

# RCFでのプロダクションについて

**Tatsuya Chujo**

# PHENIX RCRS ファームのリソース

(2003年9月現在)

## ■ *Nodes and CPUs*

■ 0.45 GHz: 11 nodes x 2 = 22 CPUs

■ 0.8 GHz: 26 nodes x 2 = 52 CPUs

■ 1.0 GHz: 56 nodes x 2 = 112 CPUs

■ 1.4 GHz: 25 nodes x 2 = 50 CPUs

■ 2.4 GHz: 25 nodes x 2 = 50 CPUs

■ Memory: 1024 MB, 1G swap space.

■ **Total : 143 nodes, 286 CPUs** (~ RCAS)

(old: 118 nodes, 236 CPUs)

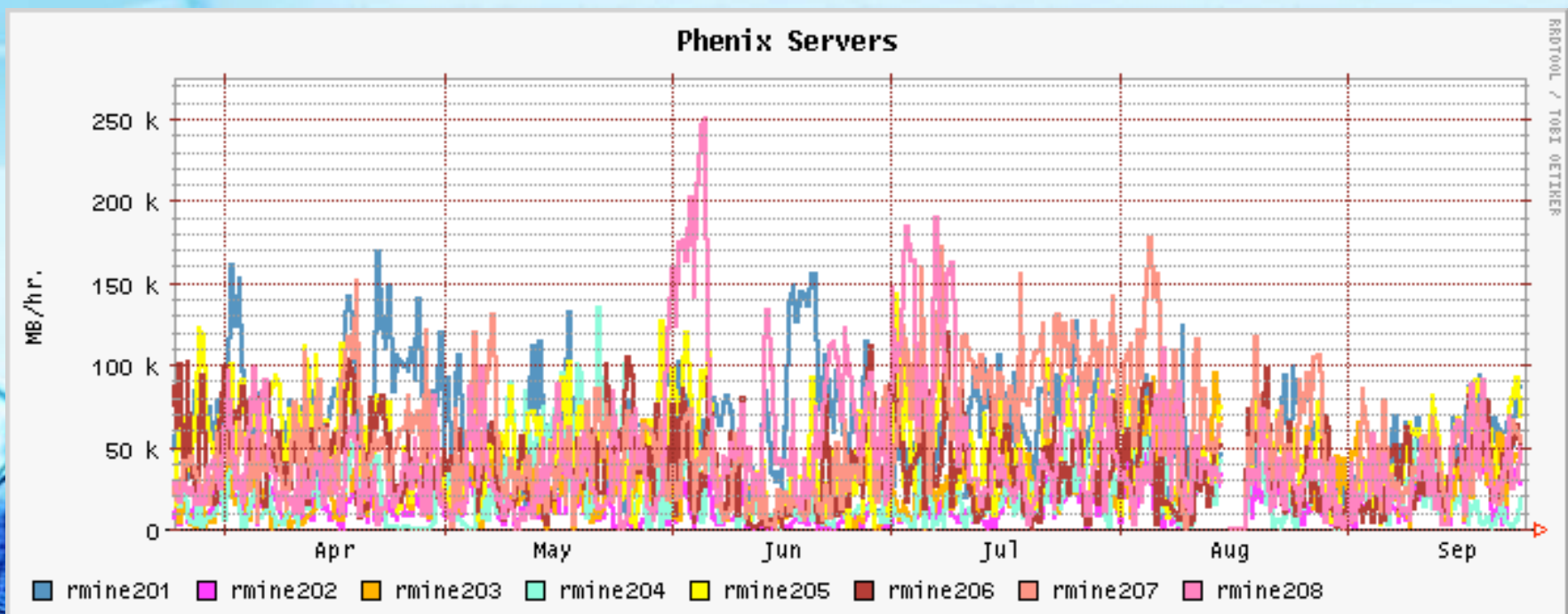
## ■ **Maximum jobs :**

■ **143 nodes x 3 = 429 jobs**

(2 executing jobs, 1 waiting job)

# 中央管理ディスクと NFS サーバ

- Production disk space
  - 30 TB (= old 21TB + new 9TB)
- NFS server: rmine201 - 208
  - 3 月ごろ 4つ新たに追加 (rmine 205 - 208)
  - 負荷バランス : アウトプットのディスク位置を考慮



# HPSS について

- DST cash disk: 1.5 TB (output file staging)
- RAW cash disk: 1.8 TB (PRDF staging)
- pftp access limit: 70 (for all 4 experiments)

# RCCRS フォームの運用

■ Not LSF batch queue, but based on “job file” per job and custom made job control scripts (made by RCF) .

