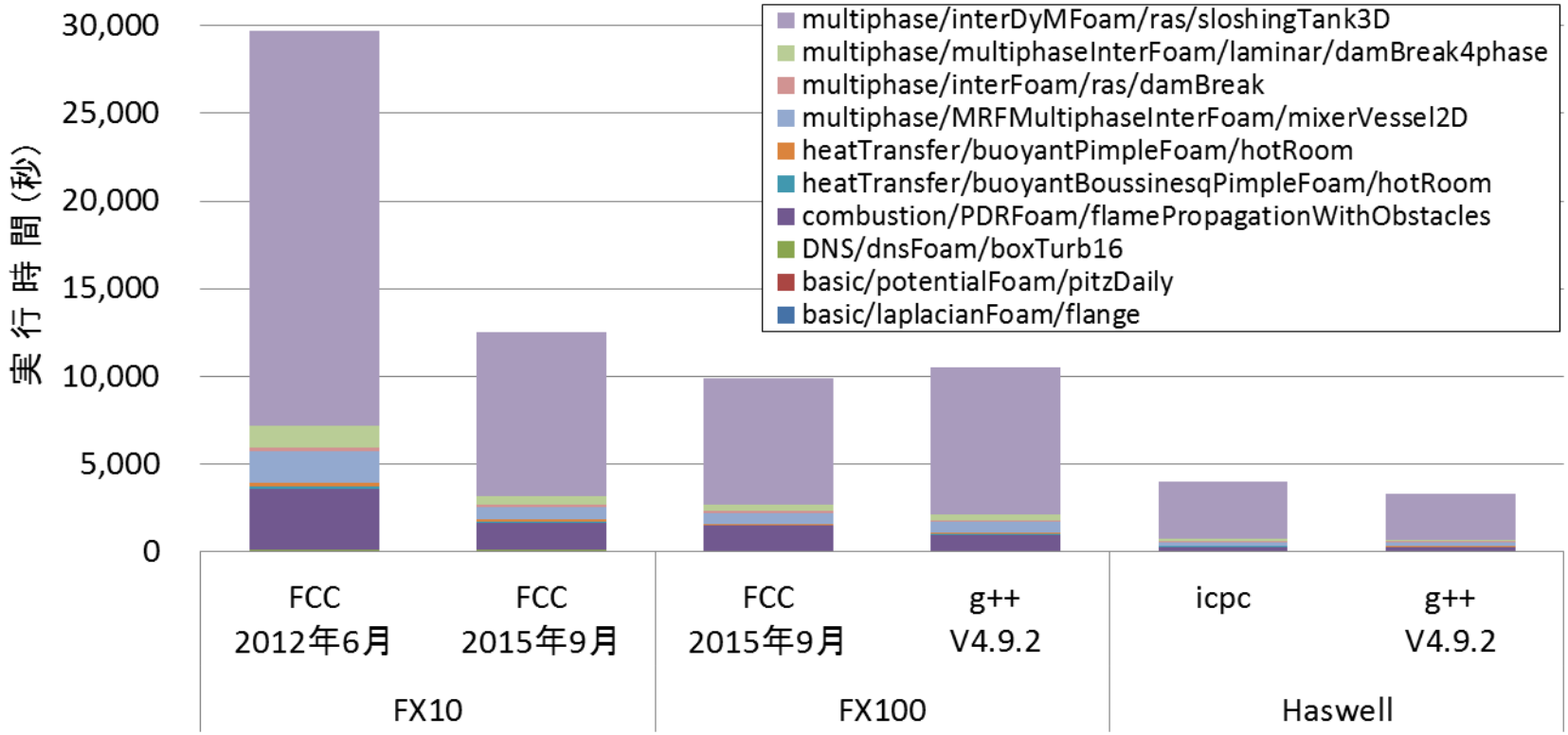
ソフトウェア面の改良で
まだ性能は上がるはず!

