Reflections on Fifty Years of Performance Analysis

Ken McDonell

Computing Landscape circa 1969

- **Machines lived in warehouse sized computer rooms**

Computing Landscape circa 1969

Computing Landscape circa 1969

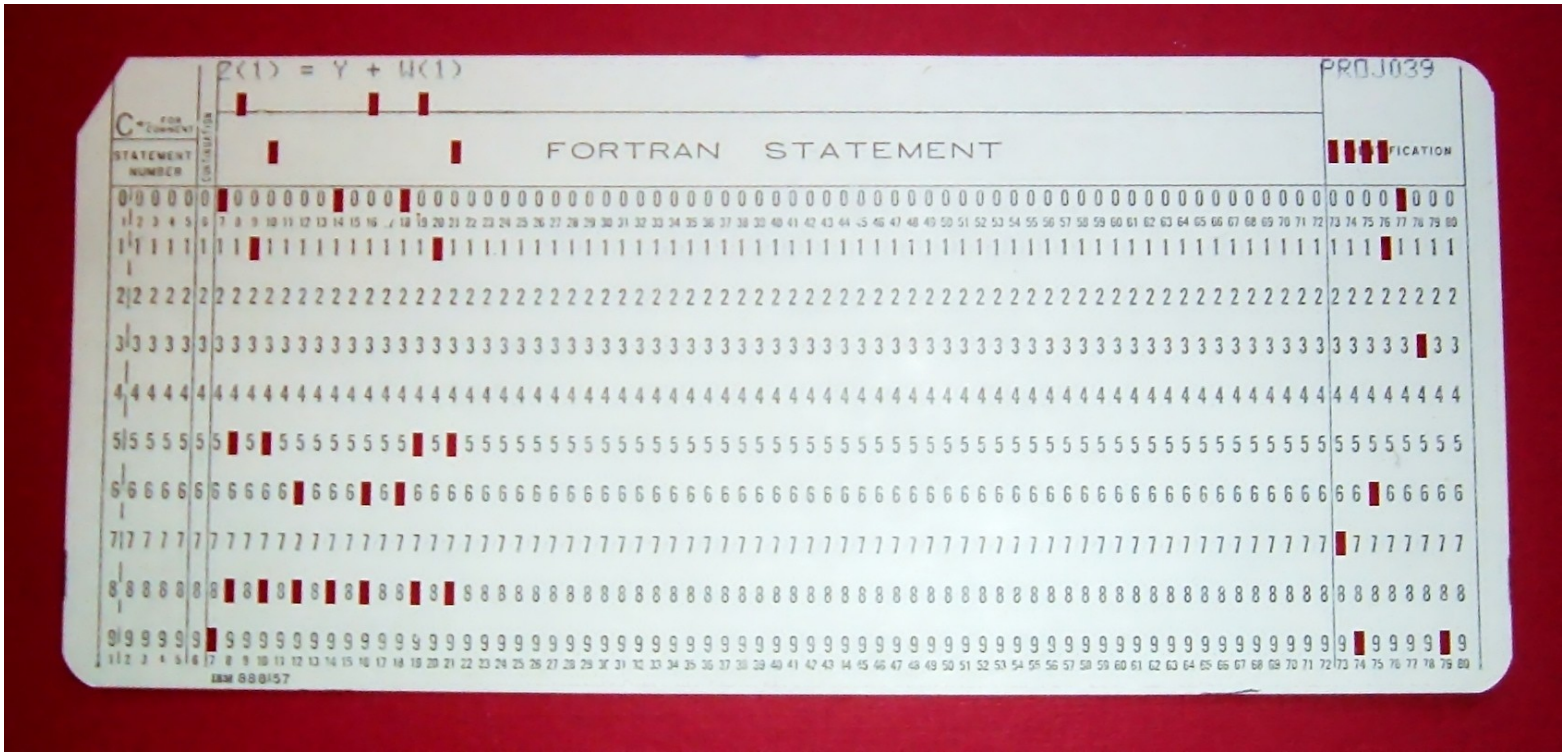
- Machines lived in warehouse sized computer rooms
- VM from 1966 but only on very large machines
- Small disk capacity in large boxes

Computing Landscape circa 1969

IBM 2311 Disk Drive

Precursor Technology and Tools

- **lines of source code**



Precursor Technology and Tools

- lines of source co
- source file

Precursor Technol

- lines of source cod
- source file
- sccs, cvs, git

Precursor Technology and Tools

- lines of source code
- source file
- sccs, cvs, git
- vi, emacs

Precursor Technology and Tools

Precursor Technology and Tools

- lines of source code
- source file
- sccs, cvs, git
- vi, emacs
- relocatable binary device driver

