

Rを用いたブートストラップ法の大規模並列計算

東京工業大学・情報理工 下平英寿

東工大のスパコン TSUBAME とサブシステム TSUBASA は多数の計算ノードからなるグリッドクラスタである．TSUBAME は Opteron が 10480 コア，TSUBASA は Xeon が 720 コアよりなる．最近，それぞれ 700 コア程度を長時間占有して利用する機会があったので，その経験を紹介する．利用したアプリケーションは R のライブラリ pvclust と scaleboot であり，ともにマルチスケール・ブートストラップ法の実装である．並列計算は R のライブラリ snow と MPI により実装されている．これらのライブラリはすべて公開されており CRAN より入手可能である．snow の導入は学術国際情報センターに依頼した．

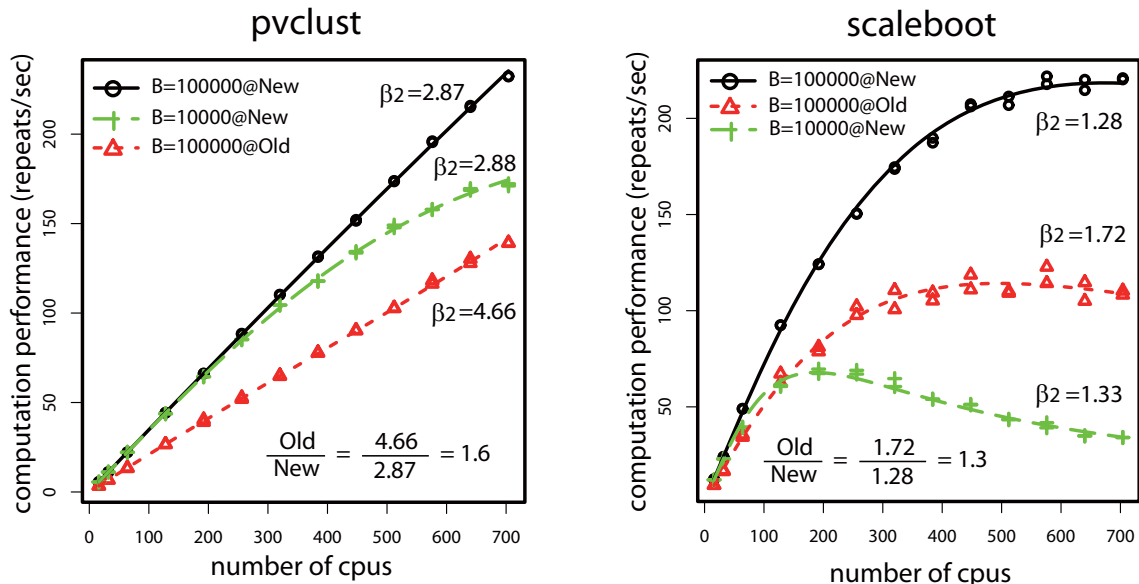
ブートストラップの反復数 B は通常 $B = 10^4$ で十分だが，計算負荷が重い状況を再現するため $B = 10^5$ も用いた（マルチスケール・ブートストラップ法のスケールは 13 個にしたので，実際の反復数は $13B$ である）．それぞれの実験条件において，コア数 x と計算時間 y 秒を測定したデータ (x_i, y_i) , $i = 1, 2, \dots, 26$ を準備した．現実を単純化したモデル

$$y = \beta_0 + \beta_1 x + \beta_2 \frac{B}{x} + \epsilon$$

を当てはめて回帰係数 $(\beta_0, \beta_1, \beta_2)$ を推定した．誤差分散が $V(\epsilon) \propto E(y)$ と仮定して， $w_i = 1/y_i$ を重みとする最小 2 乗法を用いた． $(x_i, B/y_i)$ のプロットと当てはめ曲線を図示した．

計算負荷が重い場合は 700 コアまでほぼリニアに性能が向上した．たとえば pvclust で反復数 $B = 10^5$ を指定した場合，TSUBASA で 1 コアなら 80 時間かかる計算が 704 コアなら 7 分だった．計算負荷を軽くして反復数 $B = 10^4$ にすると，通信コスト等が増える影響で計算性能の向上はリニアより若干落ちた．scaleboot では計算負荷がさらに小さいので，100 コア～200 コア程度で用いるのが妥当であろう．

新たに導入した TSUBASA と従来の TSUBAME の β_2 を比較すると，新システムのほうが 1.3～1.6 倍の計算性能であることがわかる． β_2 は，各コアにおいて 1 反復するのに要する計算時間（秒）である．



図： 並列計算のパフォーマンス測定．横軸＝コア数 (x)，縦軸＝毎秒あたりの反復数 (B/y)．New は TSUBASA(ブートストラップの反復数 $B = 10^5$ および $B = 10^4$)，Old は TSUBAME($B = 10^5$)．コア数を $x = 16, 32, 64, 128, 192, 256, 320, 384, 448, 512, 576, 640, 704$ と変化させたときの計算時間 (y 秒) を，それぞれ 2 回測定した．pvclust では lung データの階層型クラスタリング，scaleboot では mam105 データで系統樹の信頼度計算を行った．