Performance Monitoring/Analysis of Overall Job Mix on Large-Scale Pentium and Itanium Linux Clusters

Rick Kufrin

Scientific Computing Division

National Center for Supercomputing Applications

University of Illinois at Urbana-Champaign

rkufrin@ncsa.uiuc.edu

SIAM PP-2004
February 26, 2004
San Francisco, CA

These Talks

http://www.cs.utk.edu/~mucci/latest/mucci_talks.html

Outline of Talk

- **NCSA computing environment**
  - Hardware, software… *and people*
- **Project background and motivation**
- **Implementation details**
- **Experiences *(subtitle: the cycle of problems and solutions)***
- **What was learned**
- **Future plans**

NCSA Linux Clusters

- NCSA's (recent) cluster history began with "NT supercluster"
- NCSA Linux clusters ('00/'01):
  - ~512 dual processor 1 GHz Pentium III
  - ~160 dual processor 800 MHz Itanium
- "TeraGrid" Itanium 2 cluster in production 1/2004
- 3 GHz Xeon cluster on the horizon

Software Environment

- RedHat Linux used in production from outset
  - RH 7.1 (more recently 7.2 on PIII)
  - Kernel support for performance is critical
- TeraGrid uses SuSE distribution
- Xeon cluster: RH 9.0
- MPI support: NCSA Virtual Machine Interface (VMI), version 1.0
  - TG/Xeon: MPICH-GM, ChaMPIon/Pro

Project Motivation

- **NCSA transition (c. 2000) from shared-memory "traditional" supercomputers to cluster technology is a major shift:**
  - Does it translate *in practice* to high-performance cycles **delivered**?
  - What is the percentage of users making efficient use of the resource?
  - How can knowledge improve services (i.e., feedback loop)?

Project Requirements

- **Initial project definition (Jan 2003):**
  - Measure the aggregate performance of all user applications on Linux clusters, (new) IBM p690, and (retiring) Origin 2000 systems
  - Unmodified binaries – no impact on or effort required of users
  - Operational within existing job management system – no “special queues” or contacts. Avoid self-selecting users.
  - In-place and operational by March ’03 in order to gather sufficient data for NSF reporting by late summer.

Project Implementation

- **Requirement for non-Linux systems soon dropped**
  - IA-32 and IA-64 systems remained
- **Existing performance measurement software enumerated and options narrowed, typically due to:**
  - Not available on both architectures
  - Not production ready status
  - Uncertain support
  - User intervention required or measured “wrong” thing

PerfSuite “psrun”

- **NCSA-internal project for performance analysis using PAPI**
  - Tentative development started in Jan 2002, development accelerated after:
    - Experience gained with then-existing Linux performance tools with hardware counter support
    - Users feedback and experiences
    - Discussions/observations at PTools 2002 meeting

(…of course, development *really* accelerated after this project was formed)

Some Key Features of psrun

- Hardware performance counting and profiling with unmodified executables
  - Uses library preloading available on Linux
- Performance counter multiplexing
- POSIX thread support
- Input and output standardized on XML
- IA-32 and IA-64 support
- MPI interoperability

(almost) all of the above are made possible through PAPI

The Switch is Flipped – How Did It Go?

- We succeeded in breaking users' codes
- Substantial confusion/uncertainty regarding sources of errors
- Support staff not sufficiently armed with information to help diagnose properly
- Weaknesses exposed in virtually all layers of the software: PerfSuite, PAPI, VMI, batch systems, Linux, …
- *But strengths and good design in each layer were shown too.*

What Doesn’t Work (yet…)

