Performance Monitoring on NERSC's POWER 5 System

Richard Gerber
NERSC User Services
RAGerber@lbl.gov

ScicomP 14
Poughkeepsie, N.Y.
May 22, 2008

Prequel

- **Although this talk centers around identifying system problems, it also highlights the stability and excellent design of the p575 POWER 5 system.**
- **Virtually all the problems we've had with Bassi can be attributed to software complexity.**
- **Take away-point: Application testing and monitoring are necessary to ensure proper system function.**

Outline

- Why monitor performance?
- What are we monitoring?
- Procedure
- Data and results
- Discussion
- Summary

Why monitor performance?

- To provide stable, consistent high performance scientific computing resources.
- To ensure that system performance and reliability never decreases over the machine's lifetime.
- To recognize when software, hardware, and configuration changes impact performance.

What are we monitoring?

- **Parallel application performance – looks at the system from a user perspective.**
- **High-level "component" tests: IO, memory bandwidth, MPI latency and bandwidth.**
- **Serial application performance on a single packed node.**
- **The goal is to monitor, maintain, improve the user experience.**

Three parallel applications

- **GTC (Gyrokinetic Toroidal Code)**
  - **Studies energy transfer via plasma microturbulence in fusion device plasmas. PIC code.**
- **PARATEC (Parallel Total Energy Code)**
  - **Ab-initio quantum total energy calculations via pseudo-potentials and plane wave basis set. Self-consistent field conjugate gradient.**
- **CAM (Community Atmospheric Model)**
  - **Complicated multi-physics and chemistry.**

Three standard parallel benchmarks

- **NPB 2.4 Class D: SP**
- **NPB 2.4 Class D: MG**
- **NPB 2.4 Class D: FT**

- **These were chosen as models of memory-intensive kernels that benefited from good memory bandwidth**

Three "Component" Tests

- **MEMRATE**
  - **Single-node memory bandwidth.**
- **MPITEST**
  - **MPI latency**
  - **MPI bandwidth**
- **PIORAW**
  - **Parallel IO performance**

Procedure

- **Goals**
  - **Monitor the system in production mode**
  - **Small impact on users**
  - **Fast and flexible way to test SW and HW changes**
  - **Automatically run suite, gather data and post results**
- **Implementation**
  - **Run suite of codes approximately weekly**
  - **Each code typically runs a few minutes on 8 (of 111) nodes**
  - **End to end workflow accommodated through scripts that run, parse, import into DB and results displayed on the web.**