Precursor Technology and Tools

Precursor Technology and Tools

- lines of source code
- source file
- sccs, cvs, git
- vi, emacs
- relocatable binary device driver
- kernel debugger

Precursor Technology and Tools

Precursor Technology and Tools

- performance monitor

Still in the Future in 1969

- Codd’s first relational DB paper [70]
- First PDP-11 [70]
- LSI [70’s] and VLSI [80’s]
- ARPANet (Internet precursor) [72]
- DRAM replacing core memory [73]
- Unix V4 [74] and V5 [75]
- ICL Content Addressable File Store [75]

Still in the Future in 1969

- DEC VAX 11/782 [82]
- Ada [83]
- C++ [85]
- Berkeley TCP/IP code released [89]
- Linux 1.0 [94]
- multi-threading, POSIX, microprocessor, high-speed networking, solid-state disks, ...

Performance Issues circa 1969

- **Memory footprint**
- **File organisations: minimising IOPs and disk latency**
- **Algorithms and computational complexity**
- **Benchmarking as part of the hardware acquisition cycle**

Fast Forward ...

- **Computers are about 30 million times faster and bigger**
- **Memory is a non-issue (except for very large data)**
- **Disks and file organisations are invisible or largely irrelevant**
- **CPU speed makes algorithms less critical (except for very large data)**

Genesis of PCP

- Performance issues for SMP machines with 30 CPUs (hardware & software)
- NUMA adding complexity
- Performance problems in user space
- Humans need help to manage systems
- Performance analysis critical for sales revenue and support profitability
- Scarring from the sleazy benchmarking era

Experience from Real Production Environments

- Something is broken all of the time
- SLAs quote end-user performance metrics
- Cyclic patterns of workload and demand
- Transients are common
- Natural evolution changes everything, so stay awake!

Neanderthal Approaches Making the Problems Harder

- **Islands of performance data**
- **Reporting tools with narrow focus**
- **Emphasis on manual *post mortem* analysis with little automated *real-time* monitoring**

PCP Milestones

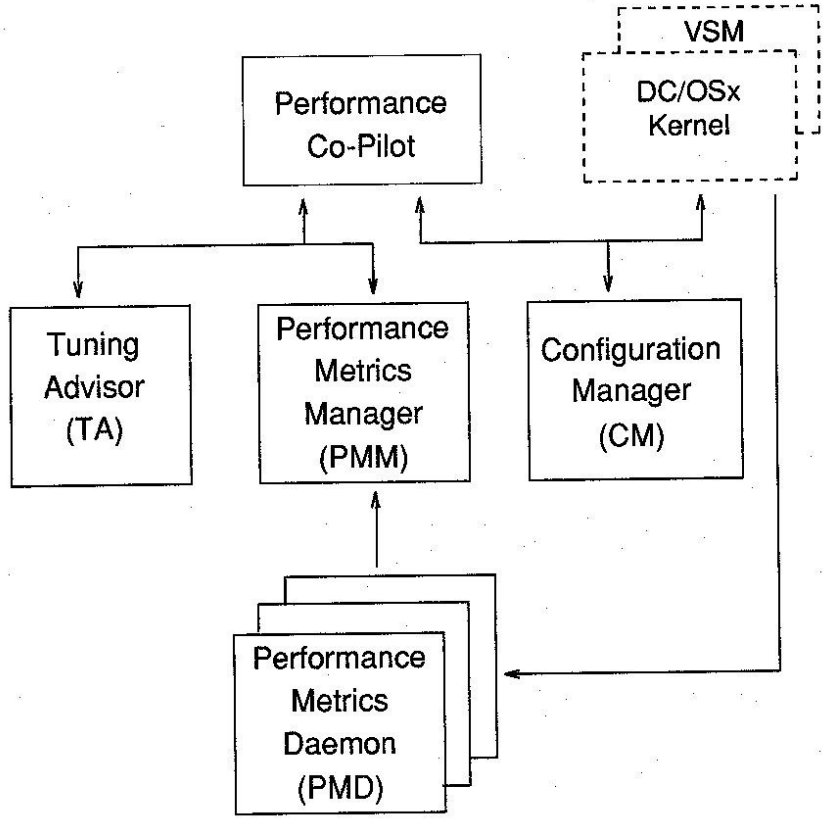
- **STL paper designs at Pyramid [early 92]**

