HPC-ASIA 2000
The Fourth International Conference/Exhibition on
High Performance Computing in Asia-Pacific Region
May 14-17, 2000
Friendship Hotel, Beijing, China

Tools for High Performance Numerical Kernels, and Performance Measurement

Jack Dongarra
University of Tennessee
and
Oak Ridge National Laboratory
http://www.cs.utk.edu/~dongarra/

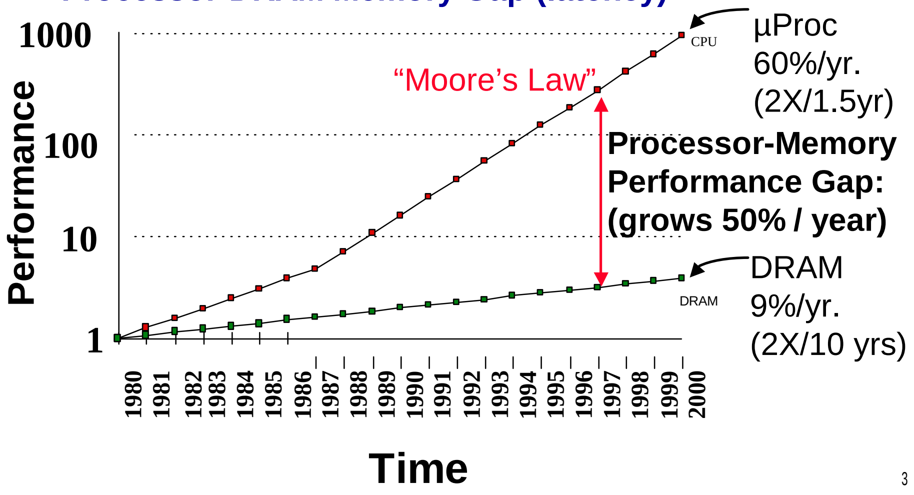
Outline

? **Automatically Tuned Linear Algebra Software (ATLAS)**

? **Standards and Tools for Performance Monitoring (PAPI)**

Where Does the Performance Go? or
Why Should I Cares About the Memory Hierarchy?

Processor-DRAM Memory Gap (latency)



µProc
60%/yr.
(2X/1.5yr)

"Moore's Law"

Processor-Memory Performance Gap: (grows 50% / year)

DRAM
9%/yr.
(2X/10 yrs)

Optimizing Computation and Memory Use

? **Computational optimizations**

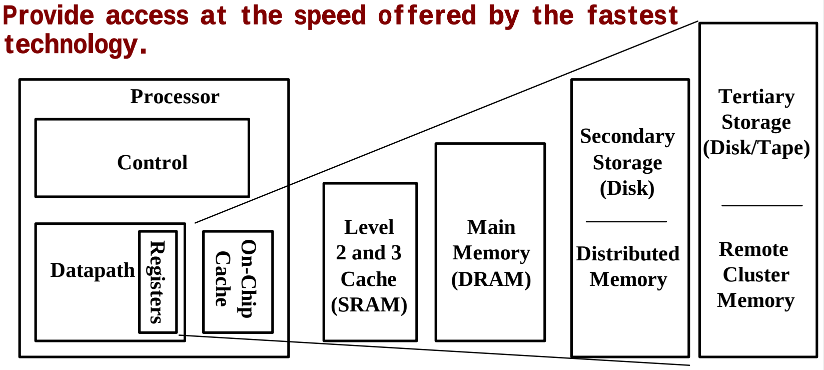
?

Memory Hierarchy

? By taking advantage of the principle of locality:

? Present the user with as much memory as is available in the cheapest technology.

? Provide access at the speed offered by the fastest technology.



Speed (ns):	1s	10s	100s	10,000,000s (10s ms)	10,000,000,000s (10s sec)
Size (bytes):	100s	Ks	Ms	100,000 s (.1s ms) Gs	10,000,000 s (10s ms) Ts

How To Get Performance From Commodity Processors?

