

システム評価法確立 スキーマとベンチマーク・ プラットフォーム整備の現状

2013/01/21

野村@東工大

目次

- ベンチマークメタデータスキーマ
 - 我々は何を記録するのか
- ベンチマークデータ
 - どの性能指標をどう測るのか
 - その性能指標、どれぐらい信用できる?
- 今後の予定

アプリベンチマークの目的

- アプリの性能モデルを立てたときに、実際にそれが成り立っていることを確認する
 - 乖離している場合、どういう離れ方をしているか
 - モデルが不十分で考慮していないパラメータがある
 - 何らかの性能劣化要因がある
- 例: SCALE3ミニアプリのカーネル部分の性能モデル
 - $\text{FLOPS} = \text{システムのメモリバンド幅} / \text{実効B/F}$

目標関数

カタログスペック

実測 → How?

- 例えばScalasca+PAPIでメモリアクセス(最外キャッシュミス)とFP演算を計測すれば測れる

ベンチマークメタデータのスキーマ

我々は何を記録するのか(1)

- 以下の内容を記録し、データベース化する
 - 環境の基礎的なデータ
 - システム名、CPUスペック、ノード数、ノード内CPU数、メモリ容量、アクセラレータ(有無、性能)、キャッシュ階層、ディスク・ファイルシステム、ネットワーク(規格、トポロジ、バンド幅)
 - ベンチマーク実行時の実行環境データ
 - 計測日、コンパイラ、コンパイルオプション、MPIライブラリ
 - コードのチューニングの有無
 - ベンチマークの規模
 - 入力データ、ノード数、プロセス配置(ノード内、ノード間)
 - ベンチマークの結果
 - 性能値(実行時間)、Flops、実効B/F、電力

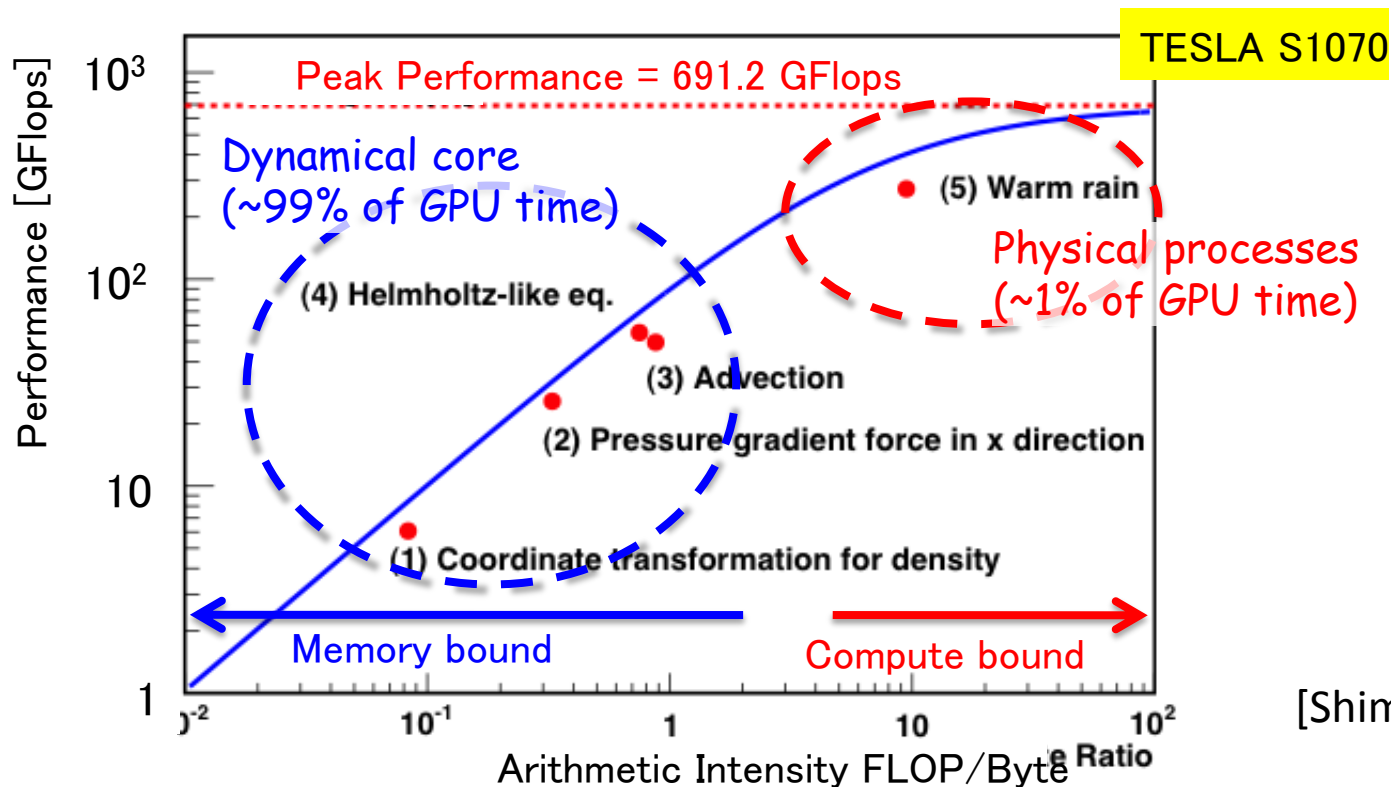
ベンチマークメタデータのスキーマ

我々は何を記録するのか(2)

