マルチプラットフォーム上でのHPFコンパイラの評価

浅岡香枝*　村井均**　平野彰雄*
岡部寿男*　金澤正憲*

* 京都大学 学術情報メディアセンター
** 海洋科学技術センター　地球シミュレータセンター

背景(1)

- HPF(High Performance Fortran)
  - Fortranを拡張したデータ並列言語
  - 最小限の指示文の追加で容易に並列化することで高性能を引き出す
  - さまざなプラットフォームへ容易に移植可能な高い可搬性をもつ
- HPF仕様の標準化
  - HPF2.0(基本仕様＋公認拡張仕様)（1997）
    - 基本仕様: 実装が必須
    - 公認拡張仕様: 実装が望まれるが必須でない
  - HPF/JA拡張仕様　　（1999）
  - ➢ HPF/JA拡張仕様までサポートした処理系
    NEC HPF/SX,HPF/ES　Fujitsu HPF/VPP

⇒新世代のFortranとしての普及の進展に期待！

背景(2)

しかしながら…

- 仕様では実装方法まで決められていない
- 仕様のどこまでをサポートするかは処理系に依存

→ **実行性能は処理系ごとに大きく異なる可能性**

- 従来の研究

特定のプラットフォーム/コンパイラを対象として、実行性能を評価

我々のアプローチ

複数のプラットフォーム/コンパイラを比較

- 文法レベルの互換性
- 同じコードが複数のプラットフォーム/コンパイラでどれくらいの性能差があるのか

今回のアプローチ

- 複数のプラットフォーム
- 複数のHPFコンパイラ
- 複数のHPFコード

を用いて

- 文法レベルの互換性を検証
- 実行性能を評価

京都大学 学術情報メディアセンター
Fujitsu　VPP800/63　(VPP)

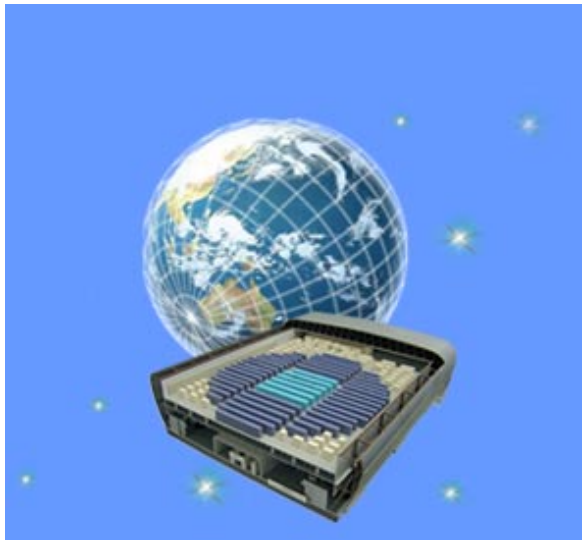
- PE(Processor Element) 台数　:63
- 各PEはベクトル型
  - ベクトル演算性能　:8GFlops / PE
  - メモリ　:　8GB / PE
- 分散メモリ型並列
  - PE間はクロスバ結合

海洋科学技術センター 地球シミュレータセンター
地球シミュレータ(NEC製造)　(ES)

- PN(Processors Node)台数　:640
- 各PNはSMP構成
  - AP(Arithmetric Processos)　:　8個
  - メモリ　:　16GByte
  - ベクトル演算性能　:　8GFlops / AP
- PN間はクロスバ結合

✓ 1 AP = 1 MPIプロセスを起動

✓ 8並列まではノード内、
それ以上はノード間に
またがって実行



京都大学学術情報メディアセンター
Fujitsu　GP7000F/M900 (SPP)

- 1台のSMP構成
  - CPU：24台
  - メモリ：24GByte
- CPU
  - スーパースカラ型　SPARC64GP
  - ピーク性能　1.35GFLOPS
  - 一次キャッシュ128KB
    （命令64KB、データ64KB）
  - 二次キャッシュ　8MB

HPFコンパイラ

コンパイラ	サポートする仕様	備考
HPF/VPP	HPF2.0 HPF/JA拡張	・富士通開発
HPF/ES	HPF2.0 HPF/JA拡張	・NEC開発 ・MPIトランスレータ
Adaptor	HPF2.0	・GMD National Research Center開発 ・パブリックドメイン ・MPIトランスレータ ・ VPP,ES,SPPに移植

HPFのコード

- HPFBench
  - http://www.cs.rice.edu/~ychu/hpfbench
- NAS　Parallel Benchmarks
  - 我々が開発したHPF/VPP向けにチューニングしたNPB
- Impact-3D
  - 3次元流体シミュレーションの実用コード
  - 姫路工業大学の坂上助教授からの提供