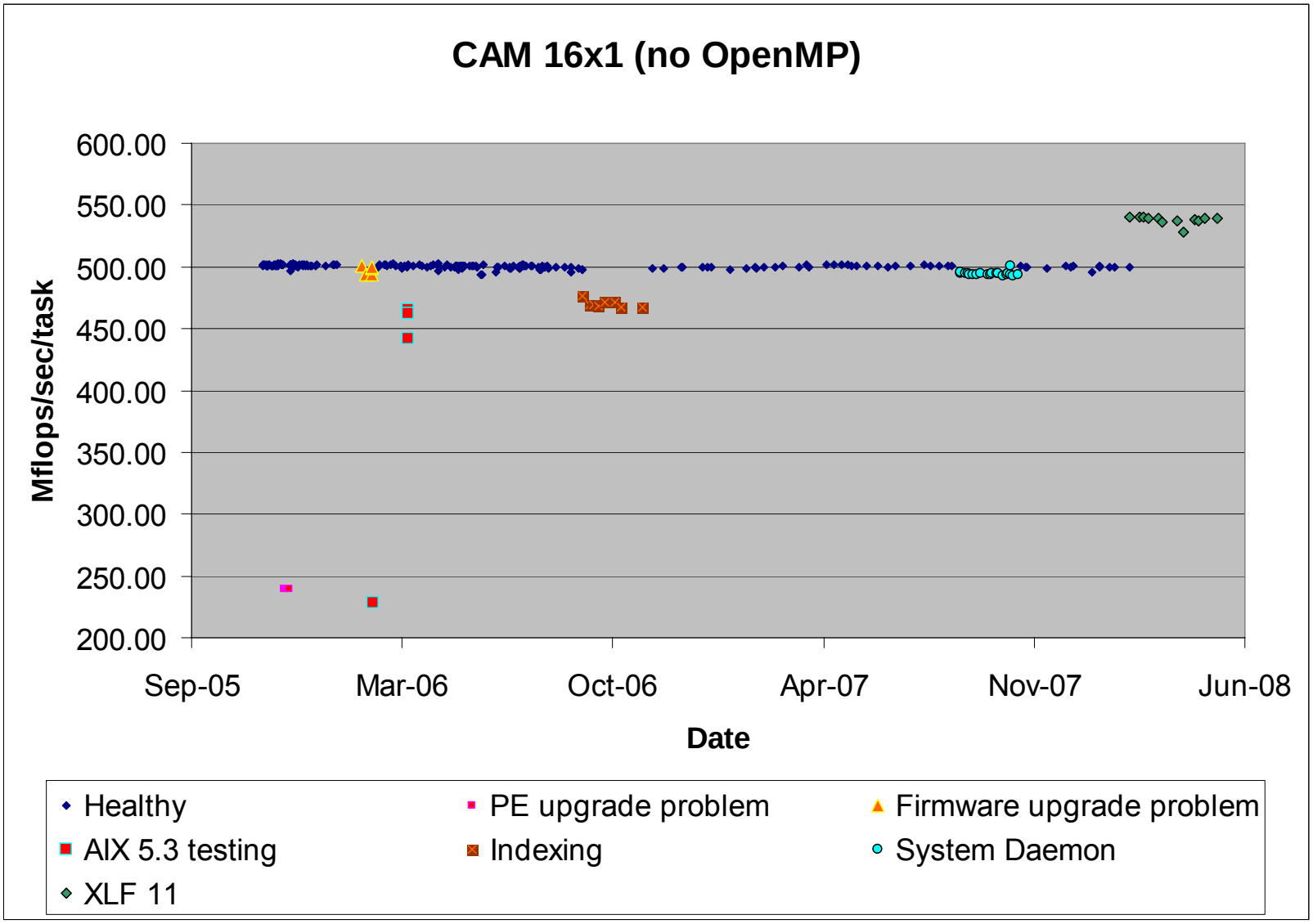
Results Web Interface

- **The monitoring results are publicly available on the web**
  - **http://www.nersc.gov/nusers/systems/bassi/monitor.php**
- **This is useful for visually scanning for anomalies.**
- **Data is quickly available to IBM management and technical staff.**