? Today's processors can achieve high-performance, but this requires extensive machine-specific hand tuning.

? Hardware and software have a large design space w/many parameters

? Blocking sizes, loop nesting permutations, loop unrolling depths, software pipelining strategies, register allocations, and instruction schedules.

? Complicated interactions with the increasingly sophisticated micro-architectures of new microprocessors.

? Until recently, no tuned BLAS for Pentium for Linux.

? Need for quick/dynamic deployment of optimized routines.

? ATLAS - Automatic Tuned Linear Algebra Software

? PhiPac from Berkeley

?

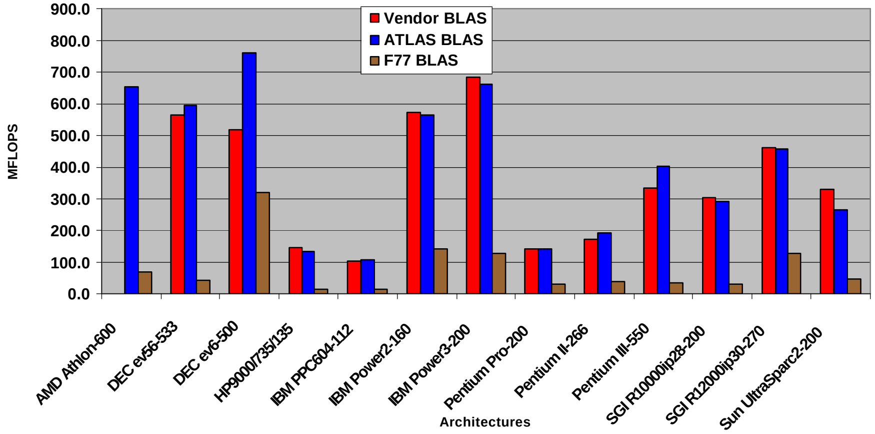
ATLAS

? **An adaptive software architecture**

- ?**High-performance**
- ?**Portability**
- ?**Elegance**

? **ATLAS is faster than all other portable BLAS implementations and it is comparable with machine-specific libraries provided by the vendor.**

ATLAS (DGEMM n = 500)

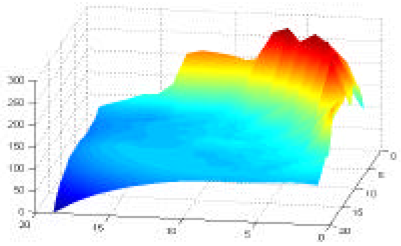


? **ATLAS is faster than all other portable BLAS implementations and it is comparable with machine-specific libraries provided by the vendor.**

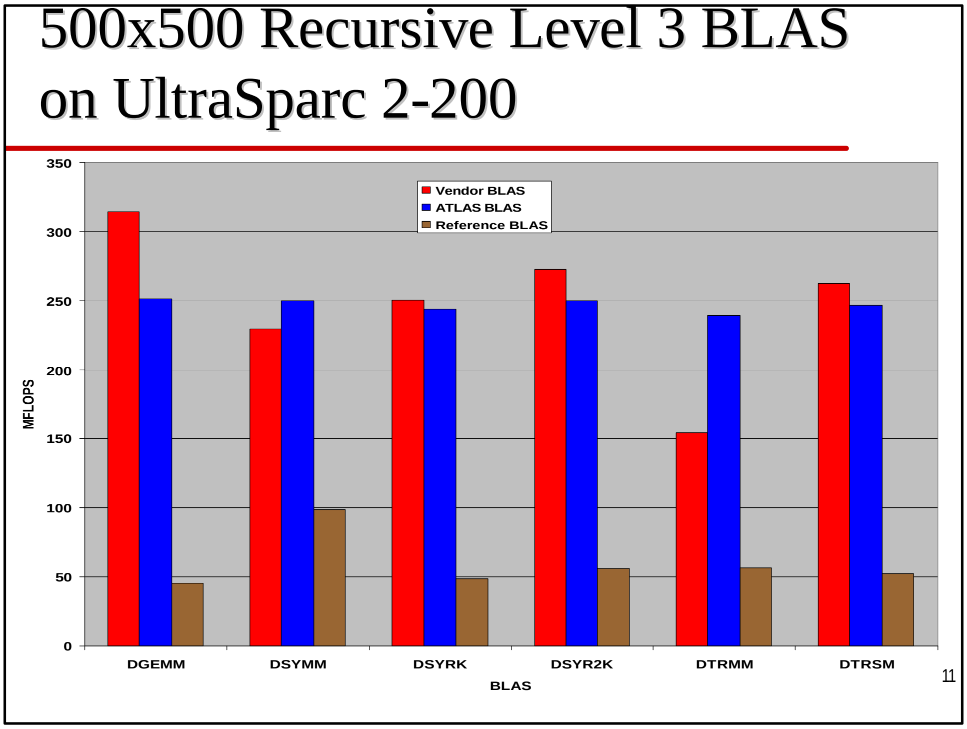
Why ATLAS Is Fast?

