

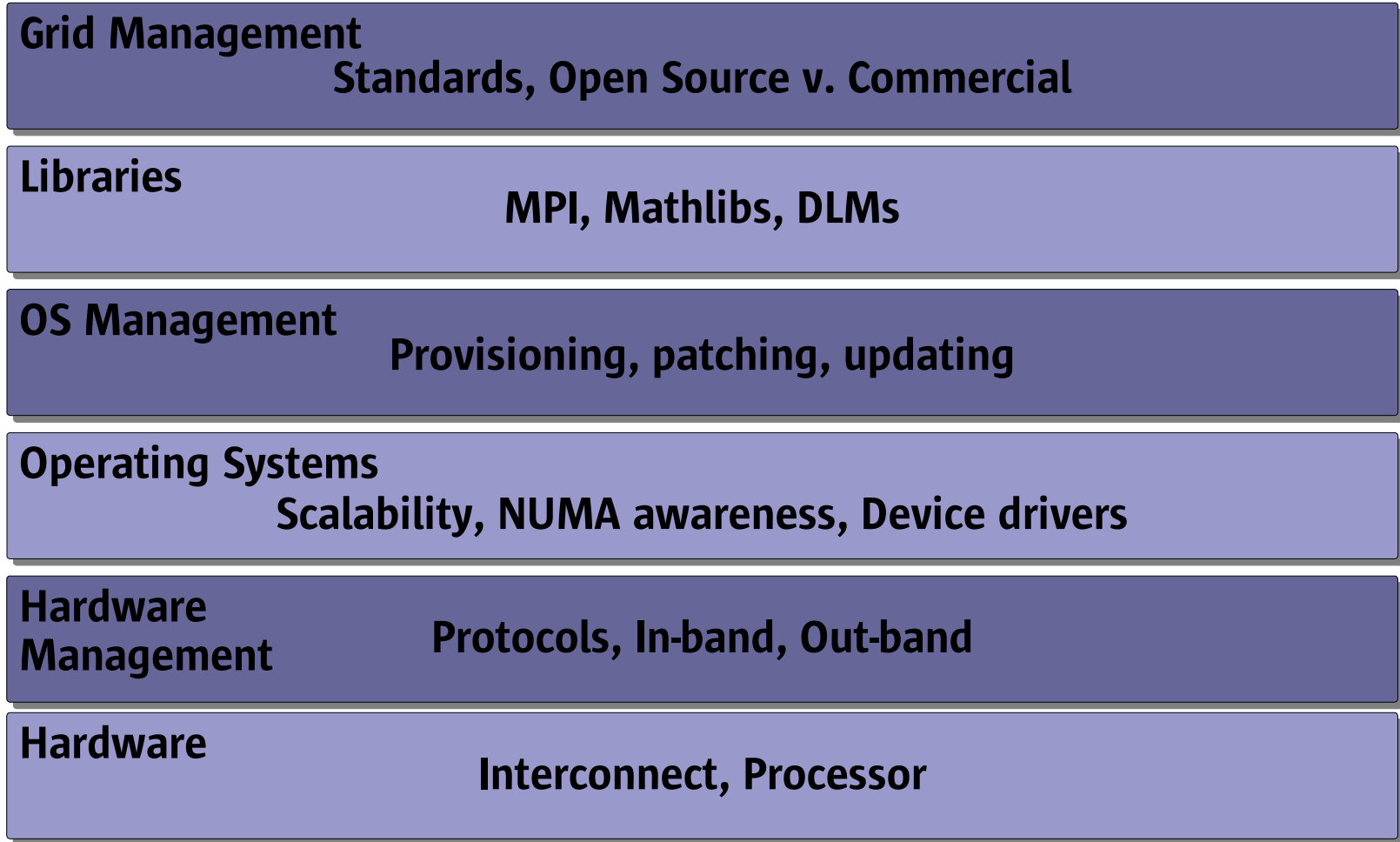
Performance Tools and Environments

Carlo Nardone
Technical Systems Ambassador
GSO – Client Solutions



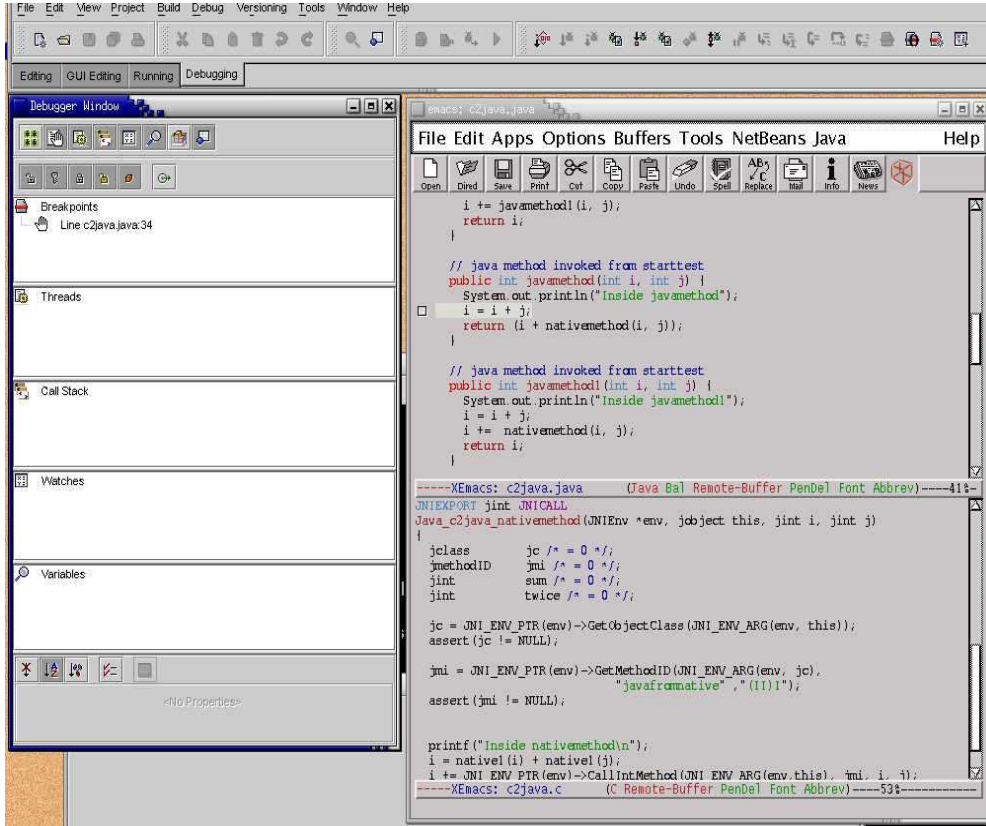
The Stack

Applications



Sun's Developer Products

State-of-the-Art Tools



For Corporate Java Developers:
Sun Java Studio Creator
 Visual development for rapid
 service creation

For Enterprise Java Developers:
Sun Java Studio Enterprise
 Complete Develop-Test-
 Deploy to the Sun Java
 Enterprise System

For C/C++/Fortran Developers:
Sun Studio
 Full application environment
 for Solaris (SPARC, Xeon, AMD)
 and Linux

All Sun's Developer products based on the
 NetBeans open source IDE platform

Sun Studio Products


www.sun.com/software/products/studio



- Sun Studio 9 (July '04)
 - Optimize development and performance of apps on UltraSPARC IV and AMD systems
 - Solaris 8, 9, 10, Java Desktop 2.0, SuSE Enterprise Server 8, RedHat Enterprise 3
 - <http://www.sun.com/software/products/studio>
- Back-rev products
 - Sun Studio 8 (Mar '04)
 - Sun ONE Studio 8 Compiler Collection (May '03)
 - Sun ONE Studio 7 Solaris Enterprise Edition (Oct '02, EOL '04)
 - Forte Developer 6 Update 2 (July '01)

Sun Studio 9

Tools for optimal app performance on Sun's new systems (SPARC, Xeon, AMD)

 Improved Performance

- UltraSPARC IV/CMT


- World Record OpenMP performance reported in Feb '04

