

# 色々なOSSで競技プログラミング

2014-10-19

@OSC 2014 Tokyo/Fall

by @nhirokinet

# 自己紹介

- @nhirokinet (にろき)
- 一応オンラインの競技プログラミングに参加
- 兵庫出身/東京在住



# (いわゆる) 競技プログラミング

- 入力を元に計算を行うプログラムを書く。
- イメージ：
  - 蜜柑が $n$ 個ある。
  - $m$ 人の人がいる。
  - $i$  ( $0 \leq i < m$ )番の人は最低 $r[i]$ 個の蜜柑を食べたい。
  - $n, m, r$ が入力される。配布方法は何通りあるか求めよ。
  - 入力方法についての指示：（スライドの都合上省略）
  - 制限時間は2秒、制限メモリは64MB。
  - $n < 10000, m < 10000$

# 使える言語

- **TopCoder:** Java, C++, VB, C#, Python
- **CodeForces:** GNU C++ 4, GNU C++ 11, GNU C 4, MS VS C++, Free Pascal 2, Delphi 7, C# Mono 2, C# .NET, Java 6,7 Ruby, Python 2, Python 3, PHP 5, Haskell GHC 7, D, OCaml, Scala, JavaScript V8
- **AtCoder:** C/C++/C++11 (gcc), Java(OpenJDK), C#(Mono), D, PHP, Python2, Perl, Ruby, Haskell(GHC), Pascal, Javascript(Node.js), Python3, OCaml, Scheme, Scala, Bash, VB(Mono), Text(cat)

# なるほど色々なOSSがある

- 今回はこれら以外のものを使います。

# 色々な言語での競技に関する試み

- TeX [hak7a3, 2012]
- iptables [nhirokinet, 2013]
- FPGA [xiangze, 2014]
- VyOS:失敗 [nhirokinet, 2014]
  