HPFBenchを用いた評価

- HPFBench　Hu,Y.C. et.al(2000)
  - HPFの汎用ベンチマークスウィート
  - ベンチマーク報告例
    - PGHPF(Portrand Groups,Inc),IBM　SP2
      Hu,Y.C. et.al(2000)
    - HPF/SX　Murai,H. et .al (2002)
- 今回とりあげたベンチマーク
  - diff-3d（3次元拡散方程式）
  - gmo（一般化された地震揺動）
  - n-body（n体問題）4種
    broadcast,cshift,cshift-symmetry,spread

コンパイルの可否

	HPF/VPP	HPF/ES	Adaptor
計時等の共通ルーチン	×	○	×
diff-3d	×	○	○
gmo	×	○	△
n-body(broadcast)	△	○	×
n-body(cshift)	△	○	×
n-body(cshift-sym)	△	○	×
n-body(spread)	△	○	×

○ 修正不要
△ HPF指示文の修正が必要
× Fortran部分の修正が必要

文法レベルの互換性は不十分

- HPFBenchベンチマークルール
  - HPF指示文の書換えは許す（△までは許容範囲）

必要となった修正（1）

- !hpf$ distribute kout(block),kin(block)

  PGHPF独自拡張。HPF2.0仕様を逸脱

  指示文の修正

  !hpf$ distribute kout(block)

  !hpf$ distribute kin(block)

- !hpf$ distribute dynamic(*, block)

