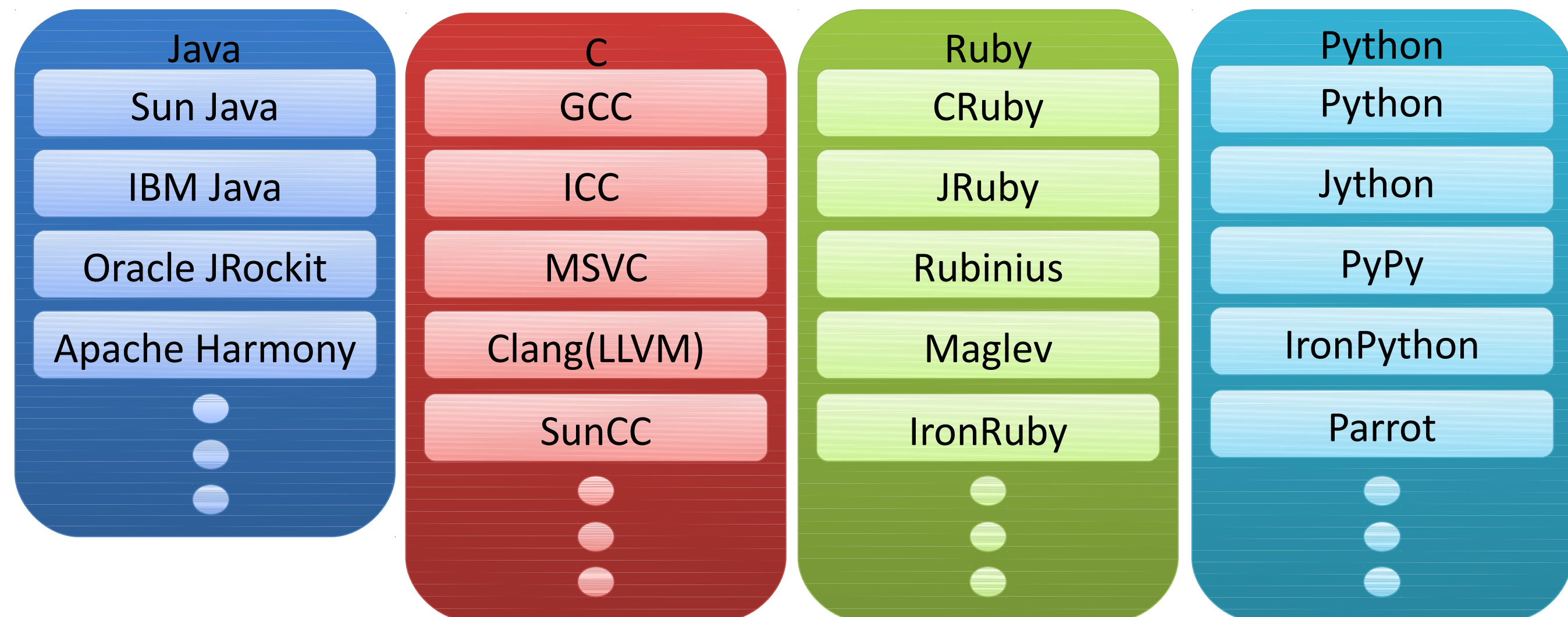


言語トランスレータを用いる多種言語処理系性能の評価

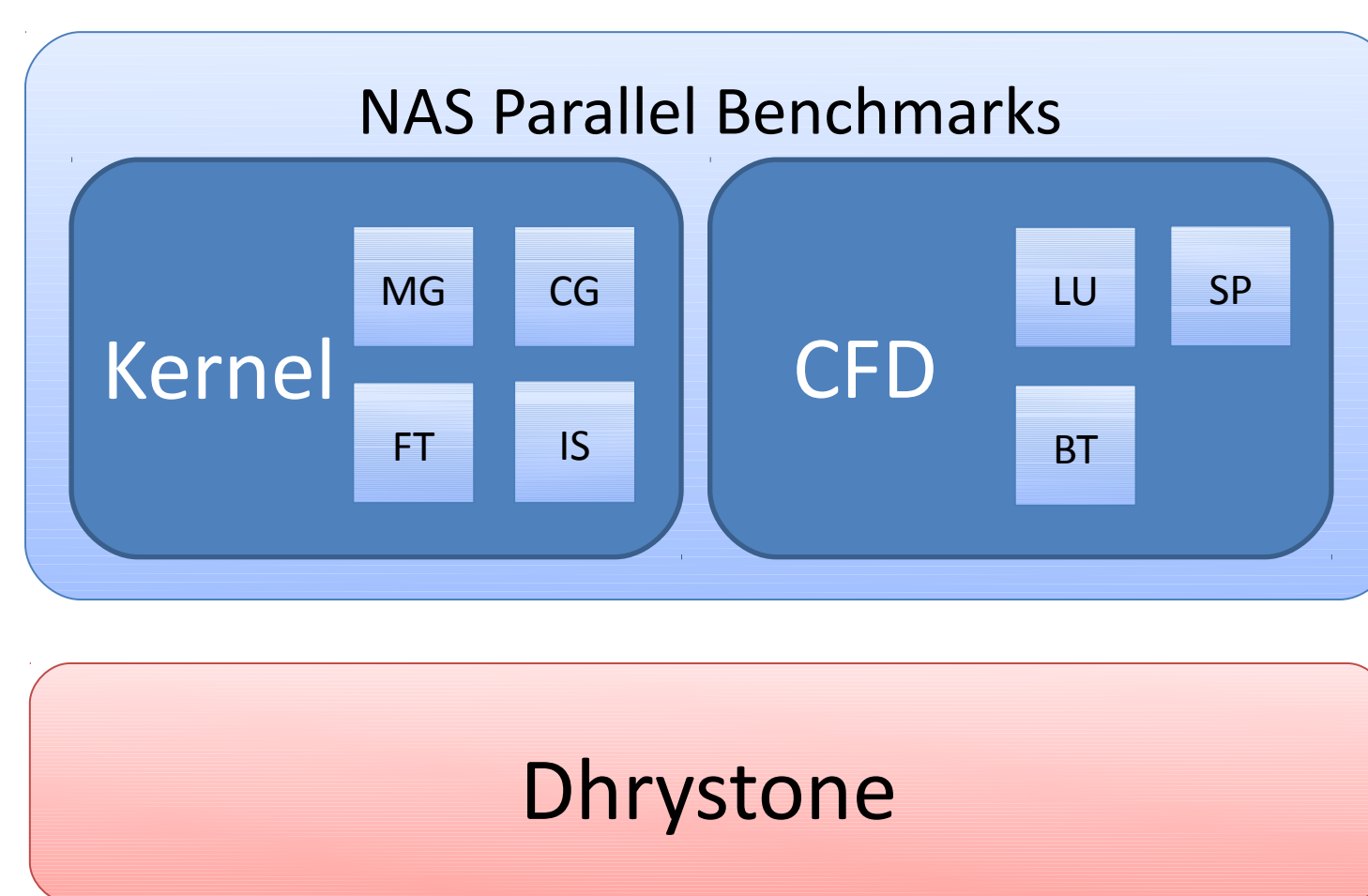
東京大学 野瀬貴史, 泊久信, 平木敬

言語比較ベンチマークの必要性

コンピュータプログラム, 特に数値計算プログラムを作成するにあたり, 生産性・性能・動作環境を勘案して言語及び言語処理系を選択することが重要であるが, 言語環境は多種多様であり, 性能を概観するために何らかの基準となるベンチマークを各言語で走らせることが必要である。