■ Input/output info.

■ Macro, library version, setup environment.

■ E-mail based automated job submission.

■ Monitoring.

■ Involved Deputy DPMs.

■ Job control software (perl/tk).

■ HPSS monitor (cash, pftp via web)

■ NFS server monitoring (web).

■ Objectivity DB monitoring.

Node	Status	Job Limit	Running Jobs	Queue
rcrs2001.rcf.bnl.gov	available	2	0	1
rcrs2002.rcf.bnl.gov	available	2	0	1
rcrs2003.rcf.bnl.gov	available	2	0	1
rcrs2004.rcf.bnl.gov	available	2	0	1
rcrs2005.rcf.bnl.gov	available	2	0	1
rcrs2006.rcf.bnl.gov	available	2	0	1
rcrs2007.rcf.bnl.gov	available	2	0	1
rcrs2008.rcf.bnl.gov	available	2	0	1
rcrs2009.rcf.bnl.gov	available	2	0	1
rcrs2010.rcf.bnl.gov	available	2	0	1

Buttons: Refresh, Details, df, ls, ps, top, Mark unavailable, Mark available

Subjob ID	Status	Node	Job File
-----------	--------	------	----------

Buttons: Refresh, Details, Kill, Clean, Suspend, Resume, Help, HPSS

EXIT

# Run-2 DST プロダクション @ RCF

(2001.4 – 2002.11)

1. **Muon MDC** (muon code test for run2)
2. **Run1 v05** (run1 data, final production)
3. **Run2 v01** (run2 data, selected runs, 1st calibrations)
4. **Run2 v02** (run2 data, selected runs, 2nd calibrations)
5. **Run2 v03** (run2 data, all AuAu runs, QM02 and publication, DST→ nDST and afterburner step)
6. **Run2 pp v01 @ CCJ & RCF** (DST→ nDST)
7. **Run2 pp dimuon** (event extraction and reconstruction)
8. **Simulated Event Reconstruction.**
  - Processed 13 simulation projects at RCF.  
(PISA output production @ Vanderbilt Univ.. )
  - Embedding project.

# 典型的な問題とその解決方法

## 1. Bad Coding

### ■ メモリーリーク

- 暗黒時代：run2 AuAu v01 production 3.5 ヶ月かかった！
- “valgrind”の導入によりだいぶ改善されている。  
(Insure++ build はあまり役に立たず)

### ■ 2GB DST 問題

- DCH hit tableの変更など。

## 2. Objectivity DB

### ■ AMS too many clients と多数の DB lock

- AMS server のリスタート!
- ネットワークの問題と絡んでいる場合がある。

## 3. HPSS file transfer

### ■ キャッシュディスクの不足（特に早いジョブだがアウトプットが大きい場合、例：afterburner pass）

- キャッシュの容量を増やした。ジョブを投げる間隔を再考。

### ■ HPSS File transfer の失敗（ゼロバイト DST問題）

- ジョブの再投入。

## 4. NFS disk

### ■ NFS サーバの高い負荷（アウトプットファイルの配分を再考）

### ■ ディスクスペース

# まとめ

- 全てのプロジェクトに関してドキュメントを残すことは非常に大切。
- テストプロダクションは平均 1.5ヶ月かかる(経験則)。
- プロダクションの際に最も頻発するエラーはObjy DB とHPSS アクセスであった。
- アウトプットのディスク配分を考えることにより、NFS サーバの負荷は平均化された(Run3 から)。
- Objectivity DBはプロダクションを遅らせる最大の原因の1つに今後もなりうる。