3J1-2

# マルチコアクラスタ環境での並列SATソルバの評価

## Evaluation of the Parallel SAT Solver in the Multi-Core Cluster

大村圭[*1]　上田和紀[*2]
Kei OHMURA　Kazunori UEDA

[*1]早稲田大学大学院基幹理工学研究科
Graduate School of Fundamental Science and Engineering, Waseda University

[*2]早稲田大学理工学術院
Faculty of Science and Engineering, Waseda University

This paper describes the parallelization of the sequential SAT solver MiniSat with MPI. We have implemented random search that gives a different seed to each PE and the dynamic division of search trees. In addition, we have implemented lemma sharing useful for pruning search space. The former was effective in satisfiable problems while the latter was effective in unsatisfiable problems. Our parallel MiniSat using the former method was about 30 times faster than the original MiniSat on a PC cluster with 97PEs.

## 1. はじめに

命題論理式の充足可能性問題（propositional SATisfiability problem: SAT）は，ある論理式が与えられたときに，式が真になるような変数の割り当てを求めるか，どのような割り当てをしても真にならないことを示す問題である．SAT は NP 完全という特性から多項式時間で解くことは一般にできないと考えられる．しかしソフトウェア・ハードウェア検証，AI プランニングといった様々な実アプリケーションが多項式時間で SAT に還元可能であり，SAT を高速に解く意義は大きい．また，ある問題を解くための特別なアルゴリズムを考えるより，SAT 形式に還元して SAT ソルバで解く方が容易で，かつ効率的であることが多い．

本研究では，優秀な逐次ソルバである MiniSat を MPI を用いて並列化し，クラスタ上で実行することにより高速化を実現した．探索方法としては，各 PE 毎に異なるシードを与えるランダムな探索と，探索木の動的分割を行い，各 PE 同士で探索域が重複しない探索がある．前者は SAT の問題において，後者は UNSAT の問題において高い性能向上を得ることができた．

MiniSat では探索の途中で衝突を起こした際に，その原因を解析し，learned clause（lemma）と呼ばれる clause として学習する．この特定の探索域に解がないことを示す lemma を各 PE 間で通信することにより，探索域の削減を全 PE で共有し，大幅な高速化を実現した．

## 2. SAT

SAT とはある命題論理式が充足可能（SAT) になるような変数の割り当てを求めるか，またはどのような割り当てをしても充足不可能（UNSAT) であることを示す問題である．SAT は literal を論理和でつなげた clause を論理積でつなげた CNF（Conjunctive Normal Form）を入力としており，どのような命題論理式も多項式時間内で CNF 形式に変換可能であることが証明されている．

CNF の例

式 $= (a + b')(b + c')(a + b + c)$

式において，$a = F$ と割り当てた場合，左の clause を真にするために，$b = F$ という割り当てを連鎖的に行うことができる．このような割り当て作業を BCP（Boolean Constraint Propagation）と呼ぶ．SAT の探索では，特定のヒューリスティックに従い，変数を選択し割り当てるよりも，BCP によって割り当てが決まる変数の方が圧倒的に多い．

続いて，中央の clause を真にするためには $c = F$ という割り当てをしなければならない．一方，右の clause を真にするためには $c = T$ と割り当てなければならない．このようにある変数に対して真と偽のどちらを割り当てても式が偽となってしまうことを conflict という．conflict を起こした際に，原因の解析をし lemma と呼ばれる clause として学習することにより，同じ conflict を二度と起こさないようにできる．

連絡先: 大村圭，早稲田大学大学院基幹理工学研究科情報理工学専攻，〒 169-8555 新宿区大久保 3-4-1 63 号館 5 階 02 号，ohmura(at)ueda.info.waseda.ac.jp