変換したベンチマークの概要と結果

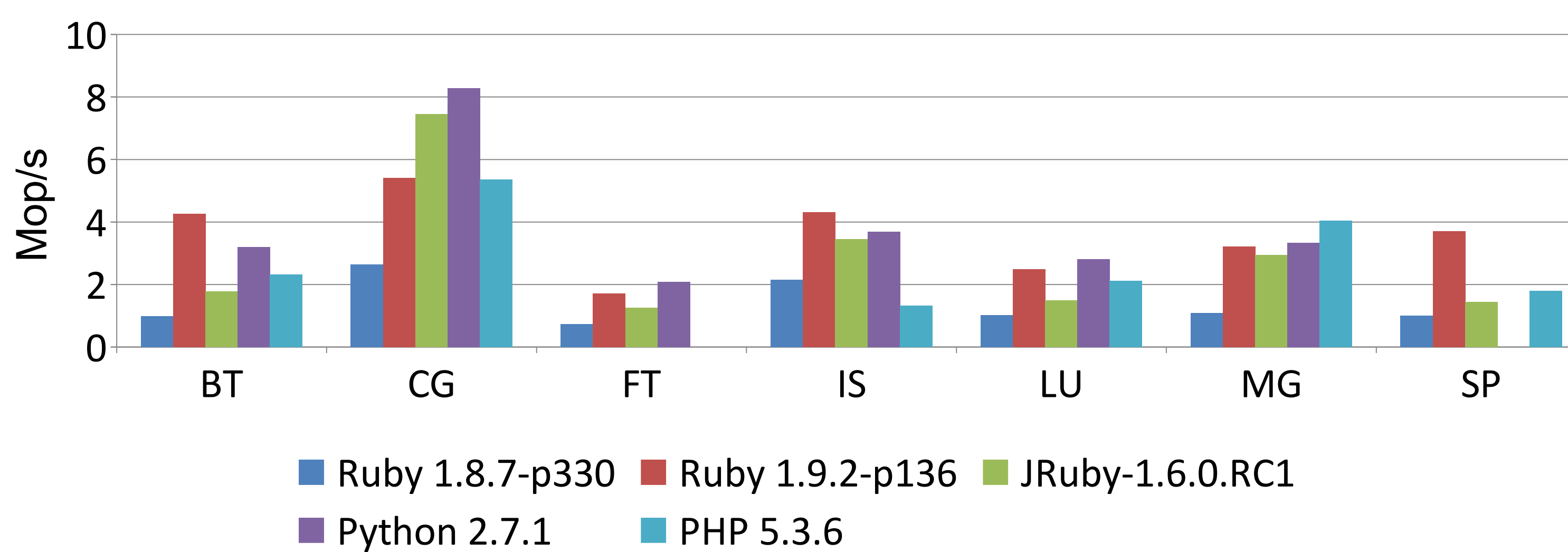
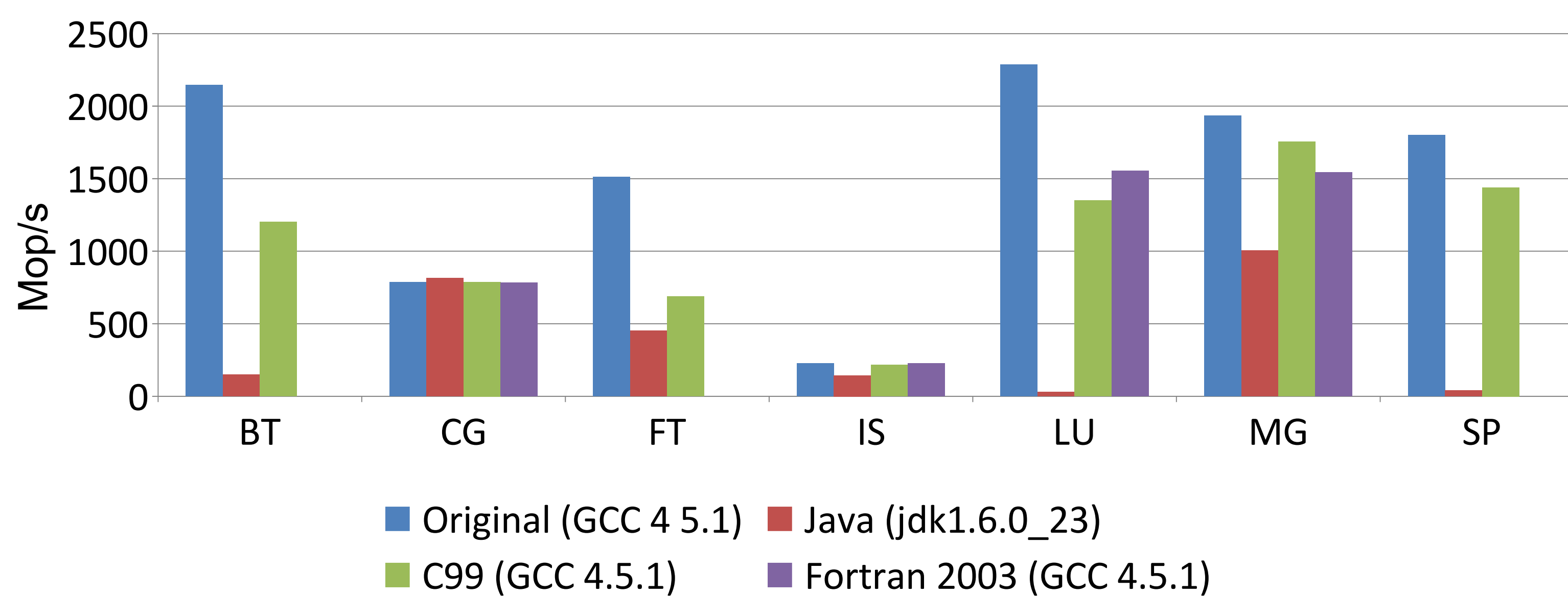


