



田浦研紹介～霧園氣編～ + 自分の研究の紹介

B4 大貫 匠

2019-11-27

自己紹介

- 大貫 匠 (おおぬきたくみ)
- 電情 B4 (→修士も田浦研)

田浦研の雰囲気：メンバー

- 学生の人数……B4: 5人, M1: 3人, M2: 3人, D: 5人
 - 頼れる先輩が多い！
- 留学生が比較的多い？
 - ミーティングは基本 English
 - 最近は研究室内では中国語の会話が多く聞こえる気がする？

田浦研の雰囲気：環境

- 1人あたりデスク1台とディスプレイ1台を貸し出し
- **使えるもの**：技術書，スペコンアカウント，計算サーバ(GPU, Xeon Phi, …)，電子レンジ，冷蔵庫，電気ケトル，コーヒーミル，調味料，調理器具，昼寝用ベッド，ぶらさがり健康器，……
- (院生になると) ラップトップが貸与される（らしい）

田浦研の雰囲気：普段の様子

- 基本的に午前は人は少なめ、午後になると増えてくる
 - 午前にミーティングがある日はにぎやか
- 共有スペースを囲んで研究内容に関する議論が交わされることもしばしば



- コーヒー基金（有志の出資）で買ったコーヒー豆を挽いて淹れて飲む文化がある
 - コーヒーミルで豆を挽く音が日常の一部

自分の研究：背景

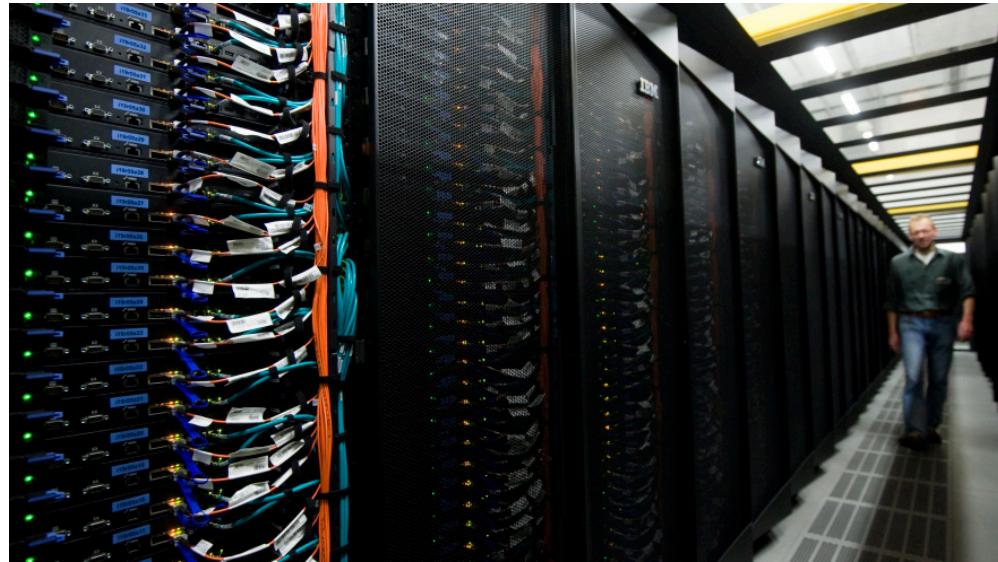
- ソフトウェアにバグがあると、困りますよね？



<https://gigazine.net/news/20140509-super-mario-minus-stage/>

自分の研究：背景

- 処理を並列化することで高速化したい
- が、並列プログラムを正しく書くのは難しい
 - 同期
 - 排他制御
 - 再現性の低さ
 -



<https://99bitcoins.com/not-even-the-top-500-supercomputers-combined-are-more-powerful-than-the-bitcoin-network/>

自分の研究：背景

```
1 class Bakery implements Lock {  
2     boolean[] flag;  
3     Label[] label;  
4     public Bakery (int n) {  
5         flag = new boolean[n];  
6         label = new Label[n];  
7         for (int i = 0; i < n; i++) {  
8             flag[i] = false; label[i] = 0;  
9         }  
10    }  
11    public void lock() {  
12        int i = ThreadID.get();  
13        flag[i] = true;  
14        label[i] = max(label[0], ..., label[n-1]) + 1;  
15        while ((∃k != i)(flag[k] && (label[k],k) << (label[i],i))) {};  
16    }  
17    public void unlock() {  
18        flag[ThreadID.get()] = false;  
19    }  
20 }
```

これが正しいことを
証明するには？

Figure 2.9 The Bakery lock algorithm.

Herlihy, M., & Shavit, N. (2011). *The art of multiprocessor programming*. Morgan Kaufmann.

自分の研究：背景

- さらにプログラム上と実際とでメモリアクセスの順番が入れ替わることがある
 - プロセッサのアウトオブオーダー実行
 - ストアバッファ
 - コンパイラの最適化
 -



自分の研究

- モデル検査という手法を用いると、あらゆる実行結果を機械的にしらみつぶしに調べることで、正しいことが確かめられる (or 正しくないことがわかる)
- 田浦研で開発されている並列ライブラリ MassiveThreads (>10000 行) をモデル検査しようとしている

おわり

- 質問があれば何でもどうぞ