## 3. 並列 MiniSat の設計と実装

本節では，逐次 SAT ソルバ MiniSat の MPI による並列化の設計と実装について述べる．

### 3.1 探索アルゴリズム

並列版では各 PE はそれぞれが探索する探索域を決めたのちは，逐次版と同様には DPLL アルゴリズム [2] に従って探索を進める．SAT の変数割り当ては，2 節で述べたように，BCP により行われることが多いため，探索木の大きさを予測することは難しく，探索木を予め PE 毎に分割することはできない．本研究では，以下に挙げる二つの手法を実装し評価した．

#### 3.1.1 変数選択オーダリング変更方式

各 PE は，最初の数回ランダムシードに従って，適当な変数を選択し探索を進める．最初に選択する変数を変えるだけで，SAT の探索にかかる時間は大幅に変わることがあるため，各 PE に様々な探索木を並列に探索させる効果は大きいと考えられる．また，他 PE の探索状況に関係なく探索を進めることができるため，通信によるオーバーヘッドが起こらない．しかし，ある程度の探索域の重複は避けることができない．

#### 3.1.2 探索木の動的分割

PE 0 がある変数 $x$ に $F$ という割り当てをした場合，待機中の PE 1 に対して，$x = T$ という割り当てをさせることにより，探索木を動的に分割していき，探索域の重複が起こらないようにできる．これを実現するためには，変数割り当て，BCP による割り当て，BCP が起きた原因といった情報を待機中の PE に知らせる必要があり，通信コストが比較的大きい．そのため，本実装では，変数割り当ての情報のみを待機中の PE に渡し，

情報を受け取った PE はその情報に基づいて探索を行い，最後の割り当ての際に，別の割り当てを行うことにより，探索域が重複しないようにしている（図 1 参照）．

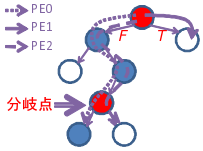


図 1: 探索のトレースによる動的分割

この方法では，通信コストは比較的小さくて済むが，探索をトレースする分，無駄な探索を行うことになる．そのため，分岐レベルを制限して，浅いレベルで分岐を行うようにしている．

### 3.2 lemma 共有

各 PE が学習した lemma を送受信することにより，探索域の削減の情報を共有し，大幅な探索域の削減を実現できる．また他 PE と同じ失敗を繰り返さないためには，頻繁に lemma を送受信しなければならないが，通信コストが膨大になってしまうため，通信のタイミングと送受信する lemma の量は制限する必要がある．本実装では，マスタ・ワーカ階層モデルとし，通信を分散することにより通信コストの削減を図った．また lemma の clause 長を制限することにより共有する lemma 量を制限している．

## 4. 評価と考察

評価に用いた並列クラスタは，GigE で接続された Intel Core2Duo 2.33GHz を搭載したマシンで構成され，主記憶容量は 1 台あたり 4 GB である．ソルバに解かせる問題として，SAT-Race 2006，SAT Competition 2007 で使用された問題を扱い，5 回実行した結果の平均を測定値とした．また，便宜的に変数選択オーダリング変更方式を v1.0，探索のトレースによる動的分割を v1.1 としている．lemma の共有は両方共に実装されている．

### 4.1 探索方法の比較

二つの探索方法を比較評価した結果について述べる．

表 1: 各ソルバが解けた問題数

| | 逐次版 | v1.0（4PEs） | v1.1（4PEs） |
|---|---|---|---|
| SAT | 25 | 31 | 32 |
| UNSAT | 35 | 38 | 40 |
| total | 60 | 69 | 72 |

表 2: 性能向上比

| | v1.0（4PEs） | v1.1（4PEs） |
|---|---|---|
| SAT | 2.85 | 2.04 |
| UNSAT | 1.32 | 1.69 |