ベンチマークの構成:
NAS Parallel Benchmarks
LU, SP, BT...数値流体力学
MG...マルチグリッド法
CG...共役勾配法
FT...高速フーリエ変換
IS...整数ソート
Dhystone
整数演算性能を測る。
関数コールと文字列操作が多い。

NAS Parallel Benchmarks の規模はJava版では各プログラム数千行程度である。これは既存の言語比較ベンチマーク集より1桁大きい。DhystoneにもJava版が存在するため合わせて変換を行った。

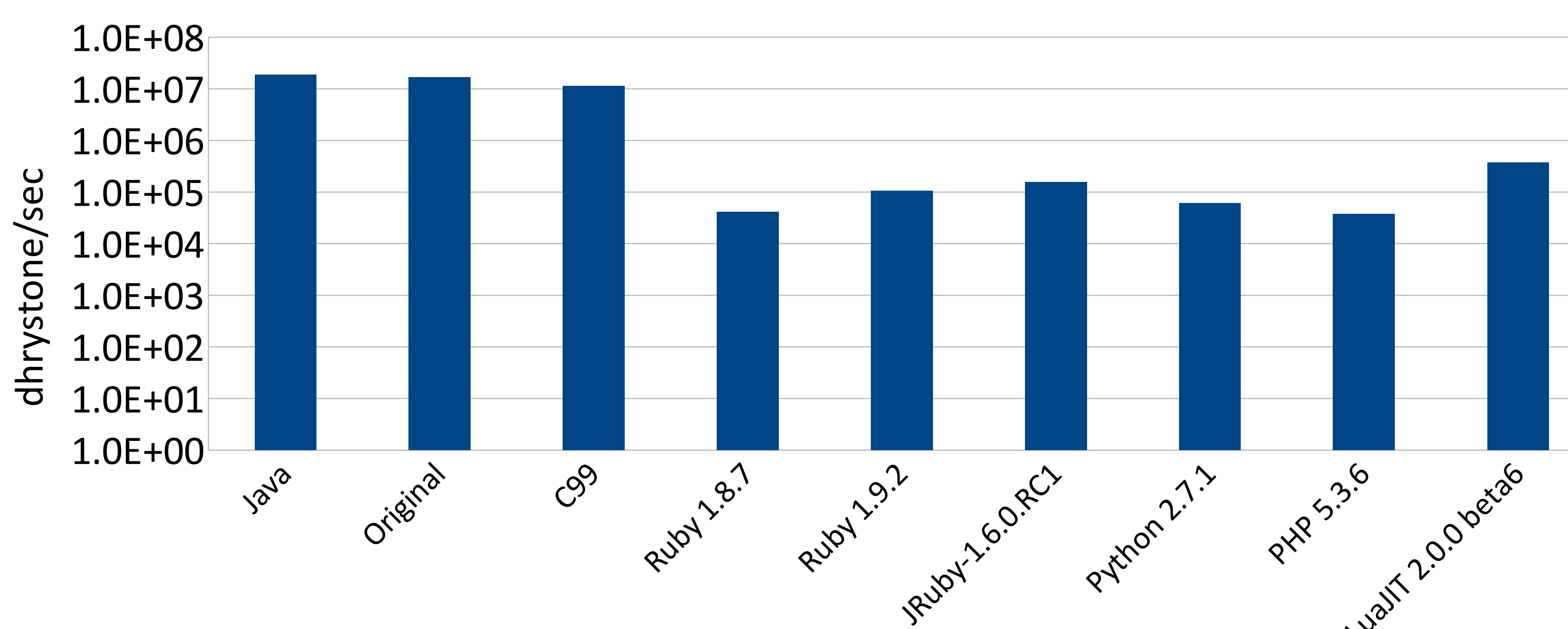
評価環境:
Dell PowerEdge R410 (Xeon E5530 2.4GHz, CentOS 5.5, メモリ12GB)