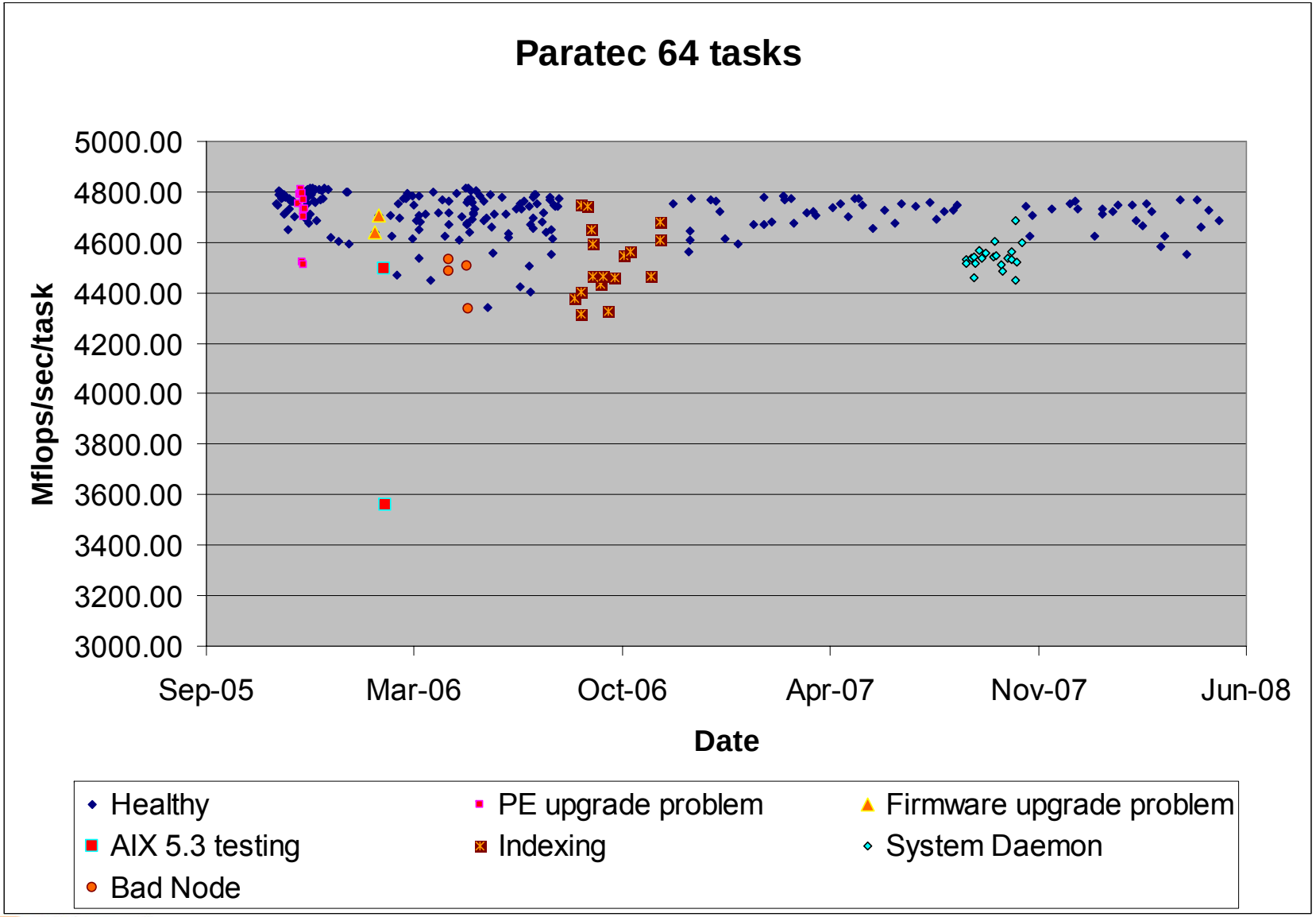
Examples of Problems Revealed

- PE upgrade problem
- HPS firmware upgrade problem
- AIX 5.3 upgrade problem
- Password file indexing problem
- System daemon problem
- Compiler upgrade issues
- Random hardware problems (e.g., L3 cache)