  変数名dynamicがdynamic属性（公認拡張仕様）と誤って解釈

  指示文の修正

  !hpf$ distribute (*,block) :: dynamic

  HPF/VPP: OK

  Adaptor : NG → 変数名の変更

必要となった修正（2）

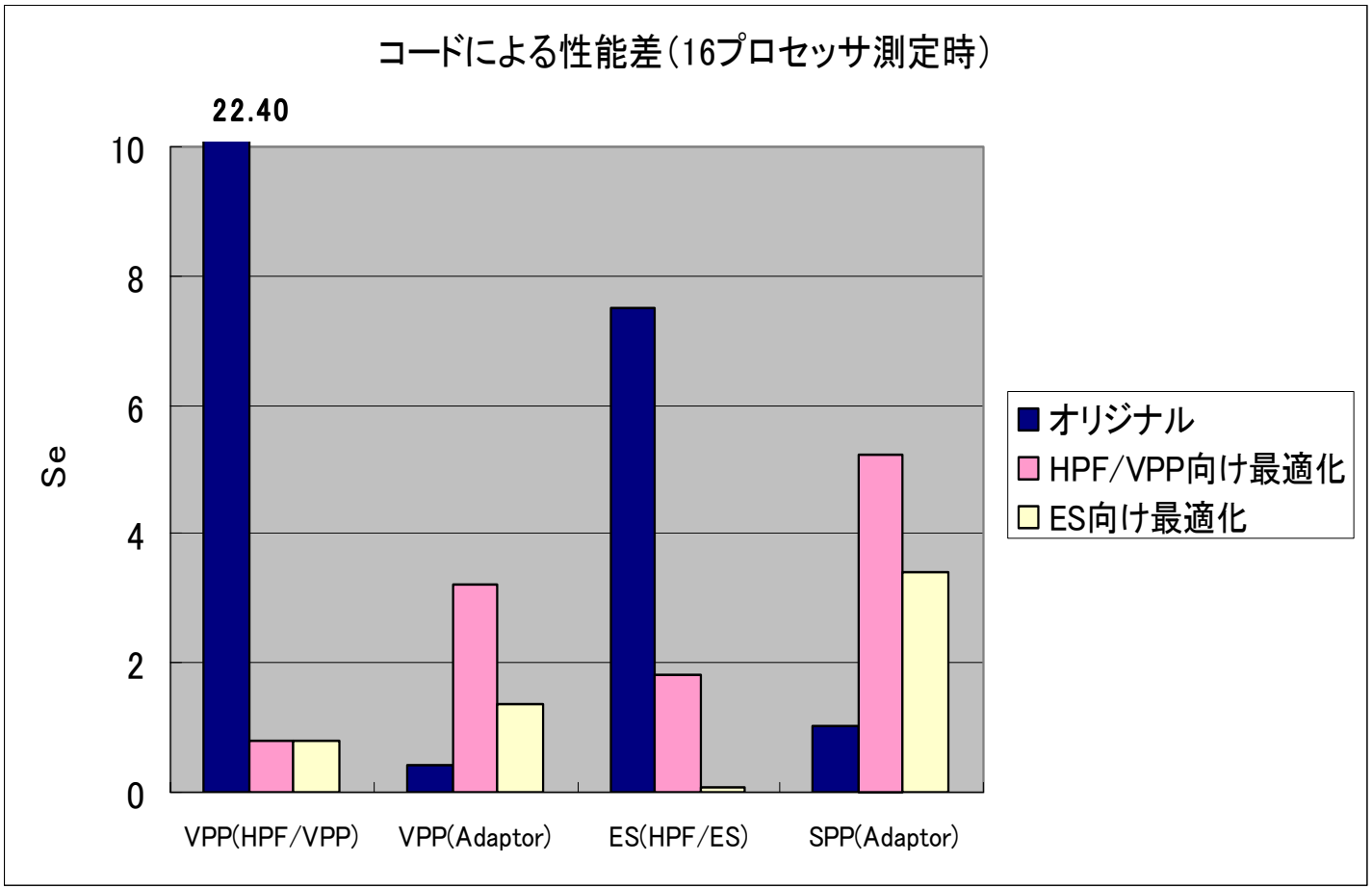
- HPF/VPPの制限のため
  - 整合配列渡し

    → 形状引継ぎ配列渡しへ変更
  - Fortran90標準ルーチン

    system_clock、random_number

    → 別ルーチンへ置換え

性能測定 diff-3d

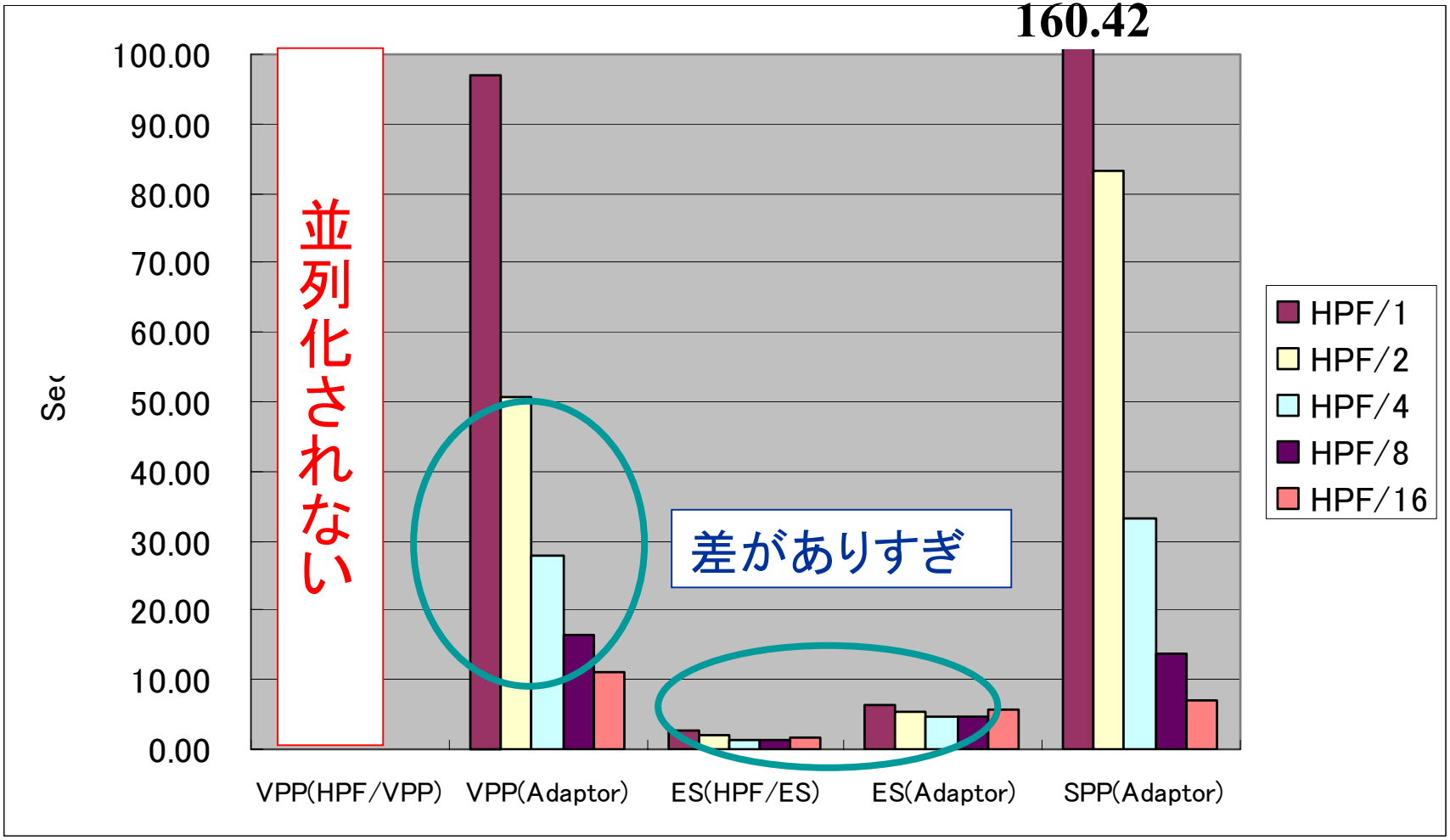


HPF/VPP向け最適化
　オリジナルコードをベースに
　・配列代入文をforallに書き換え
　・詳細にHPF指示文を指定
　・分散配列をcommonに移動

HPF/ES向け最適化
　HPF/VPP向け最適化をベースに
　・配列宣言の変更（バンクコンフリクト回避）
　・詳細してしたHPF指示文を削除

性能測定 n-body(broadcast)