PCP at Pyramid - Architecture

PYRAMID TECHNOLOGY
Systems Technology Laboratory

28 May 92
MESHine: Performance Co-Pilot – Phase 1 Review
FOIL 3

Performance Co-Pilot
Architectural Overview

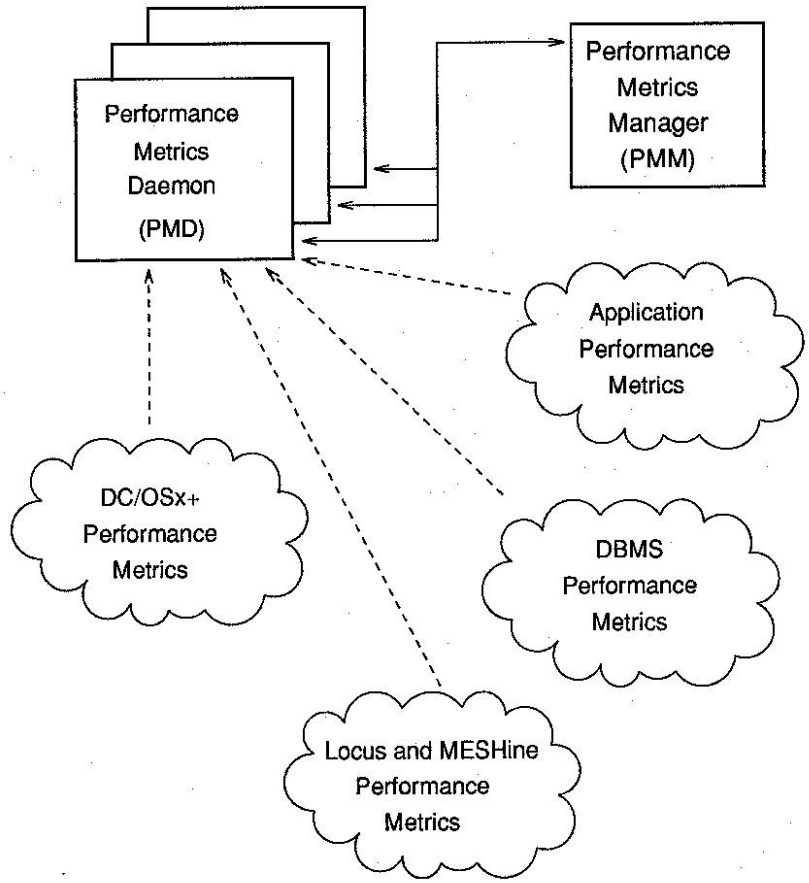


PCP at Pyramid – PMM (pmcd)

PYRAMID TECHNOLOGY
Systems Technology Laboratory

28 May 92
MESHine: Performance Co-Pilot – Phase 1 Review
FOIL 10

PMM and PMD Architecture



PCP Milestones

- STL paper designs at Pyramid [early 92]
- CMG paper [Aug 93]
- SGI team formed in Melbourne [Oct 93]
- PCP 1.0 SGI release [Apr 95]
- Linux porting experiments [95]
- PCP 1.2 SGI release [Aug 96]
- PCP 2.0 SGI release [May 98]

PCP Milestones (continued)

- **CXFS driving ports to other systems**
  - Solaris, AIX, HP/UX [95-96]
  - Mac, Windows, Linux [98]
- **Perl bindings for PMDAs [Sep 99]**
- **Open source releases**
  - collector infrastructure, core libs and tools [Dec 99]
  - QA suite [Oct 02]
  - odds and sods [Mar 05]

PCP Milestones (continued)

- SGI Altix and ia64 porting [01]
- Aconex [Oct 05]
- Dynamic names in the PMNS [Oct 09]
- RedHat: Mark [09], Nathan [12]
- Security work [Oct 00, Aug 12, Feb 17]
- Derived metrics [Jan 10]

PCP Milestones (continued)

- Thread-safe libpcp [Mar 12 … Dec 18]
- Python bindings [Aug 12]
- On-the-fly decompression of xz archives [Sep 14]
- Metric labels [Feb 19]
- 90 PMDAs in the source tree [Mar 19]

Stuff We Got Right

- Client-server architecture and PMAPI
- PMDA plug-in architecture
- Metadata accessible via PMAPI
- Data model
- Interchangeable real-time and historical data sources
- Flexible archive logging scope and frequency

Stuff We Got Right (continued)