- 記録は多次元のマトリックス状になる
 - マシンの差
 - アプリケーション
 - データサイズ、入力の種類
 - 並列数
 - 性能指標(Time, Flops, Power, I/O, NW...)
- 切り出してVisualizeするツールが必要
 - 任意の面で切ってCSV/Excel/Matlabを吐くような

結果をVisualizeすると何が見えるのか

- 例: Roofline Model
 - 横軸にB/F, 縦軸にFlopsのグラフ
 - メモリバウンドか、計算バウンドか



[Shimokawabe, et al]

ベンチマークデータ

どういう性能指標をどう収集するか(1)

- 性能情報はベンチマークデータとしても、アプリ最適化のための情報としても重要
- メモリアクセス
 - 字面上
 - ソースコード解析ツールが利用可能? (交渉中)
 - 実際のアドレス・アクセス数
 - PIN(佐藤先生(JAIST)のフロントエンドあり)で拾える
 - W/Rの別、命令アドレス、データアドレス
 - これを整理するとアクセスパターンやワーキングセットがわかる
 - 実際のキャッシュミス数
 - PAPIその他のカウンタから計測、B/Fはこれで論じないと無意味

ベンチマークデータ

どういう性能指標をどう収集するか(2)

- 浮動小数点演算数

- PAPIによる計測 → 信頼性はどれぐらいか?

- 前回のデモ時: あまり正確ではなかった
 - PAPIで取得できるFPのカウンタにはあまり意味のない値を出すものがある(次ページ)
 - FPカウンタに限らず、PAPIによる性能計測では、値の妥当性検証が必要

- 字面上の計測

- ソースコード解析して演算数を測るツールが存在

姫野ベンチにおける各種FPカウンタの値

～papi_native_avail イベントの採取例～

	「-02」 オプション		「-03 -xHost」 オプション	
	単精度版	倍精度版	単精度版	倍精度版
ログ-TIME[s]	14	34	14	27
ログ-MFLOPS	4,118	1,615	3,927	2,073
PAPIF_flops (MFLOPS)	18,700	18,510	18,711	18,650
プログラム中で計算される演算数(単 or 倍)	55,716,020,000	55,716,020,000	55,716,020,000	55,716,020,000
(papi_native_avail イベント)				
FP_COMP_OPS_EXE:SSE_SINGLE_PRECISION	16,742,390,000	0	17,761,350,000	0
FP_COMP_OPS_EXE:SSE_DOUBLE_PRECISION	0	35,637,460,000	0	31,867,180,000
FP_COMP_OPS_EXE:SSE_FP_PACKED	15,169,740,000	34,961,360,000	14,365,210,000	30,268,690,000
FP_COMP_OPS_EXE:SSE_FP_SCALAR	1,572,644,000	676,102,100	3,396,137,000	1,598,487,000
FP_COMP_OPS_EXE:X87	10	9	9	21
FP_COMP_OPS_EXE:MMX	0	0	0	0
FP_COMP_OPS_EXE:SSE_FP	16,741,150,000	35,633,900,000	17,756,580,000	31,857,360,000
FP_COMP_OPS_EXE:SSE2_INTEGER	7,208	20,529	12,055	17,625
SSEX_UOPS_RETIRED:PACKED_SINGLE	26,256,410,000	774,192,000	25,610,440,000	22,287,050,000
SSEX_UOPS_RETIRED:PACKED_DOUBLE	0	27,528,980,000	0	27,090,260,000
SSEX_UOPS_RETIRED:SCALAR_SINGLE	3,424,161,000	0	6,676,573,000	0
SSEX_UOPS_RETIRED:SCALAR_DOUBLE	796,772,800	48,257,970,000	385,483,200	5,017,759,000
INST_RETIRED:X87	3	3	3	3
INST_RETIRED:MMX	0	0	0	0
INST_RETIRED:ANY_P	39,685,980,000	94,328,630,000	39,573,730,000	68,784,220,000
INST_RETIRED:TOTAL_CYCLES	42,530,280,000	126,116,900,000	45,956,840,000	87,849,760,000
(papi_avail イベント)				
PAPI_FP_INS	16,742,830,000	35,650,540,000	17,762,180,000	31,866,210,000
PAPI_FP_OPS	16,742,830,000	35,650,540,000	17,762,180,000	31,866,210,000
PAPI_SP_OPS	62,246,790,000	104,842,600,000	60,847,480,000	90,719,180,000
PAPI_DP_OPS	15,168,520,000	70,571,150,000	14,362,890,000	62,077,790,000
PAPI_VEC_SP	15,175,060,000	34,956,320,000	14,364,840,000	30,255,290,000
PAPI_VEC_DP	15,175,060,000	34,956,320,000	14,364,840,000	30,255,290,000
PAPI_TOT_CYC	42,696,990,000	110,171,100,000	45,242,480,000	84,507,190,000

ベンチマークデータ どういう性能指標をどう収集するか(3)

- Network
 - Communcation Matrix (VapirTrace)
 - 通信命令数
 - 通信バイト数
 - デモ(MARBLE, 4ノード実行ですが)
- I/O
 - TAUによるI/O計測
 - POSIX I/Oについて、1回あたりのI/Oサイズ分布とI/O命令総数をファイルごとに計測可能

今後の東工大グループの予定

- 年度内:
 - 性能変数の確定・計測手法のドキュメント化
 - 頂いたアプリのTSUBAMEを用いた大規模実行ベンチマーク
 - 先週の資源量見積もりアンケートにご協力いただきありがとうございました
- 来年度
 - アプリケーションの性能モデル確立
 - ベンチマークデータリポジトリの整備
 - 任意の軸における切り出し・可視化機構の実装
 - ベンチマークデータの取得
 - OpenACC?