n-body(broadcast) :　配列演算式



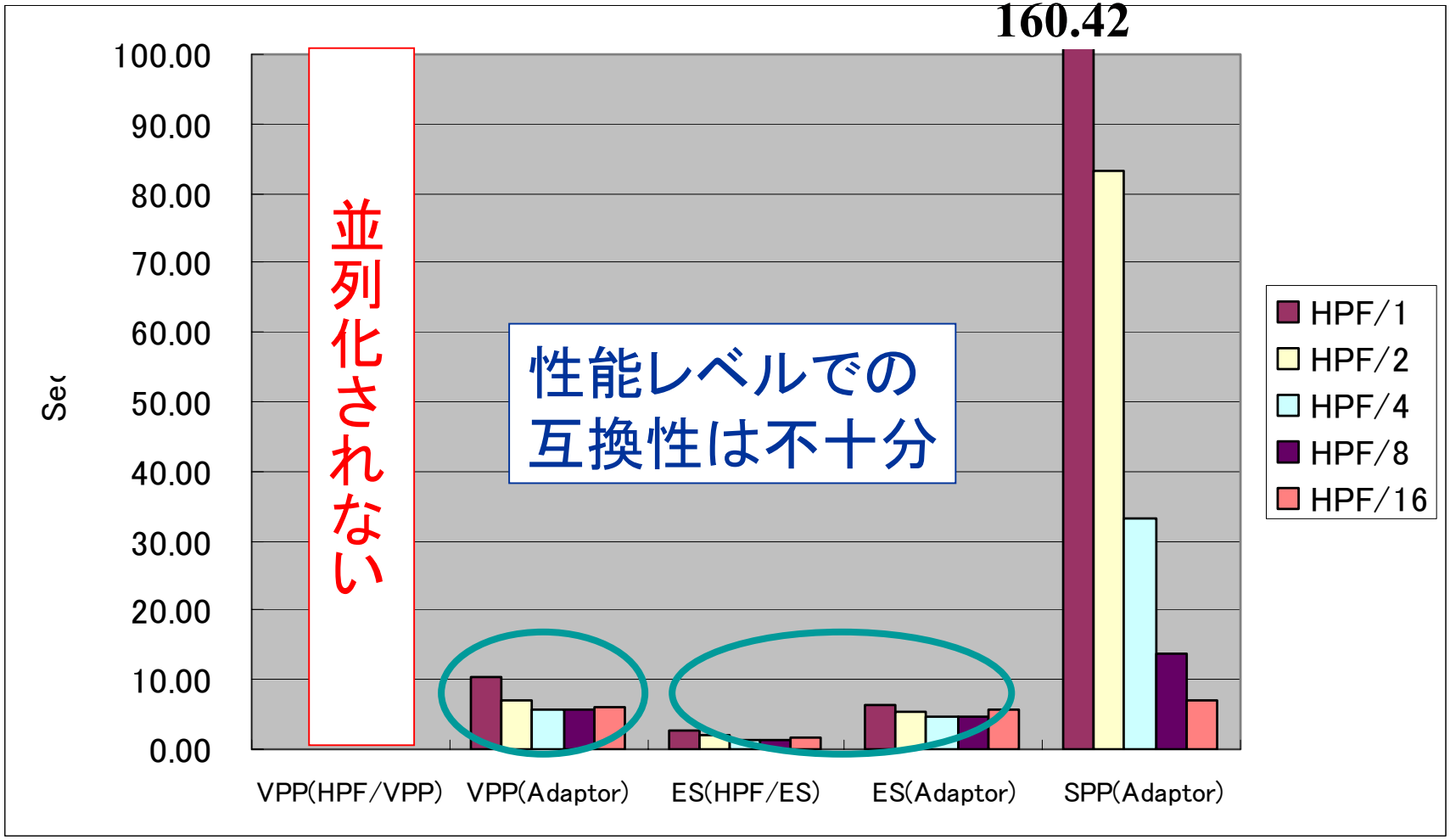
性能差の原因

- Adaptorが生成する中間コード
  - -vectorオプション指定時
  - NECのベクトルコンパイラ用最適化指示文(!CDIR)が含まれる

→ ESで、ベクトル化が促進

VPPのベクトルコンパイラ用指示文に置き換え(!ocl)