- Archive replay interval independent of logging interval
- Single threading for pmcd
- Portability
- QA infrastructure and obsession

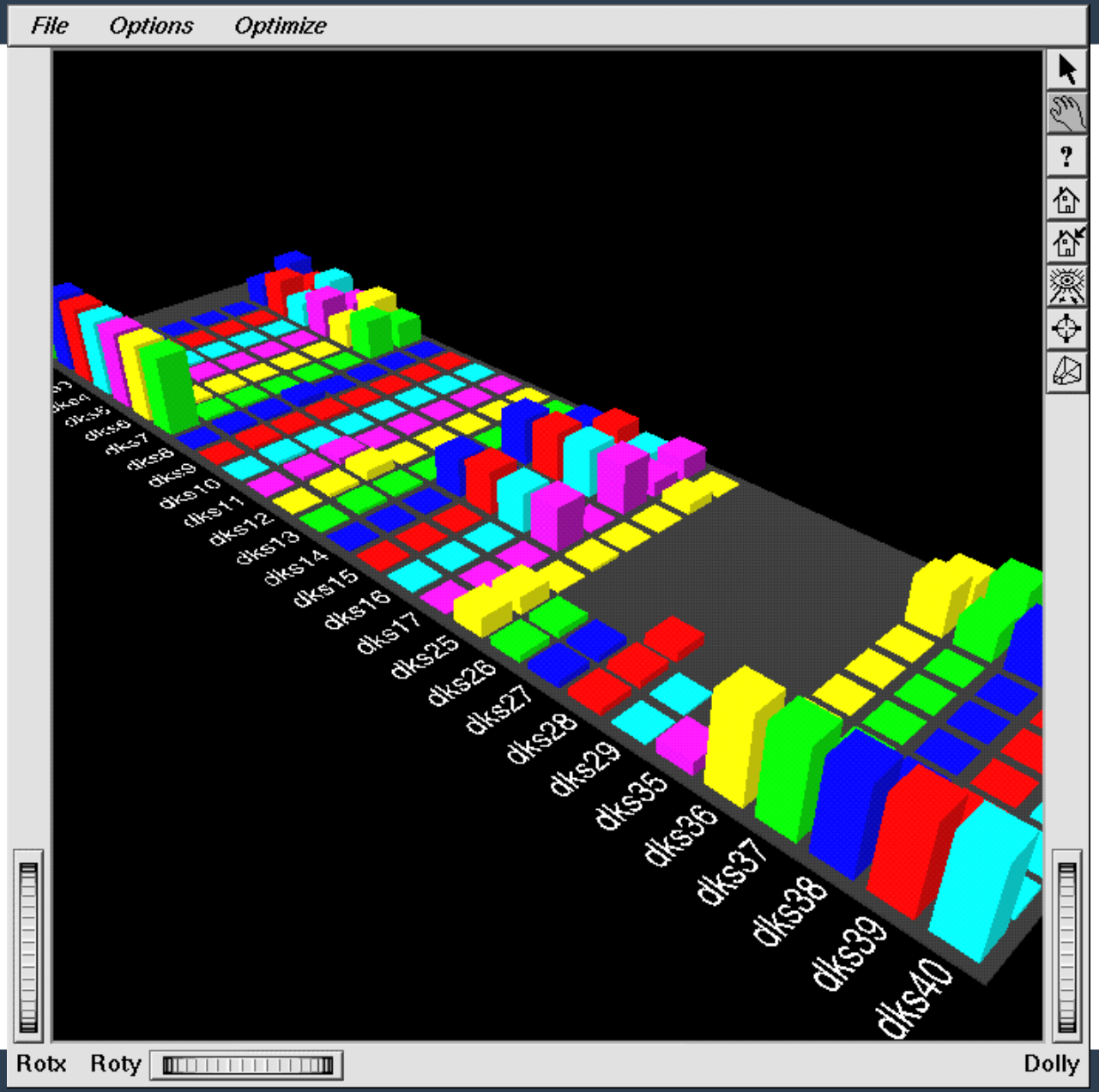
Mistakes

- Assuming we were secure behind a firewall
- Putting all the AI eggs in the pmie basket
- Adding features no one really wanted or used
- Akmal-driven premature optimizations
- Emasculated marketing