- ? ATLAS does not implement a single fixed algorithm.
- ? The code is generated by a program that tests, probes, and runs 100's of experiments on the target sw/hw architecture.
- ? During installation the program generator determines an efficient implementation
  - ? Probes systems for critical parameters
  - ? Measures the speed of different code strategies and chooses the best using an adaptive procedure.
- ? This leads to a new model of high performance programming in which performance critical code is machine generated using parameter optimization.
- ? Done once to build the library, then used on that machine.

Code Generation Strategy

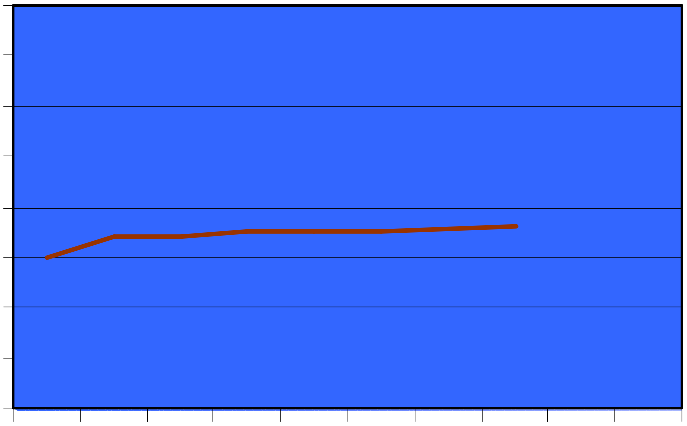


- ? On-chip multiply optimizes for:
  - ? TLB access
  - ? L1 cache reuse
  - ? FP unit usage
  - ? Memory fetch
  - ? Register reuse
  - ? Loop overhead minimization
- ? Takes a 30 minutes to a hour to run.
- ? New model of high performance programming where critical code is machine generated using parameter optimization.
- ? Code is iteratively generated & timed until optimal case is found. We try:
  - ? Differing NBs
  - ? Breaking false dependencies
  - ? M, N and K loop unrolling
- ? Designed for RISC arch
  - ? Super Scalar
  - ? Need reasonable C compiler