性能測定 n-body(broadcast)
（最適化指示文置換え後）



NPBを用いた評価

- HPF/VPP向けに開発したNAS Parallel Benchmarks
  - Applied Parallel Research社公開のHPF実装版NPB* ベース

    *(ftp://infomall.org.tenants/apri/Benchmarks)
  - HPF2.0およびHPF/JA拡張仕様の指示文を、可能な限り詳細に挿入

Asaoka,K.et.al 2002

(HOKKE2002 および ISHPC-IV/HiWEP2002)

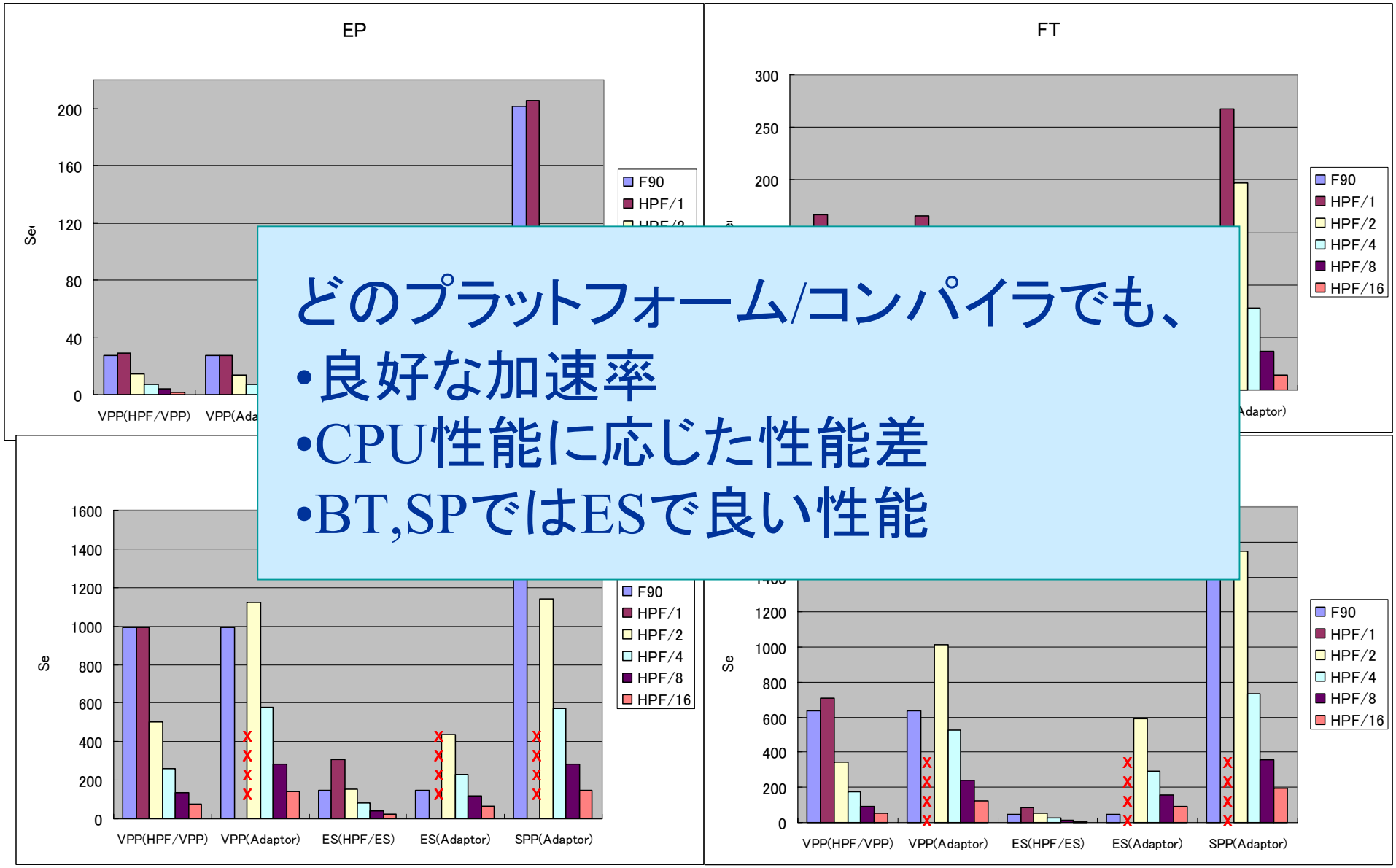
HPF/JA拡張仕様の実装上の相違

- local指示

	Local節 !hpf$ on home(..),local	Local指示文 !hpfj local
HPF/VPP	○	○
HPF/ES	○	×

- local指示とreduction指示重複時の扱い
  - ➢ 仕様には明確化されていない
    - HPF/VPP　reduction指示
    - HPF/ES　　local 指示（warning　messageあり）

