

A Scalable Cross-Platform Infrastructure for Application Performance Tuning Using Hardware Counters

Shirley Browne, Jack Dongarra, Nathan Garner,
Kevin London, Phil Mucci, and Keith Seymour



Innovative Computing Laboratory
UNIVERSITY OF TENNESSEE
COMPUTER SCIENCE DEPARTMENT

<http://icl.cs.utk.edu/>

1



Innovative Computing Laboratory
UNIVERSITY OF TENNESSEE
COMPUTER SCIENCE DEPARTMENT

Tools for Performance Evaluation

- ◆ **Timing and performance evaluation has been an art**
 - Resolution of the clock
 - Issues about cache effects
 - Different systems
 - Can be cumbersome and inefficient with traditional tools
- ◆ **Situation about to change**
 - Today's processors have internal counters



2

Performance Counters

- ◆ Almost all high performance processors include hardware performance counters.
- ◆ Some are easy to access, others not available to users.
- ◆ On most platforms the APIs, if they exist, are not appropriate for the end user or well documented.
- ◆ Existing performance counter APIs
 - Compaq Alpha EV 6 & 6/7
 - SGI MIPS R10000
 - IBM Power Series
 - CRAY T3E
 - Sun Solaris
 - Pentium Linux and Windows
 - IA-64
 - HP-PA RISC
 - Hitachi
 - Fujitsu
 - NEC



Performance Data That May Be Available

- Cycle count
- Floating point instruction count
- Integer instruction count
- Instruction count
- Load/store count
- Branch taken / not taken count
- Branch mispredictions
- Pipeline stalls due to memory subsystem
- Pipeline stalls due to resource conflicts
- I/D cache misses for different levels
- Cache invalidations
- TLB misses
- TLB invalidations

Overview of PAPI

- ◆ Performance Application Programming Interface
- ◆ The purpose of the PAPI project is to design, standardize and implement a portable and efficient API to access the hardware performance monitor counters found on most modern microprocessors



5

Goals

- ◆ Provide a solid foundation for cross platform performance analysis tools
- ◆ Standardization between vendors, academics, and users.
- ◆ Provide a number of implementations tools for HPC architectures.
- ◆ Well documented, easy to use, and portable.

6

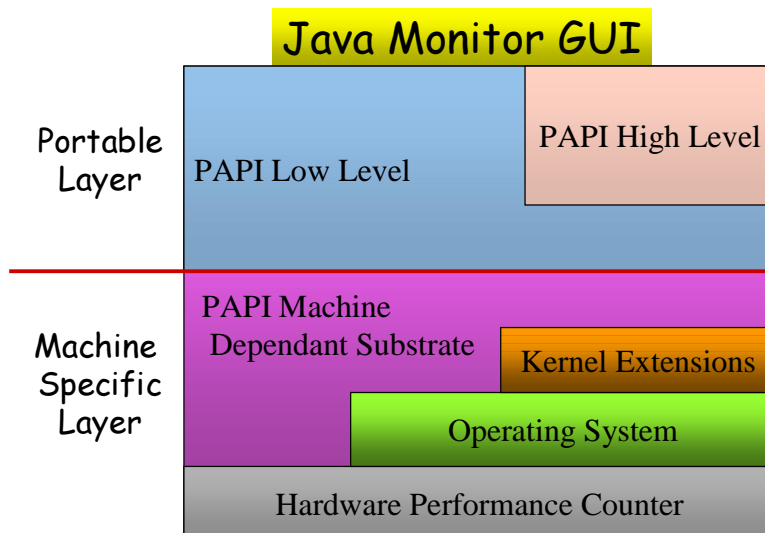


Implementation

- ◆ Counters exist as a small set of registers that count *events*.
- ◆ PAPI provides three interfaces to the underlying counter hardware:
 1. The low level interface manages hardware events in user defined groups called EventSet.
 2. The high level interface simply provides the ability to start, stop and read the counters for a specified list of events.
 3. Graphical tools to visualize information.

7

PAPI Implementation



8

Low Level API

- ◆ Increased efficiency and functionality over the high level PAPI interface
- ◆ There's about 40 functions
- ◆ Obtain information about the executable and the hardware.
- ◆ Thread safe

9

High Level API

- ◆ Meant for application programmers wanting coarse-grained measurements
- ◆ Calls the lower level API
- ◆ Not thread safe at the moment
- ◆ Only allows PAPI Presets events

10

High Level Functions

- ◆ PAPI_flops()
- ◆ PAPI_num_counters()
 - Number of counters in the system
- ◆ PAPI_start_counters()
- ◆ PAPI_stop_counters()
 - Enable counting of events and describes what to count
- ◆ PAPI_read_counters()
 - Returns event counts

