


```

1 #pragma OAT dynamic variable (para1, para2) region start
2 #pragma OAT name funcTuning
3 #pragma OAT fitting dspline2
4 #pragma OAT varied (para1) from 8 to 96 step 8
5 #pragma OAT varied (para2) from 8 to 96 step 8
6 for(i=0;i<n;i++){
7     func(...para1,para2,...); //チューニング対象
8 }
9 #pragma OAT dynamic variable (para1, para2) region end
    
```

図2 拡張した pragma (下線部) を用いた例

能パラメータの増分値を任意に設定できる。

4. 実験

4.1 評価対象

2次元に拡張した逐次追加型推定法をキャッシュブロッキングに対して適用する。キャッシュブロッキングとは、キャッシュに収まるようデータをブロック化し、キャッシュ内のデータを再利用する手法である。ブロックサイズを推定する性能パラメータとする。本実験では、行列行列積へのキャッシュブロッキングを自動チューニングする。

4.2 実験環境

CPUはIntel Core i7-3770Kを使用し、コンパイラはgcc 4.4.7を用いた。L2キャッシュサイズは256kBである。行列は倍精度浮動小数点数による一辺が1000の正方行列を使用した。行列行列積は3重ループで構成されるが、ループ順序をijkとしたとき、 i, j を推定対象とする。 k はブロックサイズ96の固定値とした。本実験では、行列行列積の実行時に行う自動チューニングを以下の2段階に分ける。

- ① 推定範囲： $8 \leq i, j \leq 96 (i, j \bmod 8 = 0)$,
組合せ総数：144
- ② 推定範囲：①の結果 $-4 \leq i, j \leq$ ①の結果 $+4$,
組合せ総数：81

i, j ともに整数である。①で粗く推定した後に、②で詳細な推定を行う。逐次追加型と、すべての性能パラメータを調べる全探索の2種類の手法で比較すると同時に、推定した性能パラメータにより最適化された理想最適化での実行時間も計測する。

4.3 実験結果

図3は横軸に反復ループ回数を取り、各反復回に実行した行列行列積の実行時間を示している。標本点逐次追加型と全探索、全探索で推定した性能パラメータによる理想最適化とで比較した。逐次追加型は、①の推定を終えると即座に②の推定を行っている。少ないループ回数で2段階の推定を終え、反復の大半を最適化された処理で実行していることがわかる。このとき、推定した性能パラメータによる実行時間は理想最適化と2%のずれで収まっており、よい推定ができたことを示している。

図4にループ処理を終えるまでに要した時間を示

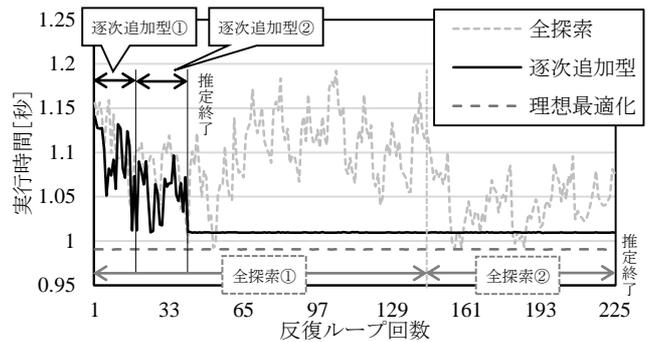


図3 反復ごとの実行時間の推移

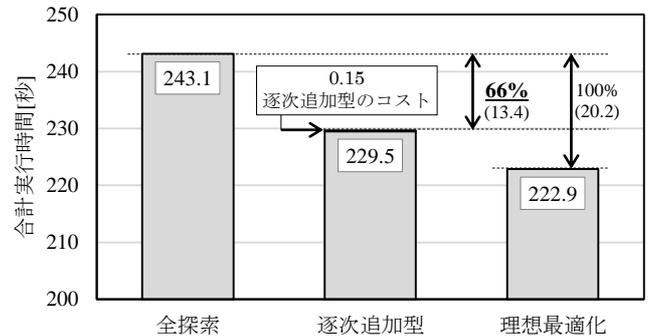


図4 合計実行時間の比較 (225反復)

す。理想最適化の実行時間に対して、逐次追加型は全探索と比較して66%短縮できた。これは、逐次追加型のコストが小さい上に推定を早く終わるが、全探索では実行時間が大きくなる性能パラメータの組合せもすべて実行するからである。

5. おわりに

複数の性能パラメータを同時に推定できる標本点逐次追加型性能パラメータ推定法を自動チューニング基盤 ppOpen-AT に実装し、評価した。少ない反復で最適値を推定し、最適化後の実行時間は十分に小さいことを示した。また、合計の実行時間が全探索と比べて削減されたことを確認した。

性能パラメータのとり得る組合せの増加やより高次元の推定では、d-Splineの更新に必要なメモリ量が増加する。そのため、より軽量な実装を開発することが今後の課題である。

参考文献

- [1] 田中輝雄, 片桐孝洋, 弓場敏嗣, ソフトウェア自動チューニングにおける標本点逐次追加型性能パラメータ推定法, 電子情報通信学会論文誌, Vol.J90-A, pp.281-291 (2007).
- [2] 片桐孝洋, 自動チューニング記述専用言語 ppOpenAT/Static の開発, 日本応用数理学会 2011 年度年会, 予稿集, pp.187-188 (2011).
- [3] T. Tanaka, R. Otsuka, A. Fujii, T. Katagiri and T. Imamura, Implementation of d-Spline-based incremental performance parameter estimation method with ppOpen-AT, Scientific Programming, Volume 22, 4, pp.299-307 (2014).