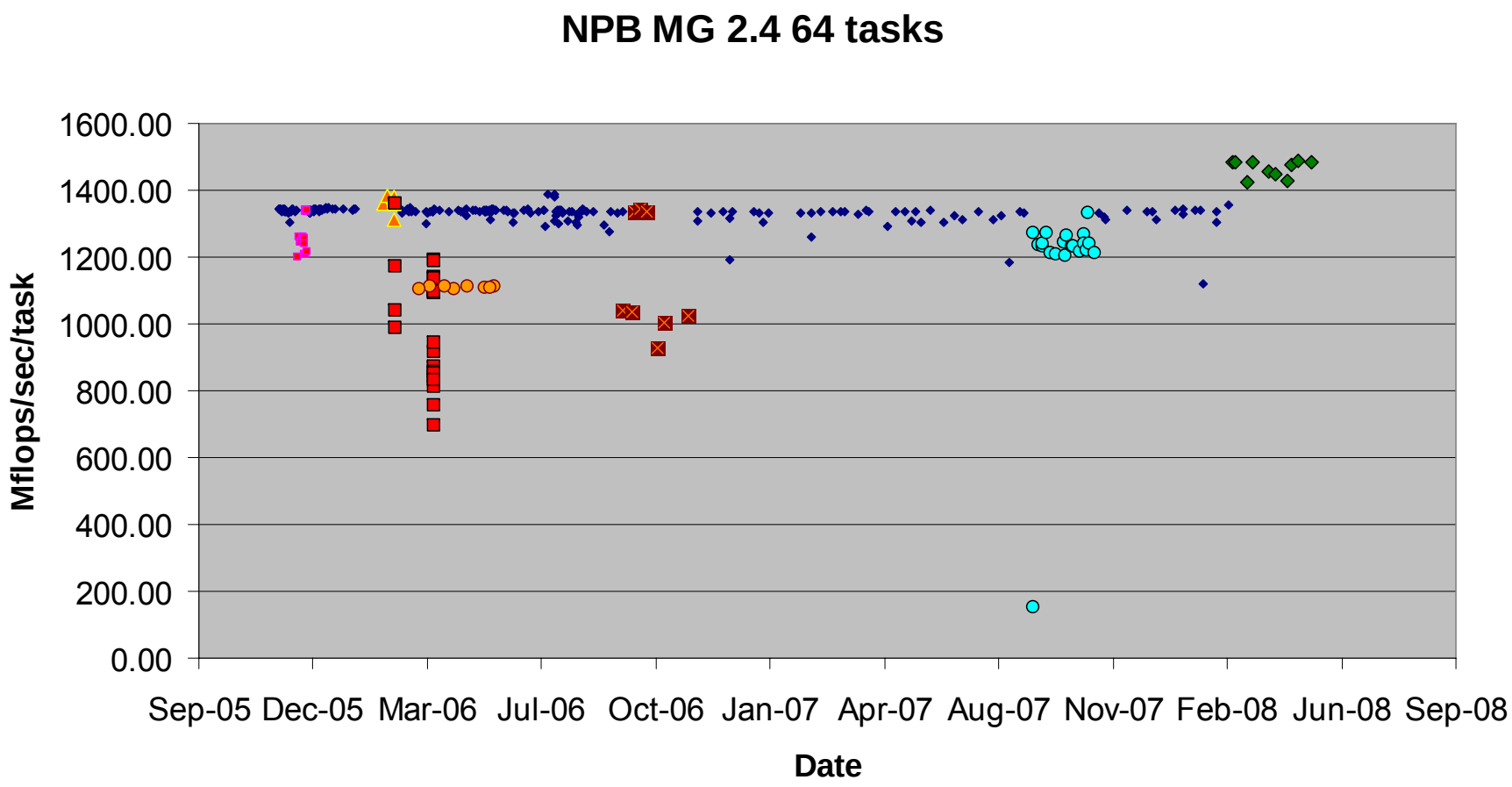
CAM 3.0 Results



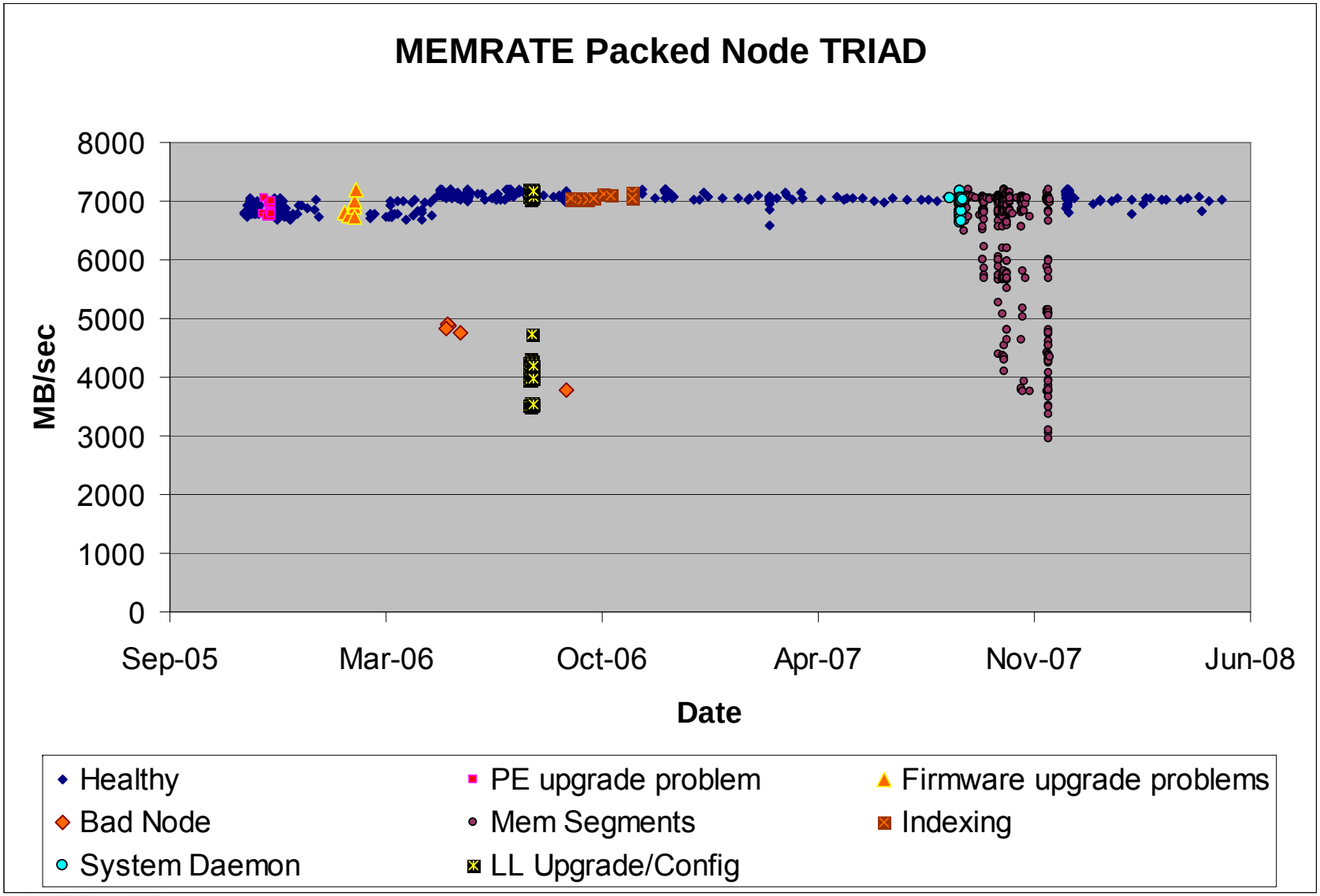
PARATEC Results



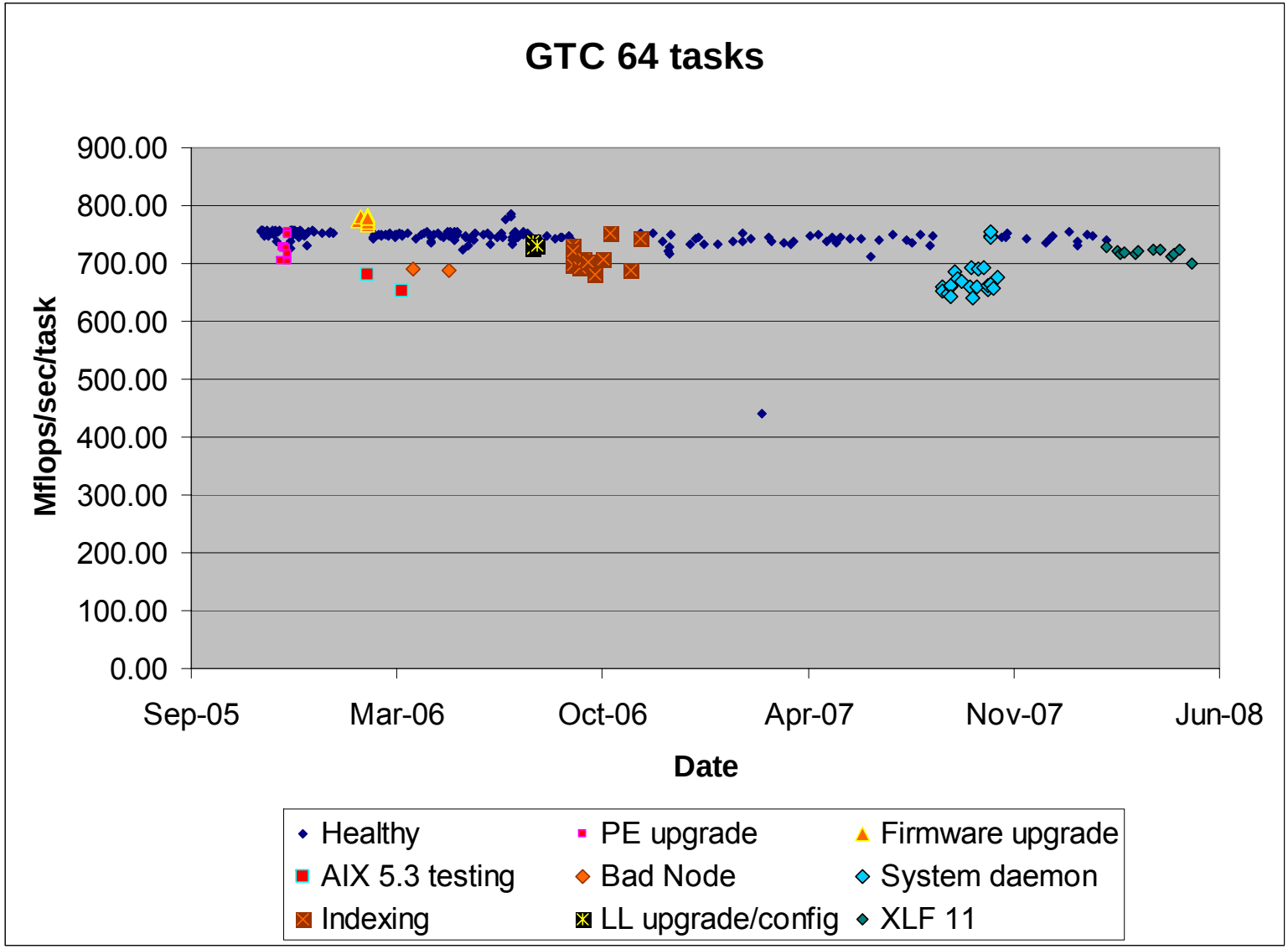
NPB MG Results



MEMRATE Packed Node



GTC Results


GTC 64 tasks
900.00
800.00
700.00
600.00
500.00
400.00
300.00
200.00
100.00
0.00
Mflops/sec/task
Sep-05
Mar-06
Oct-06
Apr-07
Nov-07
Jun-08
Date
Healthy
PE upgrade
Firmware upgrade
AIX 5.3 testing
Bad Node
System daemon
Indexing
LL upgrade/config
XLF 11

Discussion

- **Most of the problems we uncovered involved software upgrades or configuration changes**
- **None of these issues were known before monitoring revealed them**
- **None except the bad L3 caches and compiler changes were identified by "system" tests or had simple resolutions**