NAS Parallel Benchmarks



コンパイラ言語ではCGの性能差がほとんど見られない。Ruby 1.8.7とRuby 1.9.2の性能差は常に2倍以上ある。

Dhystone



インタプリタ言語処理系で最も速いLuaJITでも, なお2桁の性能差がコンパイラ言語と比較して見られる。

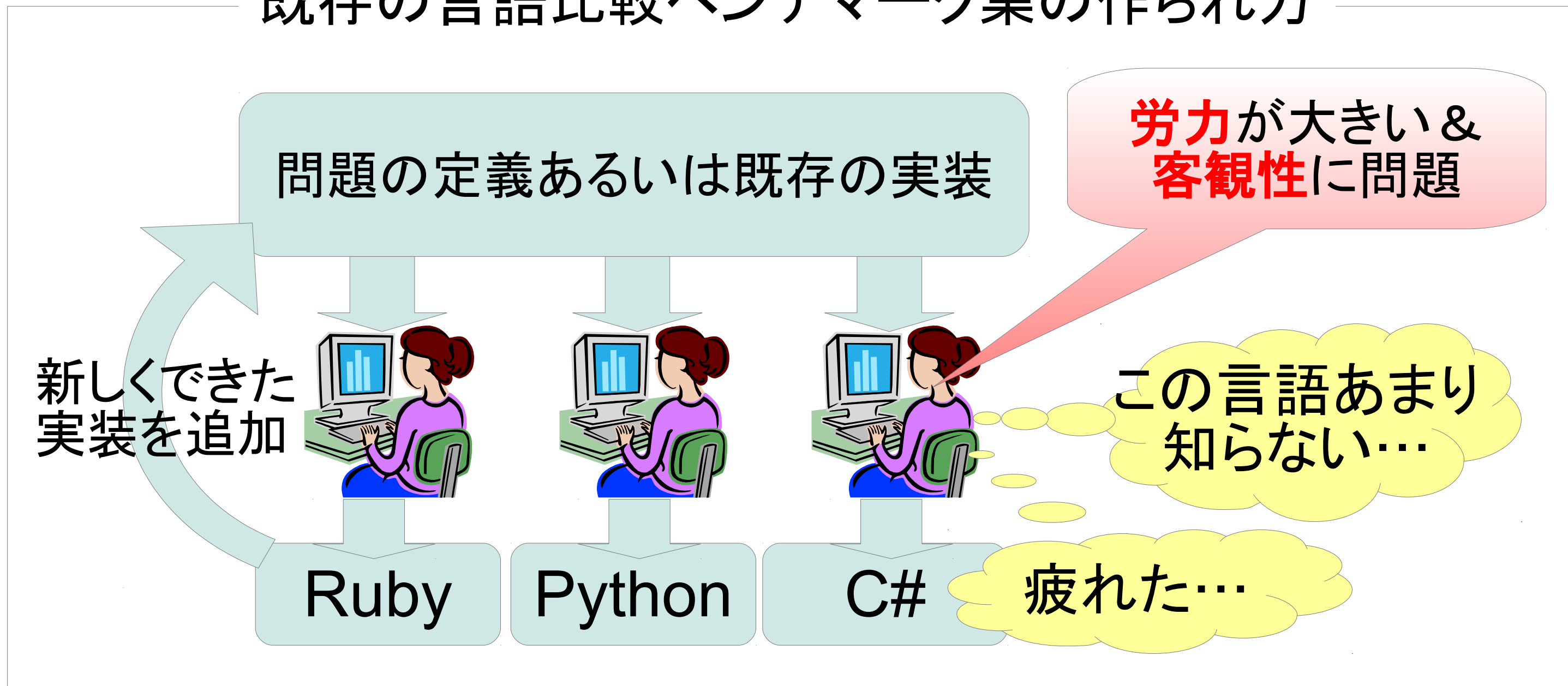
既存のベンチマーク集の問題

既存の言語比較ベンチマークは

- x 比較的小さなプログラムの集合である。
- x 人力による移植であり, 労力が大きく, かつ客観性がない。
- x SPECなどの実績のあるベンチマークと比較できない。