Copyright 2015 FUJITSU LIMITED

OpenFOAM†による評価

FUJITSU

- オープンソースのCFDソフトパッケージから、代表的なチュートリアル10本で1コア実行時間を比較



- 2012年より性能改善に取り組み、当初目標のg++性能に到達

改善項目: インライン展開の促進、オブジェクトコピーの削減、
STLライブラリの改良 など

† http://www.openfoam.com/

Copyright 2015 FUJITSU LIMITED

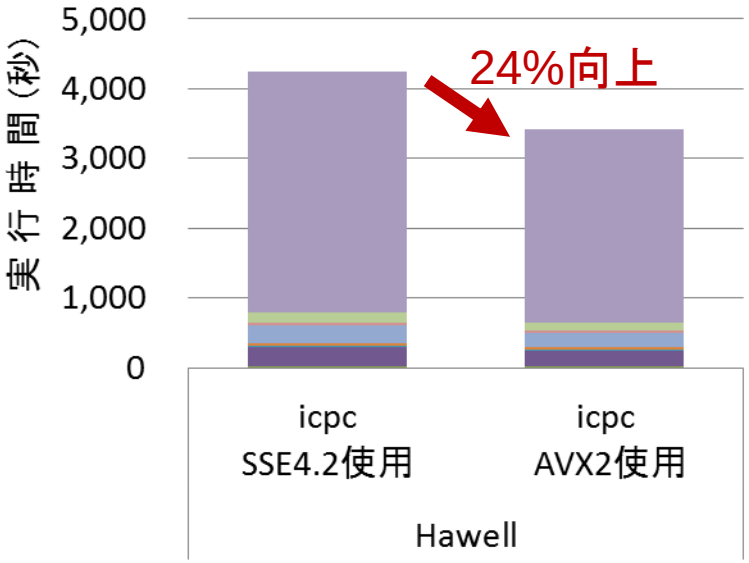
OpenFOAMの性能分析

FUJITSU

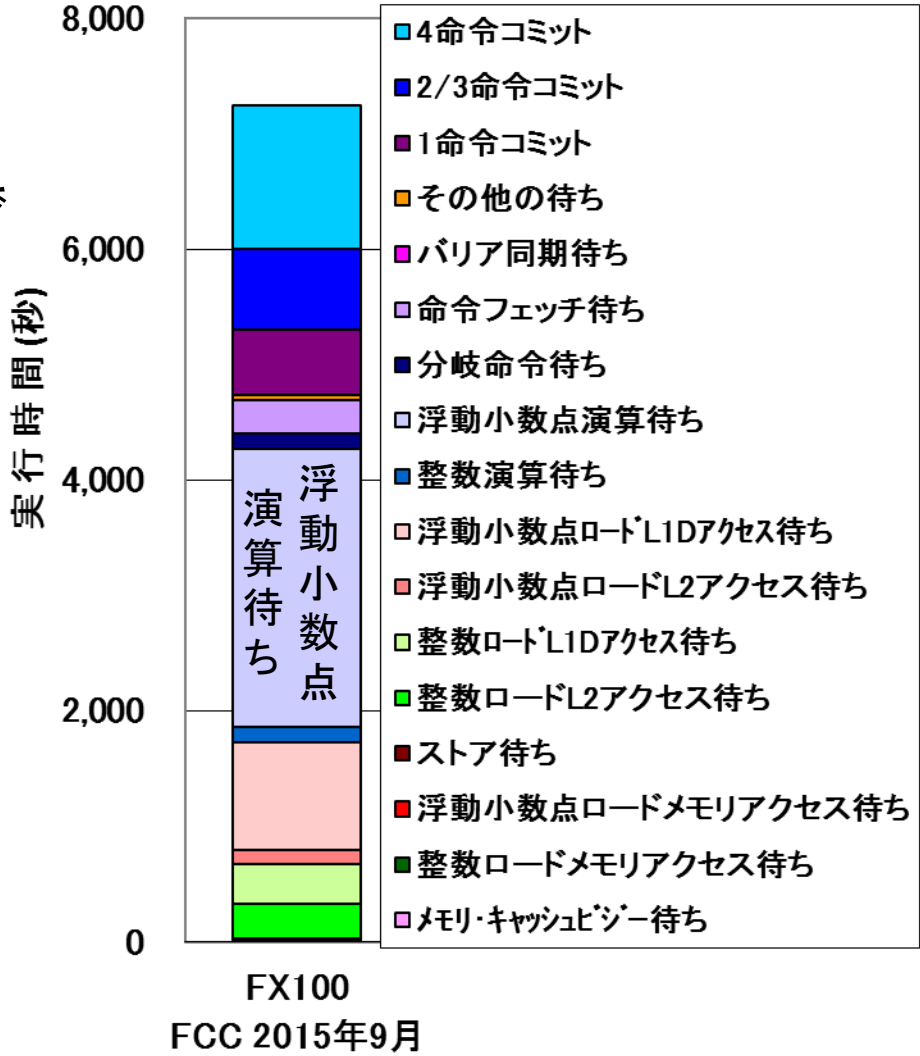
- さらなる改善項目の検討
  - サイクルアカウンティングの結果より、浮動小数点演算待ちの削減が課題
  - 現状のSIMD命令率は1%程度
  - 一方Haswellでは、SSE4.2→AVX2で24%の性能向上を実現

⇒C++向けのSIMD化強化が必要

SSE4.2とAVX2の性能比較 (Haswell)



サイクルアカウンティング結果 (sloshingTank3D、FX100)