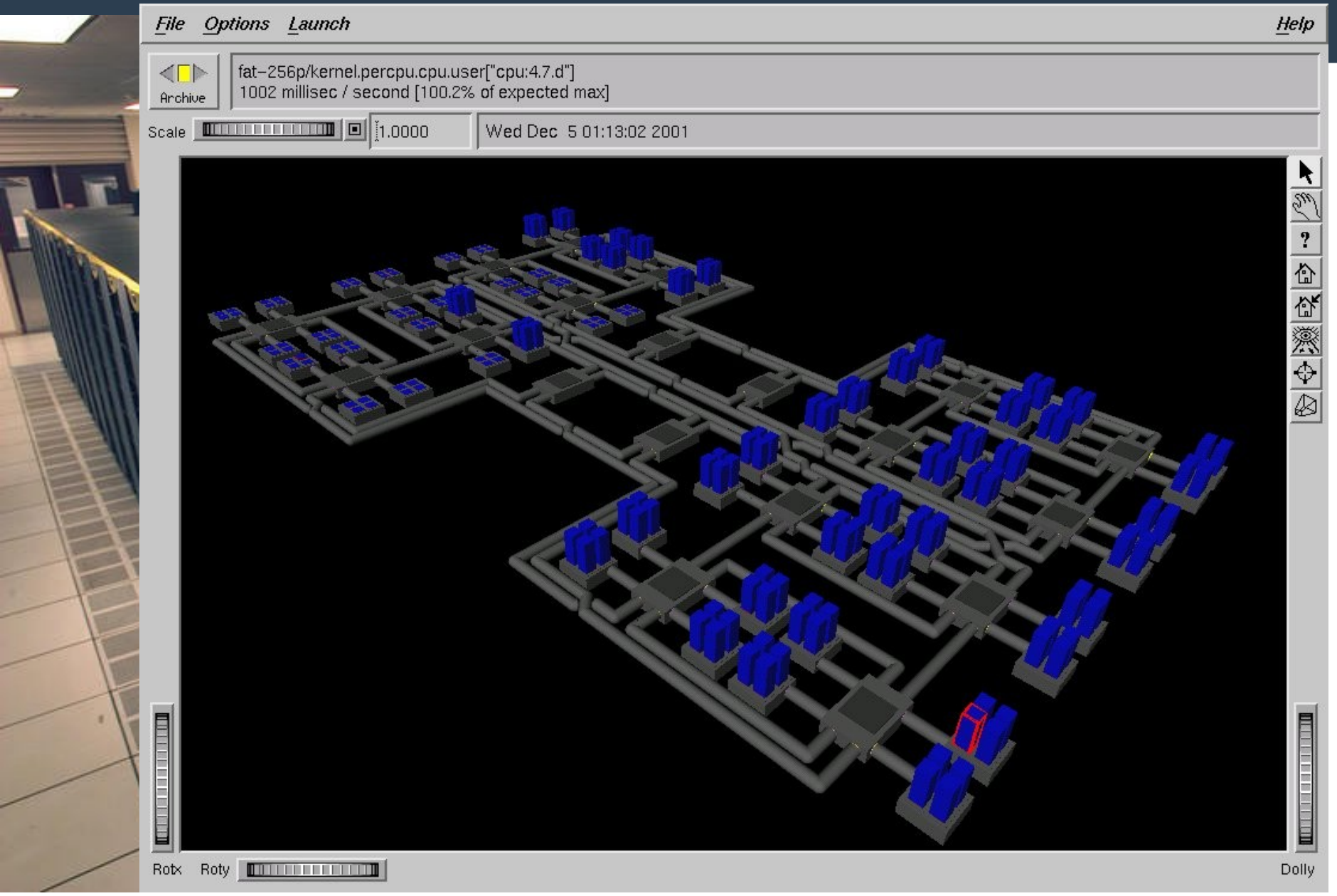
Things We Could Have Done Better

- Instance domain mapping
- Be strategic about non-Linux ports
- Near-zero effort set up and configuration
- pmie, pmlogreduce and pmgadgets
- Push harder to keep pmview working and evolving

Visualisation and Big Disk Farms



Visualisation and Big Machines



The Co-Pilot Analogy

- **Humans are unable to fly modern aircraft without on-board avionics**
- **Computer systems are just the same**
- **We've only scratched the surface here**
  - the PCP data infrastructure is robust
  - so far, only primitive reasoning and alerting
  - no automated learning
  - no tuning advisor
  - we've lost the initiative in visualisation

Performance Analysis in the Future

- **PCP archives are a data source, not the end-game**
  - we have the information to populate huge repositories of historical data
  - others will provide the query tools
- **Application of real machine learning techniques**
- **Serious understanding through new performance visualisations**

Questions?