表 1 に各ソルバが 1200 秒以内に解けた問題数，表 2 に問題ごとの性能向上比の平均値を表す．表 1 から v1.0，v1.1 は共に，逐次版より多く問題を解けていることがわかる．また，SAT も UNSAT も v1.1 がより多く問題を解くことができた．次に，表 2 から逐次版と比較して，SAT も UNSAT も性能が向上していることがわかる．特に SAT の問題に関しては v1.0 が，UNSAT の問題に関しては v1.1 が，より大きく性能向上していることがわかる．SAT の問題は解が見つかれば探索終了のため，v1.0 のように様々な探索木を他 PE に関係なく並列に探索させる効果が大きいと考えられる．一方，UNSAT の問題に関しては，全ての探索域を探索し，解がないことを示さなければならないため，ある程度探索域が重複してしまう v1.0 より，探索木を動的分割して重複が起こらない v1.1 の方が効果が大きかったと考えられる．

### 4.2 各 PEs における性能向上比

次に，PE 数を増やした際の性能向上比について述べる．v1.1 に関しては，探索木分割のために通信する情報が多いため，PE 数を増やすと通信コストが大きくなってしまい，台数効果を得ることができなかった．一方，v1.0 に関しては，通信量が比較的少なく，一定の台数効果を得ることができた．その結果を図 2 に示す．ここで，予備実験よりマスタ 1 台につきワーカ 5 台で実行した際の性能が最も高かったため，以下でも親マスタ 1 台-子マスタ n 台-ワーカ 5n 台で実行している．

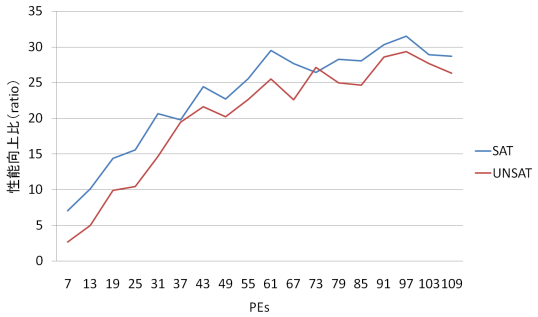


図 2: 各 PEs における性能向上比

図より、SAT も UNSAT も PE 数に比例して，性能向上比も増加していることがわかる．しかし 61PEs を超えた辺りから性能があまり向上しなくなっている．v1.0 の実装では，PE 数を増やすと，重複する探索域が増えていき，台数効果がでにくくなるためだと考えられる．また，これ以上共有する lemma 量が増えても，冗長な lemma が増加し，内部の処理が重くなっていく一方だと考えられる．

## 5. まとめと今後の課題

SAT ソルバ MiniSat を MPI を用いて並列化し，二つの探索方法と，lemma 共有により性能向上を実現した．特に，いくつかの問題においては，SAT，UNSAT ともに PE 数以上の大きな性能向上を得ることができた．また，v1.0 は SAT の問題，v1.1 は UNSAT の問題に効果があることがわかった．しかし，v1.0 に関しては，ある程度 PE 数が増えると，lemma の効果がそれほど発揮されず，性能向上に限界があった．また，v1.1 に関しては，通信コストの問題から台数効果を得ることができなかった．今後は，大規模な環境で性能向上を得るために，探索域の重複しない効率の良い探索，通信コストの削減，冗長な lemma の削減などを実現していく必要があると考えられる．

## 参考文献

[1] 大村圭, 渋谷健介, 稲垣良一, 上田和紀: ”SAT ソルバ MiniSat の並列化とそのチューニング手法”, 並列/分散/協調処理に関する『旭川』サマー・ワークショップ, 情処研報 Vol. 2007, No. 80, pp.31-36, Aug 2007.

[2] Lintao Zhang, Sharad Malik: “The Quest for Efficient Boolean Satisfiablity Solvers”, in Proc. CAV’02, LNCS 2404, Springer, pp.17-36, 2002.

[3] Niklas Eén, Niklas Sörensson: “An Extensible SAT-solver”, in SAT 2003, LNCS 2919, Springer, pp.502-518, 2004.