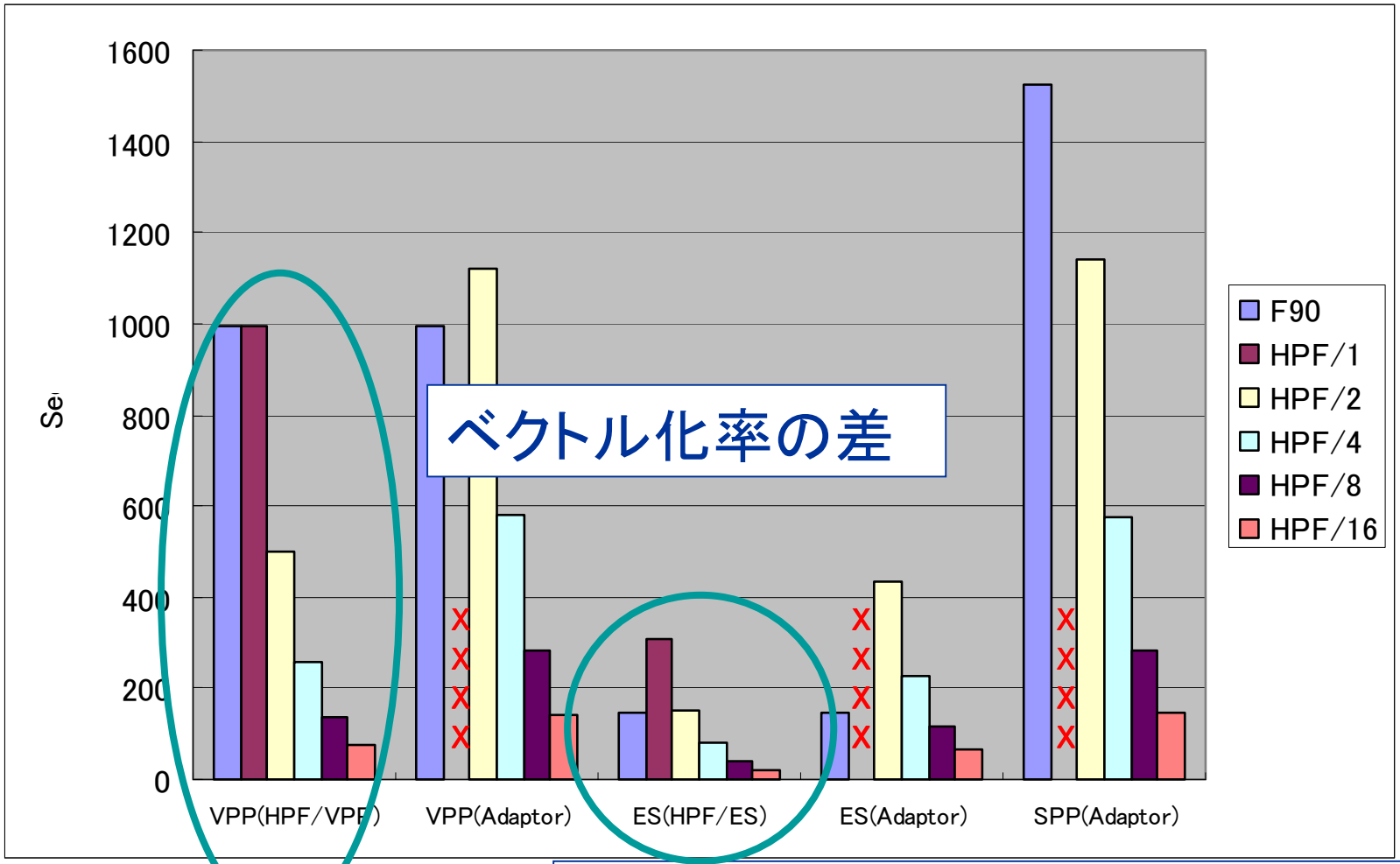
NPB　実行性能



どのプラットフォーム/コンパイラでも、

- 良好な加速率
- CPU性能に応じた性能差
- BT,SPではESで良い性能

BT(Block Tridiagonal simulated CFD application) 実行性能



Impact-3Dを用いた評価

- Impact-3Dとは
  - 3次元流体シミュレーションの実用コード
    by 坂上@姫路工大
  - HPF実装：HPF/SX用、HPF/VPP用
  - 地球シミュレータの512Nodeで14.9TFLOPS（ピーク性能比　45%）という高性能
  - 2002年Gordon Bell賞受賞