11

PAPI - Supported Processors

- ◆ Pentium, Pro, II, III,
 - Linux 2.4, 2.2, 2.0 and perf kernel patch
- ◆ Power 3, 604, 604e
 - For AIX 4.3 and pmtoolkit (in 4.3.4 available)
 - (laderose@us.ibm.com)
- ◆ UltraSparc I&II (III soon)
 - Solaris 8
- ◆ MIPS R10K, R12K
- ◆ AMD Athlon
 - Linux 2.4 and perf kernel patch
- ◆ Cray T3E, SV1, SV2
- ◆ Soon: Alpha EV6, EV67

12



Graphical Tools Perfometer Usage

- ◆ Application is instrumented with PAPI
 - call `perfometer()`
- ◆ Will be layered over the best existing vendor-specific APIs for these platforms
- ◆ Application is started, at the call to `perfometer` signal handler and timer set to collect and send the information to a Java applet containing the graphical view.
- ◆ Sections of code that are of interest can be designated with specific colors
 - Using a call to `set_perfometer('color')`



Perfometer Features

- ◆ Platform independent visualization of PAPI metrics
- ◆ Flexible interface
- ◆ Quick interpretation of complex results
- ◆ Small footprint
 - (compiled code size < 15k)
- ◆ Color coding to highlight selected procedures
- ◆ Trace file generation or real time viewing.

Innovative Computing Laboratory
UNIVERSITY OF TENNESSEE
COMPUTER SCIENCE DEPARTMENT

Perfometer

Machine info

Flops issued

Flop/s Rate

Flop/s Instantaneous Rate

Process & Real time

Call Perfometer('red')

15

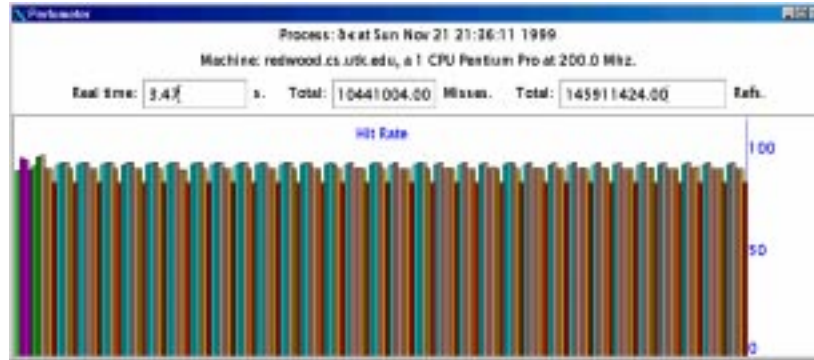
Innovative Computing Laboratory
UNIVERSITY OF TENNESSEE
COMPUTER SCIENCE DEPARTMENT

Go To Demo

PAPI

16

Cacheometer



- ◆ Cacheometer - Perfometer using cache hit rate metric

17

PAPI Release

- ◆ **Platforms**
 - Pentium Linux/x86
 - Require patch to kernel
 - Sun Solaris/Ultra 2.8
 - IBM AIX/Power
 - Contact IBM for PMtoolkit
 - SGI IRIX/MIPS
 - AMD Athlon Linux
 - Compaq Tru64/Alpha (Soon)
- ◆ **Mailing list**
 - send "subscribe ptools-perfapi" to majordomo@ptools.org
 - ptools-perfapi@ptools.org is the reflector



- ◆ **C and Fortran bindings**
- ◆ **To download software see:**
<http://icl.cs.utk.edu/papi/>

18

PAPI 1.1 beta 2 Implementation

- ◆ Fully programmable, low level interface
thread safe - directed towards the tool
developer and the sophisticated user
- ◆ High level API for the non-expert
- ◆ C and Fortran binding
- ◆ Supports native events and 104 "preset"
events, which are commonly available
metrics, some are derived.
- ◆ User can query to see if a preset exists