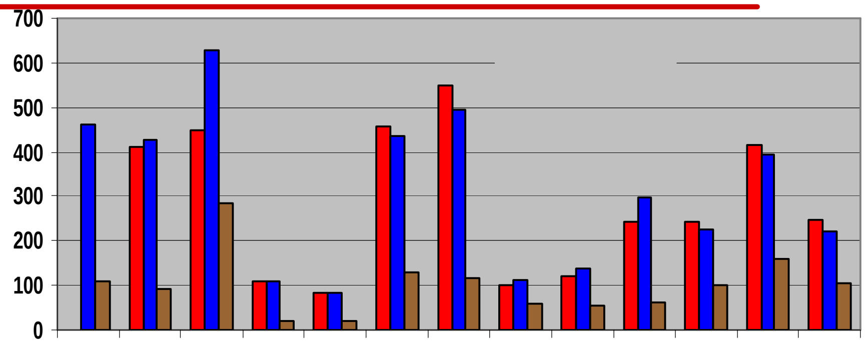


500x500 Recursive Level 3 BLAS on UltraSparc 2-200

11

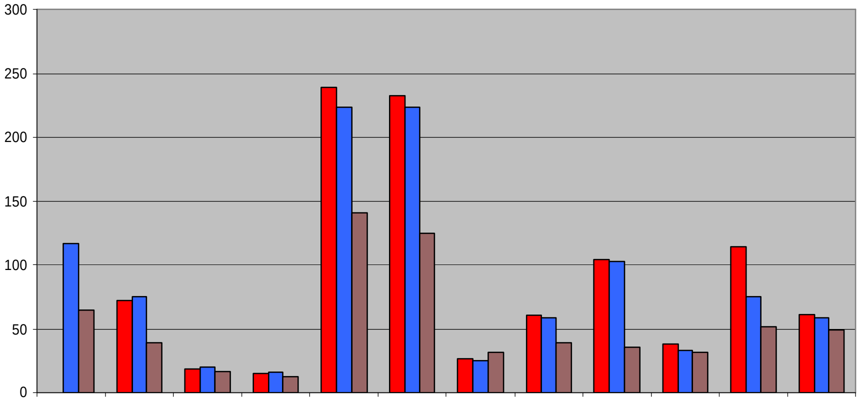


Ax = b, n = 500, (LU Right-Looking)



13

500x500 Level 2 BLAS DGEMV



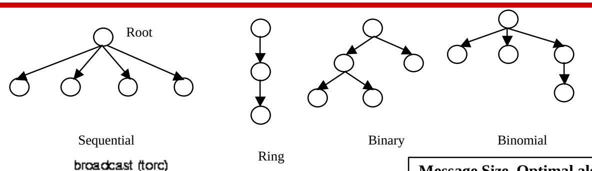
15

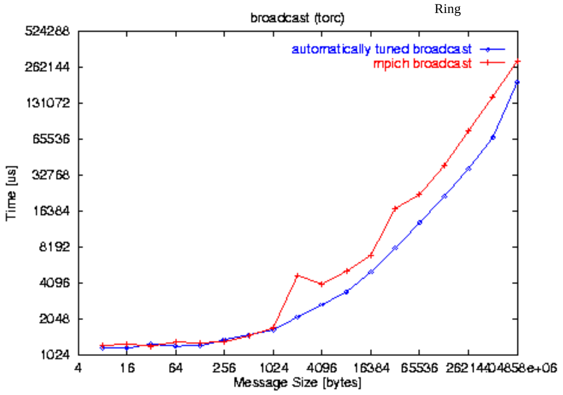
Plans for ATLAS

- **Software Release, available today:**
  - **Level 1, 2, and 3 BLAS implementations**
  - **See: www.netlib.org/atlas/**
- **Next Version:**
  - **Multi-treading**
  - **Fortran and Java generators**
- **Futures:**
  - **Optimize message passing system**
  - **Runtime adaptation**
    - **Sparsity analysis**
    - **Iterative code improvement**
  - **Specialization for user applications**
  - **Adaptive libraries**

Work in Progress: ATLAS Approach Applied to Broadcast

(PII 8 Way Cluster with 100 Mb/s switched network)





Message Size (bytes)	Optimal algorithm	Buffer Size (bytes)
8	binomial	8
16	binomial	16
32	binary	32
64	binomial	64
128	binomial	128
256	binomial	256
512	binomial	512
1K	sequential	1K
2K	binary	2K
4K	binary	2K
8K	binary	2K
16K	binary	4K
32K	binary	4K
64K	ring	4K
128K	ring	4K
256K	ring	4K
512K	ring	4K
1M	binary	4K