    Sakagami, H., Murai, H., et.al（2002）

- 今回の評価では、坂上助教授より提供を受けたHPF/VPP版を用いる

Impact-3D HPF/VPP版　実行性能

単位：（秒）

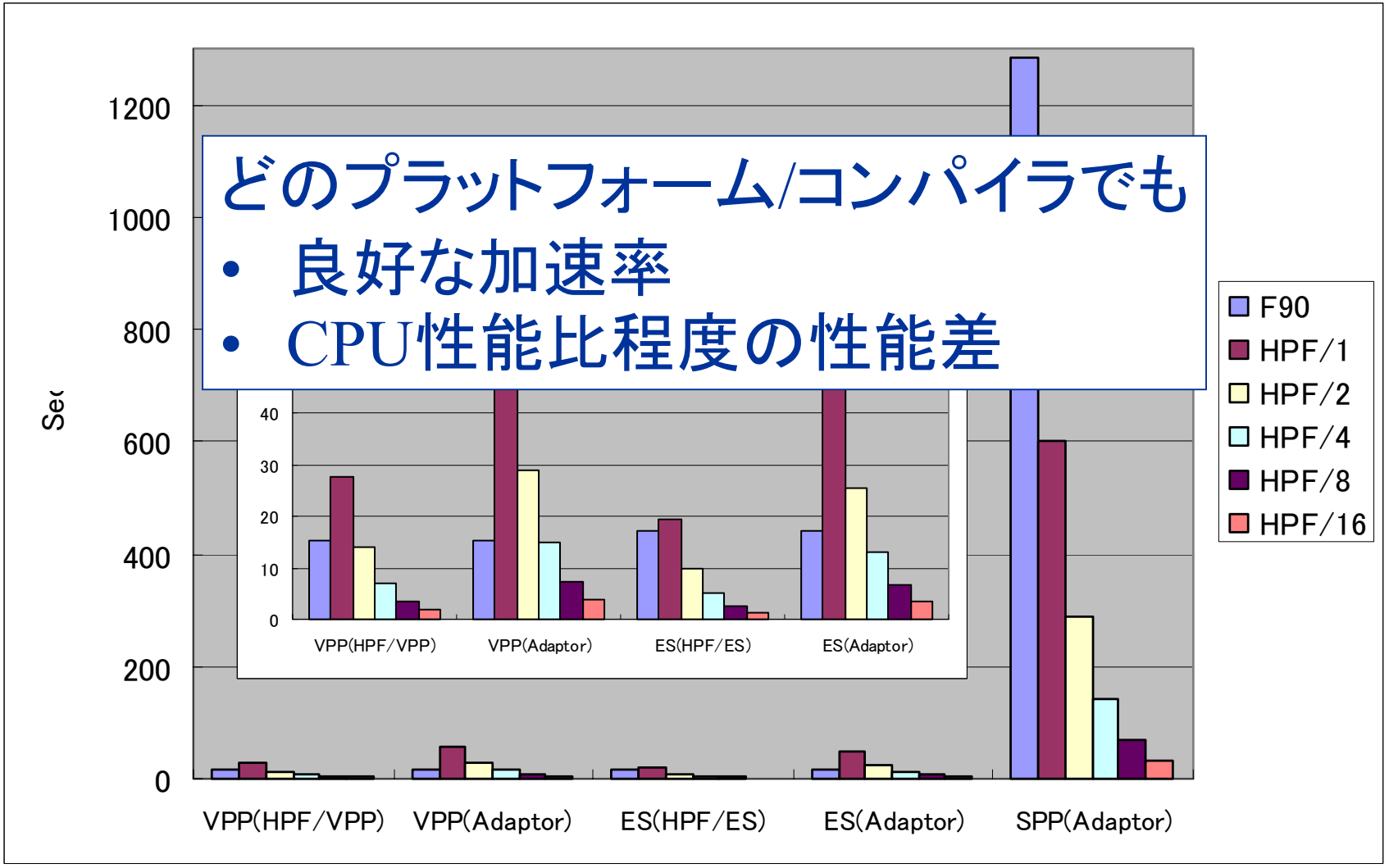
platform	compiler	F90	HPF/1	HPF/2	HPF/4	HPF/8	HPF/16
VPP	HPF/VPP	110.40	27.47	13.85	7.06	3.63	2.04
	Adaptor	15.28	57.50	28.93	14.78	7.39	3.79
ES	HPF/ES	17.22	19.31	9.85	5.03	2.58	1.32
	Adaptor		50.34	25.44	13.07	6.79	3.62
SPP	Adaptor	7934.91	1014.50	287.62	141.44	70.56	33.11
		1283.28	601.62				