- いずれもブログとして公開されている
  - いずれも日本語

# TeXで競技プログラミング

- [hak7a3, 2012]
- TeXのマクロを利用して計算を行う
- 余計な出力が必ず出るものの、標準入力から入力し、標準出力から出力する

# FPGAで競技プログラミング

- [xiangze, 2014]
- 必要な計算を行うモジュールを作成する
- ビット単位の演算が強力に行える



# 今回の話題

- 色々なOSSで競技プログラミングを行えるか？
- →OSSルーターで計算を行えるか？
- →iptablesによる実験

# 計算をざっくり分類

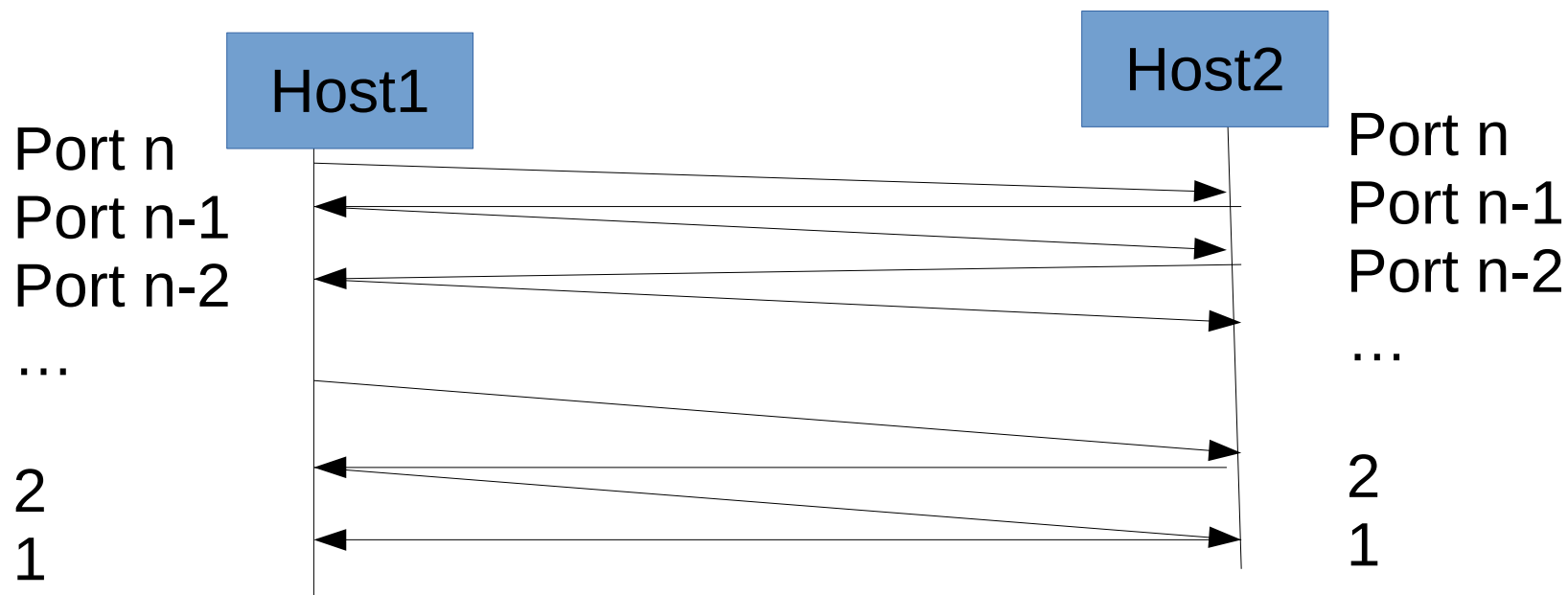
- 入力
- 計算
- 記憶
- 出力

# 計算

- 入力
  - →ポート番号
- 計算
  - →入力と出力の組み合わせ
- 記憶
  - →ポートノックに必要な機能
  - →総当り防止機能（何度も攻撃してくる人を弾く）
- 出力
  - →ポート番号

# 繰り返し

- ホスト1の*i*番ポートに来たらホスト2の(*i*-1)ポートへ
- ホスト2の*i*ポートはホスト1の*i*ポートへ
- それぞれのホストで記録は可能
- 他のホストでの処理も可能



# 関数

- 入力が65535通り
- テーブルは事前定義
- →関数そのものの挙動は答えの埋め込み
- →より複雑な関数は2入力以上として扱う

# 記録

- iptablesではstateと呼ばれるものを記憶
- 一度呼ばれたというだけでなく、一定時間に何度呼ばれたかで条件分岐可能

# 二値演算

- 一つ目の値が入力された時、その値ごとに対応するstateを記録する
- 二つ目の値が入力されると、stateとポートの対応で出力を決定する
  - 入力上限に対しても $n^2$ オーダーのルールができる