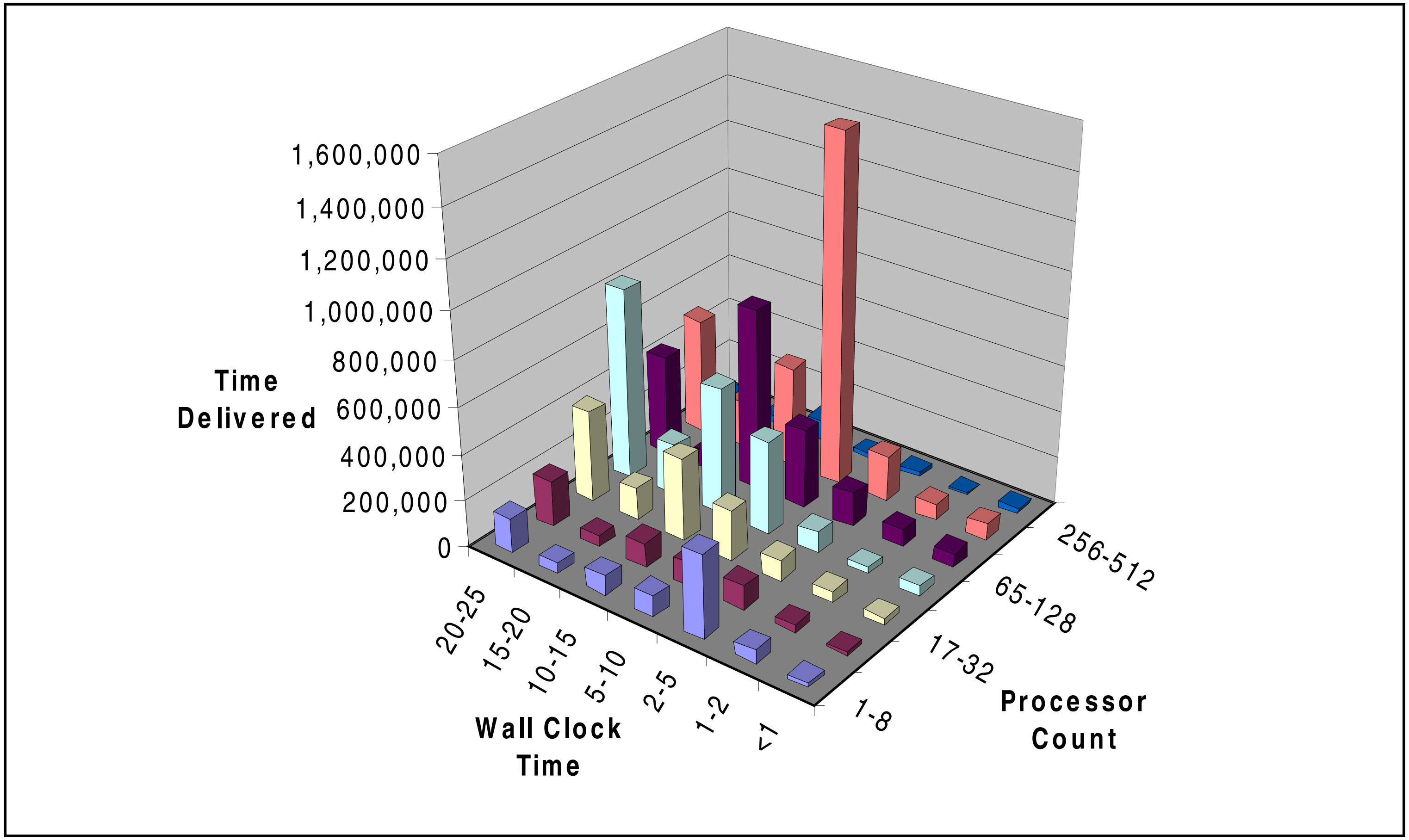
- Restricted to single point-of-launch applications (i.e., “mpirun a.out”)
- Multiplexing still too fragile to rely on for 24/7 use
- Child process monitoring not sufficiently tested – this is an issue for proposed TeraGrid “cross-site” run model with VMI-2
- RedHat 9 / Xeon/ MPICH-GM / psrun not stable for production. Alternative developed using x86 perfctr driver directly.

Notes About Collection Process

- Automatic performance collection is one piece of the picture in the pilot part of the project (2003)
- Characterizing *how the clusters were used* during this period requires information from multiple sources, collection and tracking mechanisms
- Compiler versions during this period (Intel) were primarily 6.x, some 7.x