- 配列のサイズ 128^3　→　バンクコンフリクト

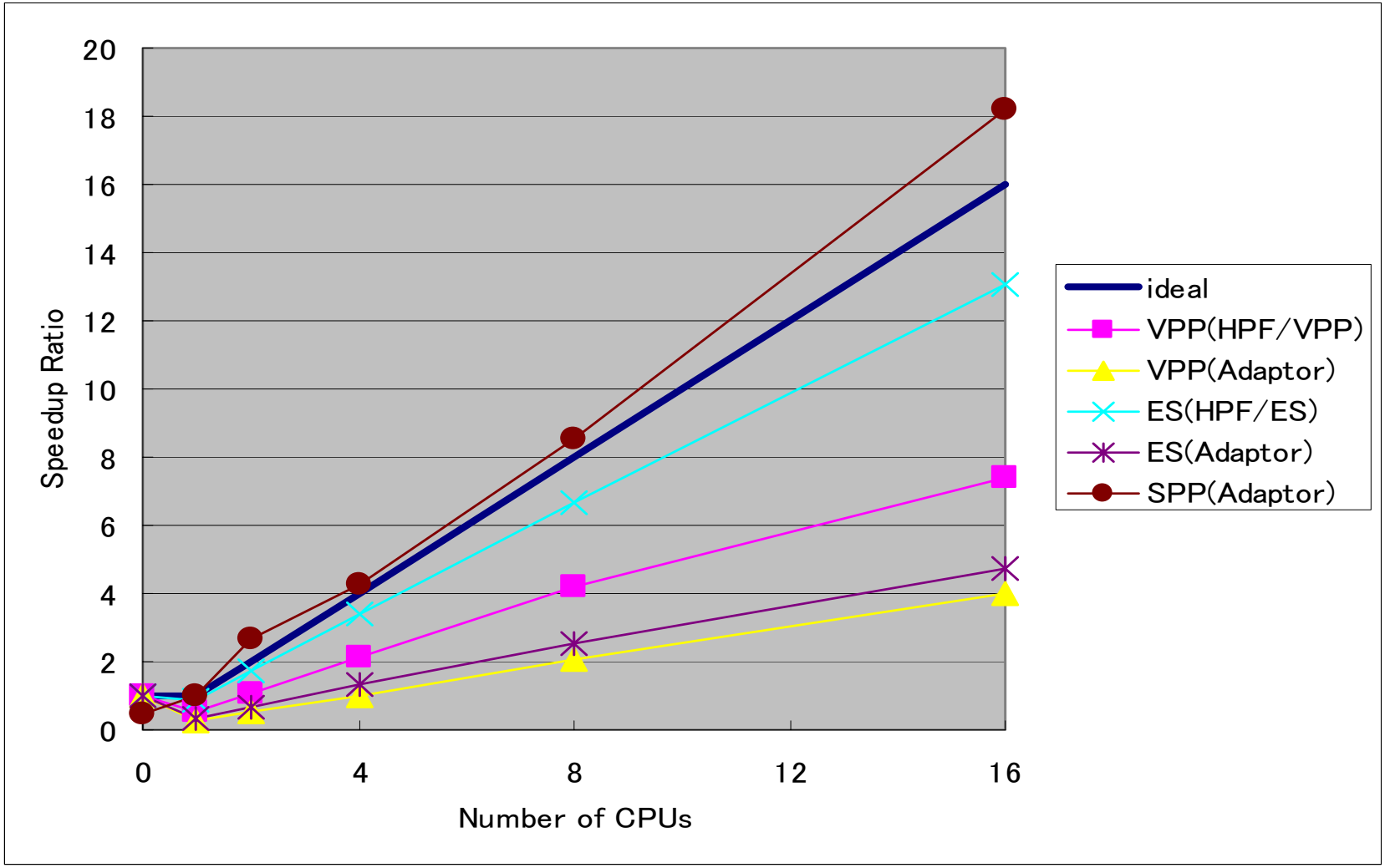
  1次元目のサイズを変更

Shadow領域の確保によるバンクコンフリクト発生条件の変化

Impact-3D HPF/VPP版　実行性能（2）



台数効果



台数効果はF90,HPF/1のうち速い方を1として算出

まとめと今後の課題

- 複数のHPFベンチマークコードを複数のプラットフォーム/複数のコンパイラ上で評価
- 文法レベルの互換性：十分ではないが、ユーザが簡単に対応できる範囲
- 性能レベルの互換性：性能まで含めた互換性があるかどうかは、コーディングスタイルに大きく依存

⇒ 基準、規範となるベンチマークコードの必要性

✓ 今後の課題

- より多くのプラットフォーム・コンパイラでの評価
- 実行時間とメモリ使用量とのトレードオフの評価
- NPB3.0α　HPF実装版　の評価

おまけ

（教訓）

- 結局、古典的なベンチマーク要素が重要
  - ベクトル化、キャッシュ、バンクコンフリクト、…
- 台数効果にだまされるな！
  - 良すぎる性能台数比は1CPU時の性能が何らかの理由で遅すぎることが原因
- ベンチマークには明確な方針を持って臨め！
  - いろんな要素に惑わされて、なにを測っているかわからなくなってくる