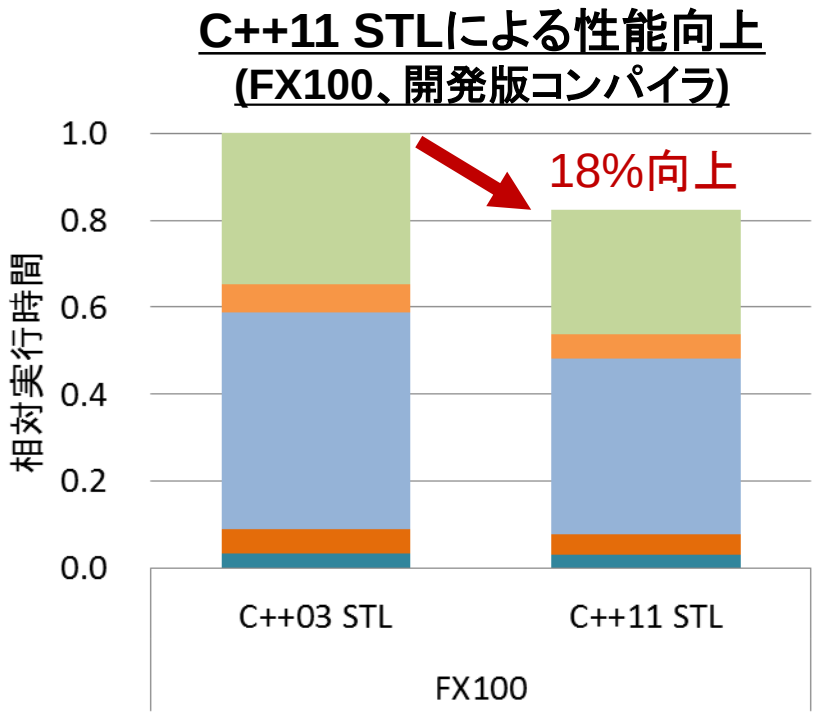
FCC 2015年9月

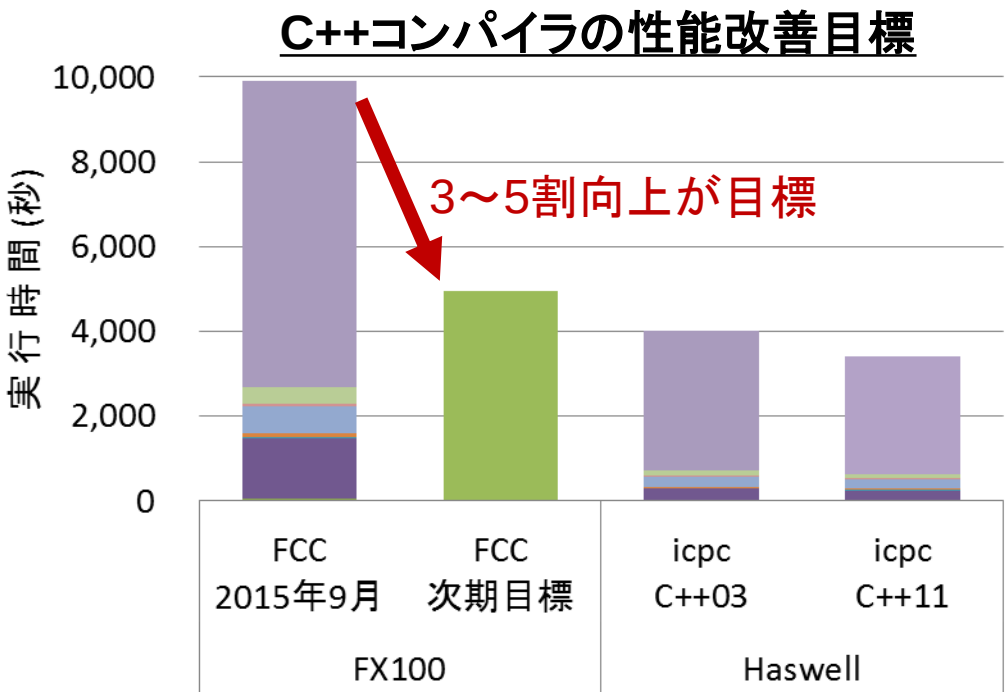
Copyright 2015 FUJITSU LIMITED

C++11規格対応による性能向上

FUJITSU

- 現在、C++11規格に対応するコンパイラを開発中
- 開発版コンパイラでは、チュートリアル5本で平均18%の性能向上
  - C++11 STLライブラリの利用による効果
- その他の改良と併せて、合計で3～5割の性能向上が次なる目標





引き続き、C++アプリの性能向上に取り組んでいきます

Copyright 2015 FUJITSU LIMITED

まとめ

Copyright 2015 FUJITSU LIMITED

まとめ

FUJITSU

- PRIMEHPC FX100は、同FX10や京コンピュータとの互換性/連続性を保ちつつ、性能強化された最新CPU SPARC64 XIfx、高バンド幅メモリHMC、およびバンド幅強化されたTofuインターコネクト2を搭載
- その結果、NAS Parallel BenchmarksやFiberミニアプリによるFX100の性能評価では、FX10に対して3.3～3.9倍、Haswellに対して1.3～1.4倍のノードあたり性能を確認
- C++アプリについても、継続的な性能改善の取り組み効果が見られ、引き続き、最新言語規格への対応などで性能向上させる予定

FX100でのこれら実績を踏まえつつ、
富士通ではエクサ世代の開発にも取り組んでいきます

Copyright 2015 FUJITSU LIMITED

FUJITSU

shaping tomorrow with you