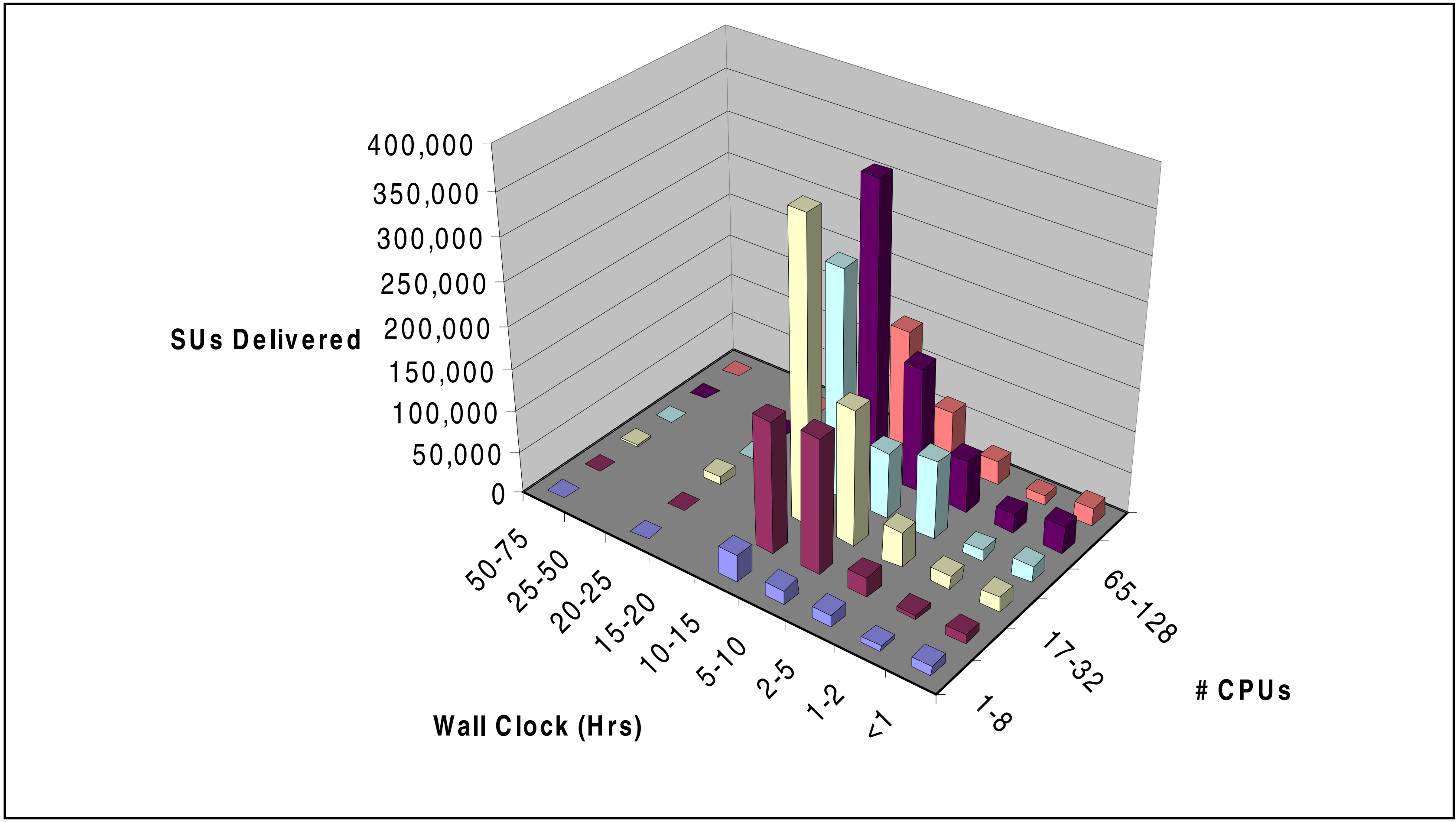
Job Scale (time, processors)