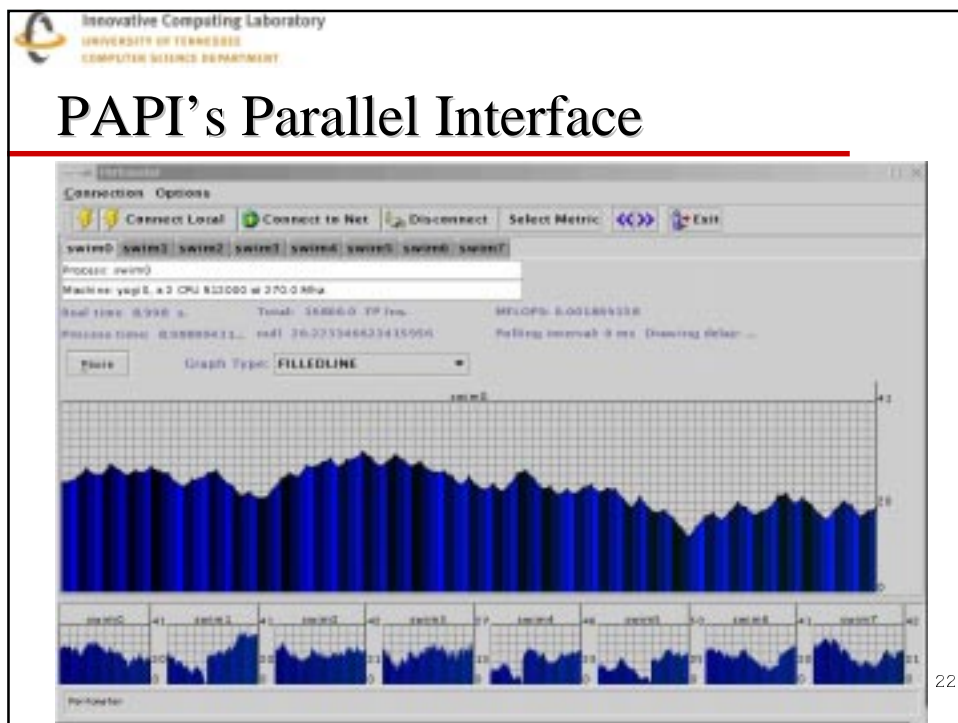
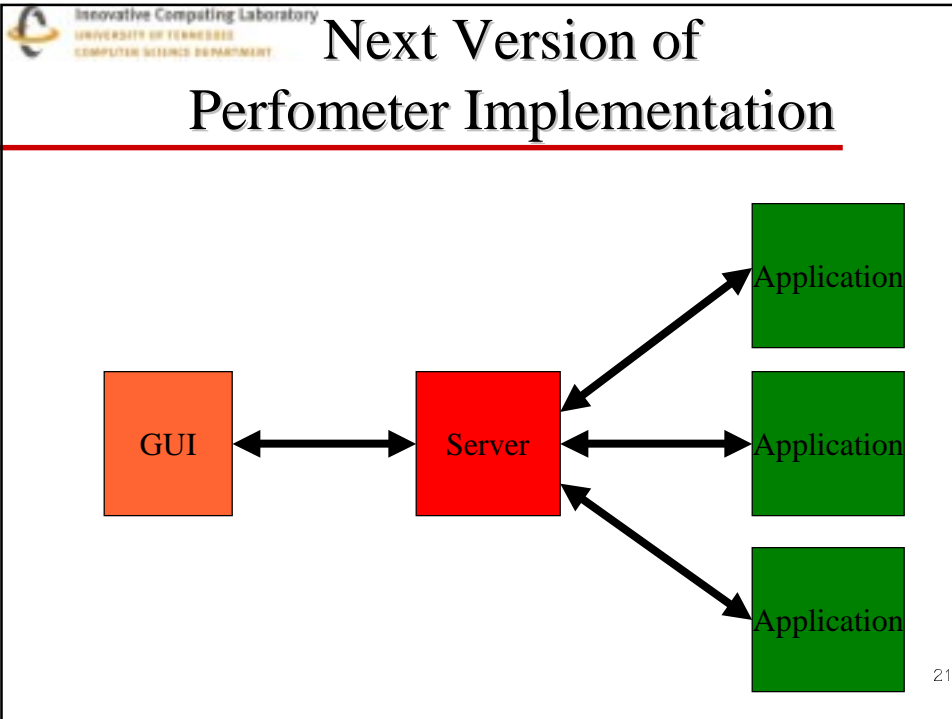
19

Early Users of PAPI



- ◆ DEEP/PAPI (Pacific Sierra) 
http://www.psrv.com/deep_papi_top.html
- ◆ TAU (Allen Mallony, U of Oregon) 
<http://www.cs.uoregon.edu/research/paracomp/tau/>
- ◆ SvPablo (Dan Reed, U of Illinois) 
<http://vibes.cs.uiuc.edu/Software/SvPablo/svPablo.htm>
- ◆ Cactus (Ed Seidel, Max Plank/U of Illinois) 
<http://www.aei-potsdam.mpg.de>
- ◆ Vprof (Curtis Janssen, Sandia Livermore Lab)
<http://aros.ca.sandia.gov/~cljanss/perf/vprof/>
- ◆ Cluster Tools (Al Geist, ORNL)
- ◆ DynaProf (Phil Mucci, UTK)
<http://www.cs.utk.edu/~mucci/dynaprof/>

20



Contributors to These Ideas

◆ **PAPI**

- Shirley Browne, UTK
- Nathan Garner, UTK
- Kevin London, UTK
- Phil Mucci, UTK
- Keith Seymour, UTK



For additional
information see...

<http://icl.cs.utk.edu/papi/>

<http://www.cs.utk.edu/~dongarra/>



Opportunities in the group for visiting and permanent positions

23

<http://icl.cs.utk.edu/papi/>



PAPI

24