Healthy Results Summary

Benchmark	Avg. MFlop/s	Std. Dev.	COV
CAM 16	502	1.4	0.28%
PARATEC 64	4719	80	1.69%
GTC	747	8.9	1.20%*
NPB MG D 64	1331	26	1.95%
MEMRATE	7070	42	0.60%

Discussion

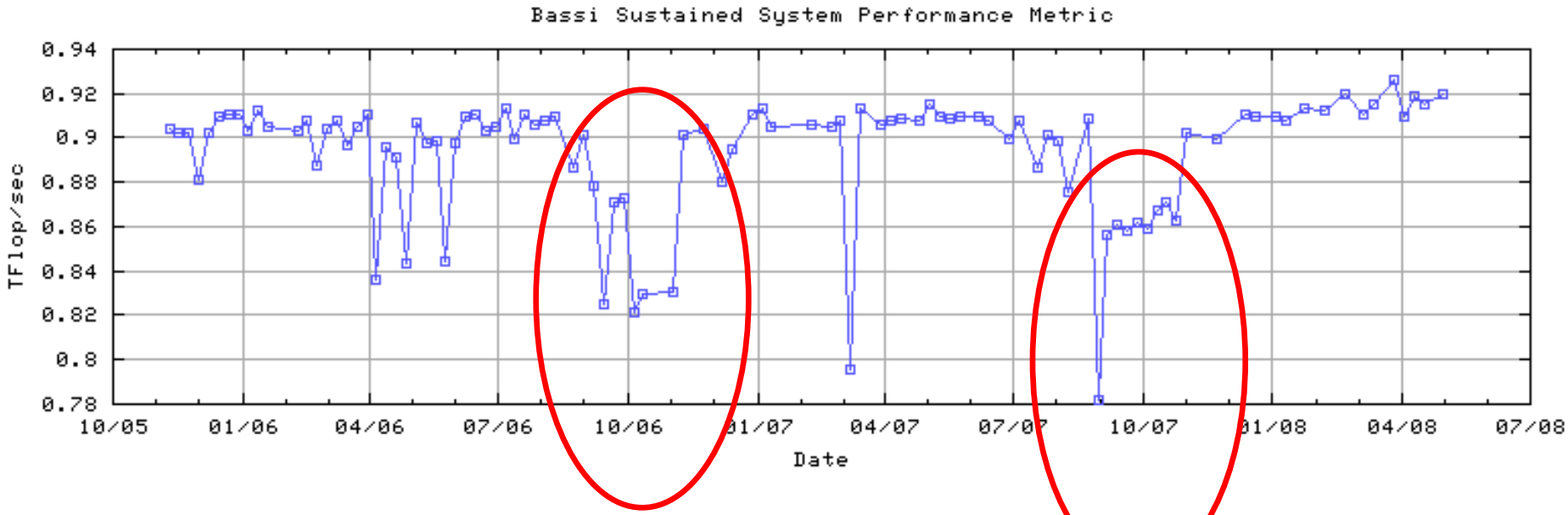
- When you run and examine the results over time you get a “feel” for when something is wrong
- The measured variation on a healthy system allows you to quickly evaluate the significance of an outlying result
- Even minor variations in run time can indicate that a system is sick
- The historical results provide quantitative evidence that a problem exists

Discussion

- A healthy Bassi has remarkably consistent run time performance characteristics.
- This is especially notable because the system is heavily loaded with a diverse workload.
- We don't have comparable long-term data for other systems at NERSC (yet!)
- Aggregate measures (SSP) are also useful, but are not as diagnostic for as many problems

Sustained System Performance (SSP)

The SSP is an aggregate metric derived from a (geometric) average of application benchmarks and standard parallel benchmarks.



Indexing

System Daemon

Summary

- **Application testing and monitoring are necessary to ensure proper system function.**
- **When configured properly application performance on Bassi is remarkably consistent.**
- **Most problems we've had with Bassi can be attributed to software complexity.**