

# PerfMiner at KTH PDC

- As everyone else we were in need of more information on all aspects of application performance. Both for optimizing present infrastructure and to make decision for new procurements
- The same problem over and over: utter lack of detail
  - Batch logs, SuperMon, CluMon, Ganglia, Nagios, PCP, NWPerf
  - Vendor specific monitoring software...
  - Only NCSA's internal system (from Rick Kufirin) met our needs. But not public!



Paralleldatorcentrum

*Daniel Ahlin  
Phil Mucci  
Lars Malinowsky  
Per Öster  
Lennart Johnsson*

<http://perfminer.pdc.kth.se>



Gelato ICE  
San Jose, April 2007

# PerfMiner: Bottom Up Performance Monitoring

- Allow performance characterization of all aspects of a technical compute centre:
  - Application Performance
  - Workload Characterization
  - System Performance
  - Resource Utilization
- Provide users, managers and administrators with a quick & easy way to track/visualize performance of jobs/system
- Full transparent integration from batch system to database to web interface



Paralleldatorcentrum

# Three Audiences

- Users: Integrating Performance into the Software Development Life-cycle
  - Quick and elegant way to obtain and maintain standardized perf. information about one's jobs.
- Administrators: Performance Focused System Administration
  - Efficient use of HW, SW and personnel
- Managers: Characterization of True Usage
  - Purchase of a new compute resource



Paralleldatorcentrum

# Site Wide Performance Monitoring

- Integrate complete job monitoring in the batch system itself.
- Track every cluster, group, user, job, node all the way down to individual threads
- Zero overhead monitoring, no source code modifications.
- Near 100% accuracy

# Batch System Integration

- PDC runs a heavily modified version of the Easy scheduler. (ANL)
  - Reservation system that twiddles node local /etc/passwd.
  - Multiple points of entry to the compute nodes
    - Kerberos authentication
    - Monitoring must catch all forms of usage.
    - MPI, Interactive, Serial, rsh, etc...
- Need to run a shell script before and after every job.
- We must use /etc/passwd as the entry point!
  - Custom wrapper that runs a prologue and execs the real shell.
  - The prologue sets up data staging area and monitoring infrastructure
- Batch system runs the epilogue.
- Data is dumped into a job specific directory
- Data about the batch system and job are collected into a METADATA file.

METADATA File

```
JOBID:111714450953
CLUSTER:j-pop
USER:lama
CHARGE:ta.lama
ACCEPTTIME:110070286
PROCS:4
FINALTIME:1100703103
```



Paralleldatorcentrum

# Data Collection with PAPIEX

- PapiEx: a command line tool that collects performance metrics along with PAPI data for each thread and process of an application
  - No recompilation required
- Based on PAPI and Monitor libraries
- Uses library preloading to insert shared libraries before the applications. (via Monitor)
  - Does not work on statically linked or SUID binaries



Paralleldatorcentrum

## Some PAPIEX Features

- Automatically detects multi-threaded executables
- Supports PAPI counter multiplexing; use more counters than available hardware provides
- Full memory usage information
- Simple instrumentation API
  - Called PapiEx Calipers

# Monitor

- Generic Linux library for preloading and catching important events
  - Process/Thread creation, destruction
  - fork/exec/dlopen
  - exit/\_exit/Exit/abort/assert
  - User can easily add any number of wrappers
- Weak symbols allow transparent implementations of dependent tool libraries



Paralleldatorcentrum

## The Back End

- After termination of every thread, PapiEX writes a file (Max 2K in length)
- Data is consumed by an offline process that imports the data to the database and archives the original data on secondary storage.

# PapiEX Sample Output

**PapiEx Version:** 0.99rc2 **Executable:**  
/afs/pdc.kth.se/home/m/mucci/summer/a.out

**Processor:** Itanium 2  
**Clockrate:** 900.000000  
**Parent Process ID:** 8632  
**Process ID:** 8633  
**Hostname:** h05n05.pdc.kth.se  
**Options:** MEMORY  
**Start:** Wed Aug 24 14:34:18  
2005  
**Finish:** Wed Aug 24 14:34:19  
2005  
**Domain:** User Real usecs:  
1077497  
**Real cycles:** 969742309  
**Proc usecs:** 970144  
**Proc cycles:** 873129600  
**PAPI\_TOT\_CYC:** 850136123  
**PAPI\_FP\_OPS:** 40001767  
**Mem Size:** 4064

**Mem Resident:** 2000  
**Mem Shared:** 1504  
**Mem Text:** 16  
**Mem Library:** 2992  
**Mem Heap:** 576  
**Mem Locked:** 0  
**Mem Stack:** 32  
**Event descriptions:**  
**Event:** PAPI\_TOT\_CYC  
**Derived:** No  
**Short Description:** Total cycles  
**Long Description:** Total cycles  
**Developer's Notes:**  
**Event:** PAPI\_FP\_OPS  
**Derived:** No  
**Short Description:** FP operations  
**Long Description:** Floating point operations  
**Developer's Notes:**



Paralleldatorcentrum

# Scalable Data Base Design

- Now in version 2, implemented in Postgres.
  - Portable to other back ends.
  - May contain many millions of rows for a production system.
  - Population by the epilogue is done through Perl scripts and DBI.
  - All DB structure contained in the scripts.
  - No external schemas or DB setup required.
- 5 keys: cluster, job-id, host-id, process-id, thread-id
- First version was implemented with a base table of 'standard' metrics and individual tables for specific metrics.
  - Version 2 has a separate table for every metric.
  - Each table has a scope (or is a node in an ontology).
- Direct measurements.
  - Events that are measured directly by the underlying performance tool (and METADATA).
- Derived measurements:
  - Events that are explicitly constructed from complex queries.
  - SQL for constructing them is embedded in the database. Measurements can be hidden as VIEWS. Rates, Ratios, etc.



Paralleldatorcentrum

# PerfMiner Interface

- Straight HTML fed by PHP scripts.
  - JpGraph/GD and PHP DBI.
- Proof of concept interface: more work to do.
- three selection criteria:
  - What event to visualize?
  - What range/scope to select?
  - What range/scope to display over?



Paralleldatorcentrum

<b>Cluster</b>	<b>Charge Group</b>	<b>User</b>	<b>Job ID</b> <input type="checkbox"/>
ALL Lucidor LinuxLab	ALL 003-04-16 003-04-19 003-04-90 003-04-94 021-03-21 free local2004.mik proj.tau staff	ALL elenius f97-mal jennie lama liuyq mucci peterbr smeds ulfa	ALL Lucidor-012714241553 Lucidor-012719302345 Lucidor-050114581564 Lucidor-051912275455 Lucidor-052012143260 Lucidor-052012143408 Lucidor-052012143503 Lucidor-052012143600 ... (139 more)
<b>First Metric</b>	<b>Second Metric</b>	<b>Output type</b>	<b>Plot type</b>
Level 1 data cache accesses	Level 1 data cache misses	User	Scatter plot
<input type="button" value="Rock on!"/>			