# 複雑な演算

- 一つの処理を一つのマシンで担当する
- 複雑な計算のプログラムは、それに耐えるだけのたくさんのマシンを使って表現する



行けそうな気がしてきました！

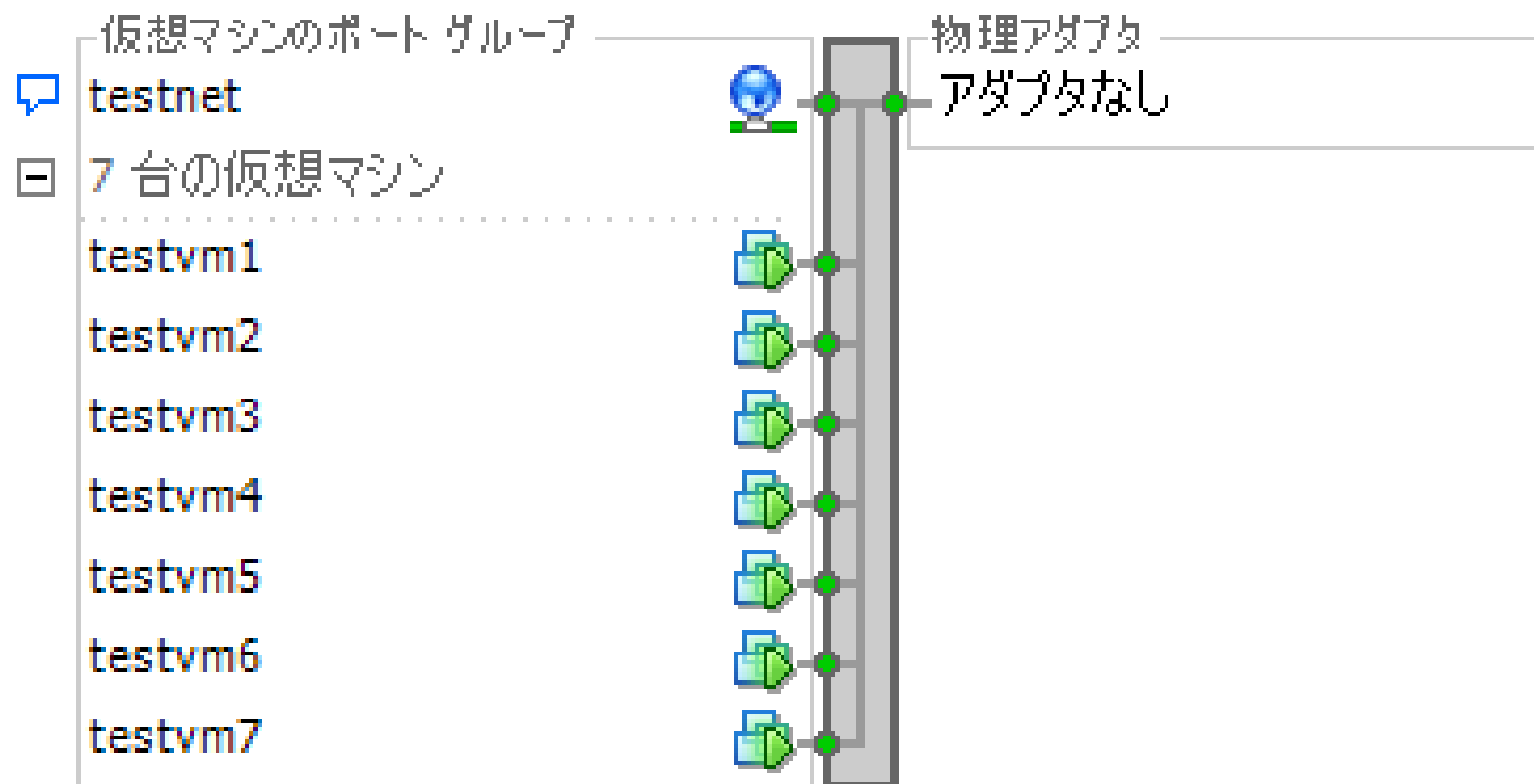
# iptablesを利用して実装

# 実験

- SRM 596 Div1 250
- 詳細は省略
- 答え
  - 入力それぞれについて以下を合計する
    - 2進表現した時立っているときのビット数の和
    - 2新表記の桁数の最大値
- この問題を実装してみた
- ただし、ひとまず入力値上限は100とした

## 標準スイッチ: vSwitch2

[削除...](#) [プロパティ...](#)



# ホスト一覧

- 1: 入出力ノード
- 2: 入力に対し、立っているビット数を計算するノード
- 3: 入力の中で最高の桁数を記録するノード
- 2: 立っているビット数をループで足し算していくノード
  - ここではn加算でn回処理しています
- 4: 2からの入力値から1を引く（ループ）ノード
- 5: 最後に2と3から結果を受け取り合計を返すノード

※同じ番号のものは同じホストで兼務

# iptablesによる結果

```
nhirokinet@testvm1:~$ nc -u 192.168.150.2 18
nhirokinet@testvm1:~$ nc -u 192.168.150.2 16
nhirokinet@testvm1:~$ nc -u 192.168.150.2 14
nhirokinet@testvm1:~$ nc -u 192.168.150.2 40000
nhirokinet@testvm1:~$ nc -u 192.168.150.3 40000
```

別窓tcpdump

```
22:53:04.278019 IP 192.168.150.1.55863 >
192.168.150.1.10: UDP, length 1
```

# iptablesによる調査

- 結論：ある程度の計算はできそう

# VyOSへの適用



# VyOSへの適用

- 問題点

# VyOSへの適用

- nat source ruleへのstateの適用ができない？
- →転送するときに情報を記録することができない？
- →情報を記録しながらループ処理することができない？

# VyOSでの計算

- 打開策見いだせず

# 今後の課題

- iptablesによるよりスケーラブルなプログラミング
- iptablesによる計算可能性の一般的な議論
- VyOSによる競技プログラミングの可能性
- その他の手法による競技プログラミングの検討

# 色々なもので競技プログラミング

- 是非お試しください

# URLs

- TeX [hak7a3, 2012]
  - <http://d.hatena.ne.jp/hak7a3/20120809/1344526951>
- iptables [nhirokinet, 2013]
  - <http://blog.nhiroki.net/2013/12/11/competitive-programming-with-router>
- 紹介したFPGA [xiangze, 2014]
  - <http://xiangze.hatenablog.com/entry/2014/02/11/020138>
- VyOS [nhirokinet, 2014]
  - <http://blog.nhiroki.net/2014/10/16/vyos-competitive-programming>