Tools for Performance Evaluation

? Timing and performance evaluation has been an art

  ? Resolution of the clock

  ? Issues about cache effects

  ? Different systems

? Situation about to change

  ? Today’s processors have internal counters

19

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

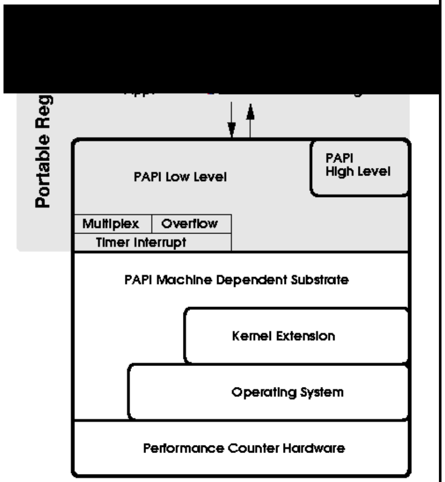
? TLB misses

? TLB invalidations

PAPI Implementation

? Performance Application Programming Interface

? The purpose of PAPI is to design, standardize and implement a portable and efficient API to access the hardware performance monitor counters found on most modern microprocessors


[redacted]

Implementation

? Counters exist as a small set of registers that count *events.*

? PAPI provides three interfaces to the underlying counter hardware:

? The low level interface manages hardware events in user defined groups called EventSet.

? The high level interface simply provides the ability to start, stop and read the counters for a specified list of events.

? Graphical tools to visualize information.

23

Graphical Tools Perfometer Usage

? Application is instrumented with PAPI

? call perfometer()

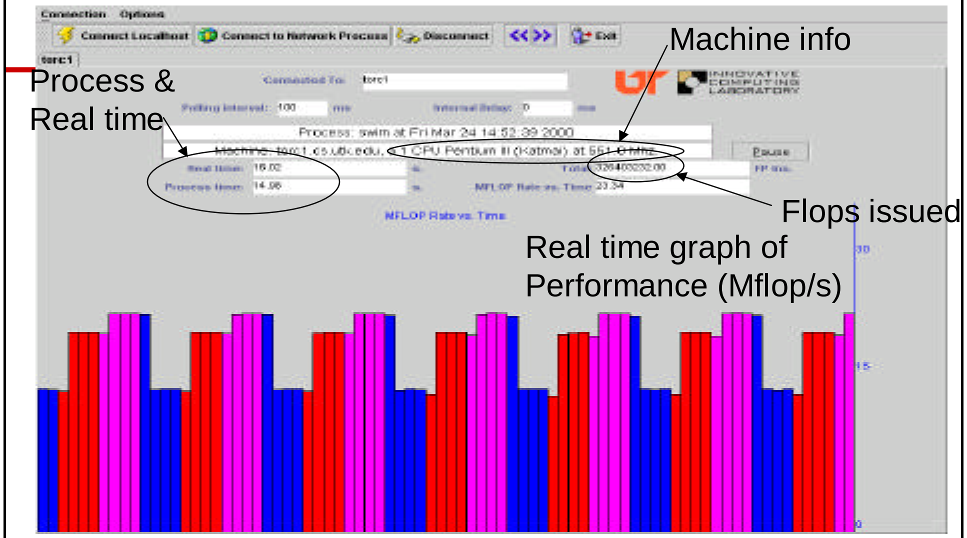
? Will be layered over the best existing vendor-specific APIs for these platforms

? Sections of code that are of interest are designated with specific colors

? Using a call to set_perfometer(color)

? Application is started, at the call to performeter a task is spawned to collect and send the information to a Java applet containing the graphical view.

Perfometer



Connected To: torc1

Polling Interval: 100 ms Internal Delay: 0 ms

Process: swim at Fri Mar 24 14:52:39 2000

Machine: torc1.cs.utk.edu, 1 CPU Pentium III (Katmai) at 551.0 MHz

Real time: 16.02 s Total: 326403232.00 FP ins.