という問題を抱えている。

既存の言語比較ベンチマーク集の作られ方

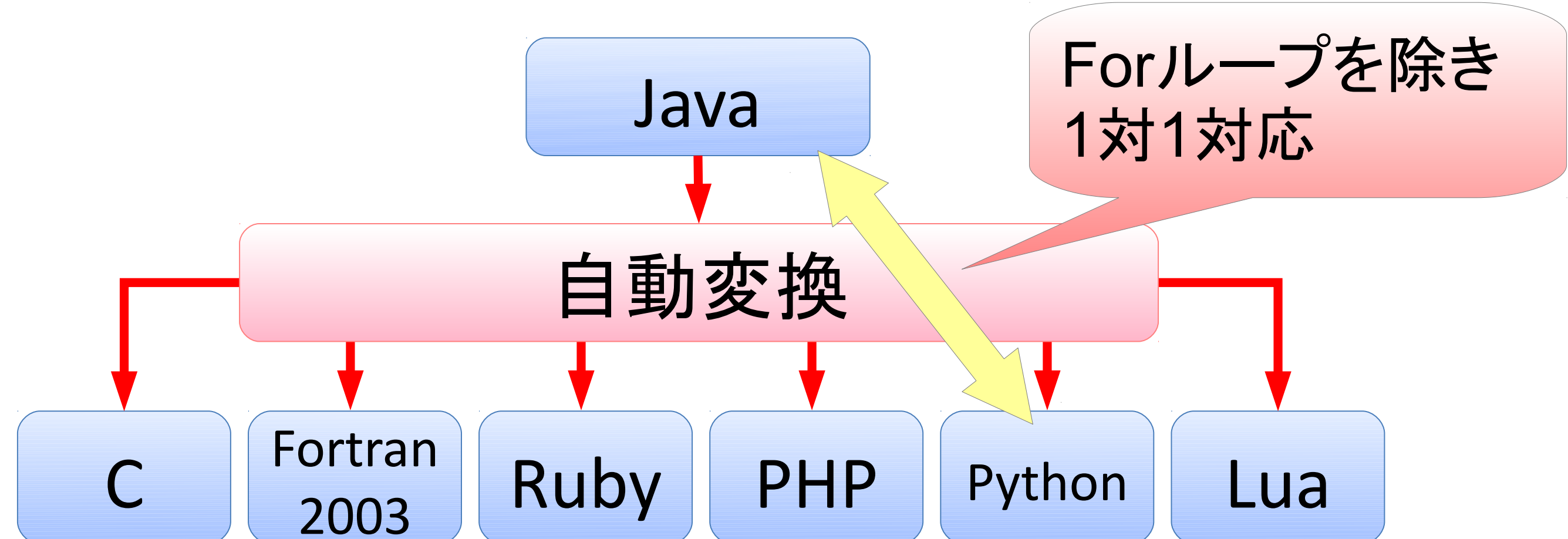


トランスレータを用いた自動変換

言語間の記述の1対1対応の取りやすい適切な変換ルールを決め, 変換の起点に適切な言語を選択して, 言語間トランスレータを構成すると

- ✓ 変換の客観性
- ✓ 労力の低減
- ✓ 既存のベンチマークとの互換性

を達成できる。また, 労力が低減されると, 大規模なベンチマークに対応しやすい。



変換の起点とする言語はJavaが最適:
•プリプロセスが存在しない
•静的型付け
•オブジェクト指向

ベンチマークはNAS Parallel Benchmarksが最適:
•Java版の存在(SPEC CPUにはない)
•適度な規模

まとめ

- 既存の言語比較ベンチマーク集は小規模であり, かつ人力による生成であった。これは労力の大きさと, 実装のムラにより言語間比較の客観性が失われるという欠点をもたらす。
- あらかじめ変換の起点に適切な言語を採用し, 1対1対応の取りやすい変換ルールを実装したトランスレータを用いてベンチマークプログラムを自動変換することにより, 従来の人力実装のベンチマーク集における弱点を克服することができる。
- 実際にトランスレータによるNAS Parallel BenchmarksとDhystoneの移植を Ruby, Python, PHP, C99, Fortran 2003, Lua に対して行うことで, 移植が従来に比べ容易となり, 多くの言語処理系同士の性能比較を, より規模を拡大し客観性を持たせて行うことができた。