Platinum (IA-32) FY03



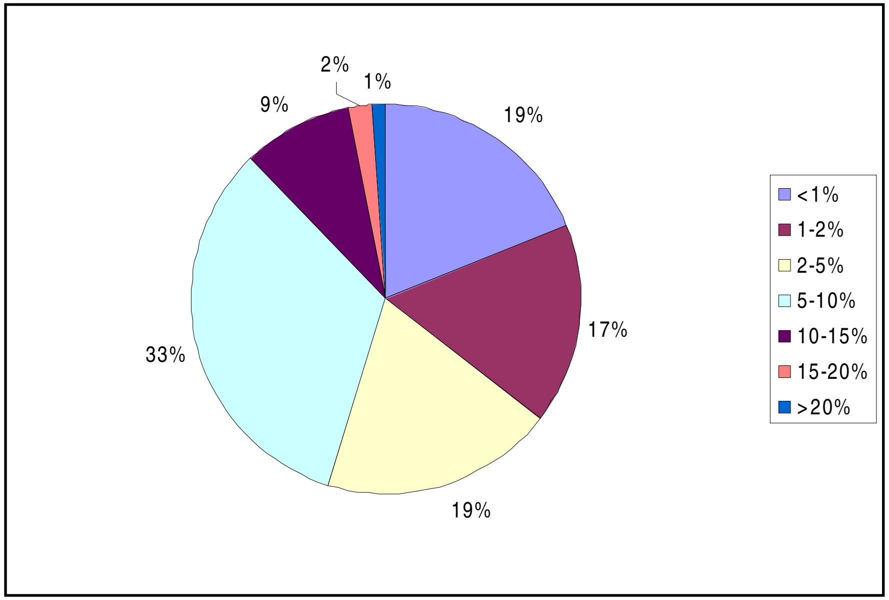
Job Scale (time, processors)

Titan (IA-64) FY03

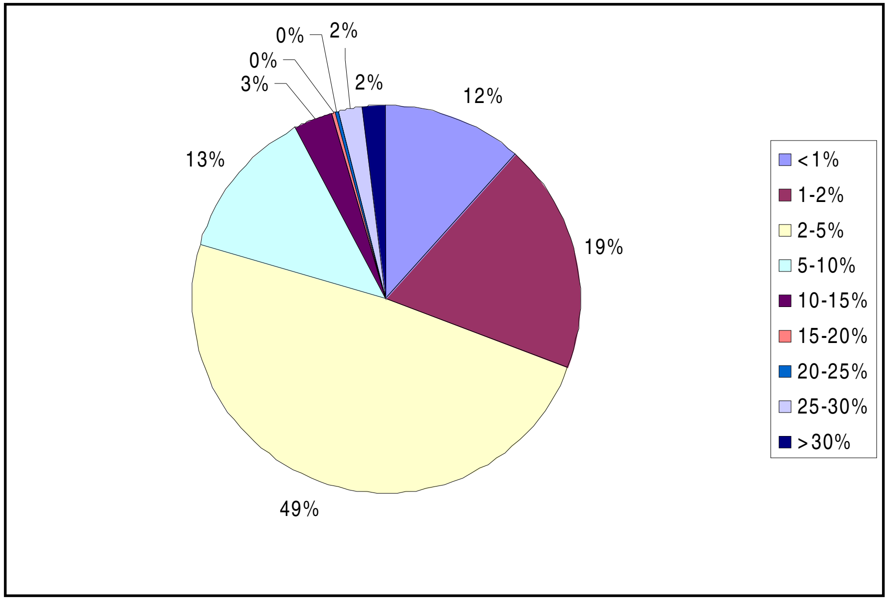


% Peak FP Performance

Platinum IA-32



Titan IA-64



- 10% of peak or greater: 12% on Pentium III, 7% on Itanium
- Note: vector/SIMD instructions not counted as FP_INS/FP_OPS by PAPI

Are Performance Counters Enough?

- **Performance counters provide valuable information required for an analysis like this, but:**
  - They only provide a CPU-centric view
  - They are not directly comparable across architectures (but we try…)
  - There is no single metric suitable for determining whether an arbitrary application is making "good use" of a machine
  - And…

Some Futures

- 2003 implementation and experiences laid the groundwork for ongoing effort
- Multiplexing is the single most important item on the “wish list”
- Similar work for other architectures at NCSA
- Possible “opening up” of data collected for use by other researchers in performance, but there are many issues to be resolved
- Better incorporation of non-performance counter data with other data sources
- Despite “automatic” collection, this is currently still a very labor-intensive analysis!

Conclusion

- Proof-of-concept project, lots of early pitfalls
- IA-64 hardware first-generation (very early compilers too)
- Linux clustering for scientific apps may be “cheap”, but there is much room for performance tuning and improvement
- Including other platforms in study would increase value substantially, provide more balanced view