Process time: 14.98 s MFLOP Rate vs. Time: 23.34

Machine info

Process &
Real time

Flops issued

Real time graph of
Performance (Mflop/s)

? **Perfometer monitoring one of many metrics available**

25

Call Perfometer(red)

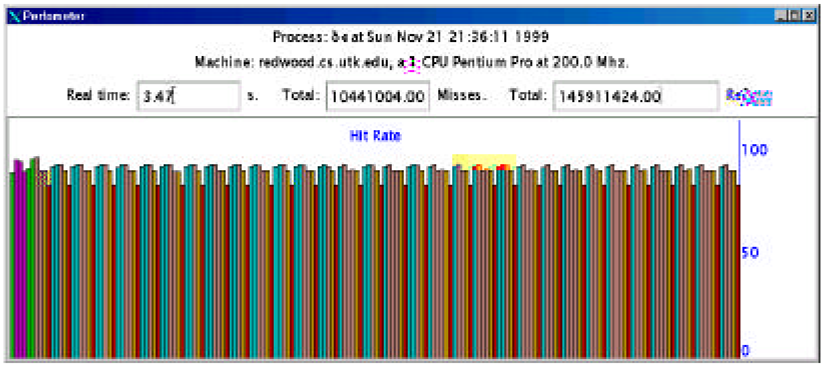
Go To Demo

26

Cacheometer Features

- Cacheometer is actually Perfometer running with a different metric specified to PAPI
- Shown here as an example of the versatility of the Perfometer application
- Zoom in to monitor certain code segments more closely

Cacheometer



Process: 6e at Sun Nov 21 21:36:11 1999

Machine: redwood.cs.utk.edu, a 1 CPU Pentium Pro at 200.0 Mhz.

Real time: 3.47 s. Total: 10441004.00 Misses. Total: 145911424.00

- Cacheometer – Perfometer using cache hit rate metric

PAPI 1.0 Release

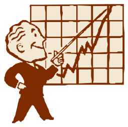
? Platforms
  - ? Linux/x86
  - ? Solaris/Ultra
  - ? AIX/Power
  - ? Tru64/Alpha
  - ? IRIX/MIPS

? Fortran wrappers

? Mailing list
  - ? send "subscribe ptools-perfapi" to `majordomo@ptools.org`
  - ? `ptools-perfapi@ptools.org` is the reflector

? To download software see:
http://icl.cs.utk.edu/projects/papi/

Early Users of PAPI



? Paradyn (Bart Miller, U of Wisconsin)
http://www.cs.wisc.edu/paradyn/libhrtime/

? TAU (Allen Mallony, U of Oregon)
http://www.cs.uoregon.edu/research/paracomp/tau/

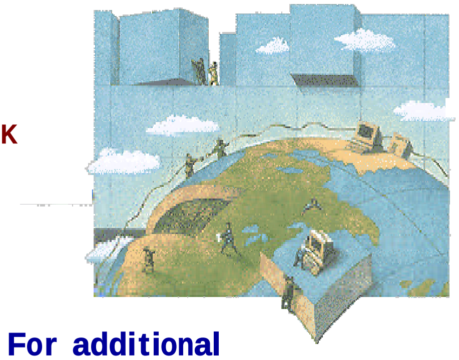
? SvPablo (Dan Reed, U of Illinois)
http://wwwpablo.cs.uiuc.edu/Project/SVPablo/SvPabloOverview.htm

? 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/

Contributors to These Ideas

- ? **ATLAS**
  - ? Clint Whaley, UTK
  - ? Antoine Petitet, UTK
  - ? Tatebe Osamu, ETL/UTK
  - ? Sathish Vadhiyar, UTK
- ? **PAPI**
  - ? Shirley Browne, UTK
  - ? Nathan Garner, UTK
  - ? Kevin London, UTK
  - ? Phil Mucci, UTK



INNOVATIVE COMPUTING LABORATORY

For additional information see…

http://www.netlib.org/atlas/

http://icl.cs.utk.edu/projects/papi/

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

31

http://icl.cs.utk.edu/projects/papi/

32