- World-class Xeon, AMD32 code generation

- SSE, SSE2, P4 instruction support



 New Tools: Solaris x86 FORTRAN compilers and math libraries; Grid-aware make; Security code checker

 New Standards: ISO C99 Standard

 New Platforms:  redhat



New in Sun Studio 9

All Platforms

- Modified compiler optimization defaults for “Easy Performance”
- Automatic pre-compiled headers (C/C++)
- Faster C++ IOStreams with `unsync_stdio`
- Improved Performance Analyzer
- Improved IDE response, flow, and startup
- Optimized LAPACK, FFT, BLAS solvers
- C99 (ISO) branding
- Grid-aware `dmake`
- Source code security checking

UltraSPARC IV/CMT:

- C++ auto-par and vectorization
- Improved OpenMP optimizations & debugging
- Scalar Perf: (loop/cache opts, prefetching, interproc. analysis)

x86 (Intel/AMD) :

- FORTRAN tools, Perflib
- World-class code generation for Xeon and AMD32
- Linux: JDS, SuSE and RedHat

x86 Optimizations in Studio 9

- SSE2 instruction scheduling
- P4, SSE2 instr in assembler
- Handle P4, SSE2 in inlines
- Strength reduction
- Branch prediction
- Induction variable elim
- Invariant hoisting
- Loop interchange
- Loop unswitching
- Alignment of symbol blocks
- Loop unrolling
- Alignment
- Constant propagation
- Vectorization

-xarch=sse2 turns on SSE2 instructions on S9u6, S10

All optimizations are on for -xO4 and -fast

Sun Studio 10

www.sun.com/software/products/studio/s10preview.html

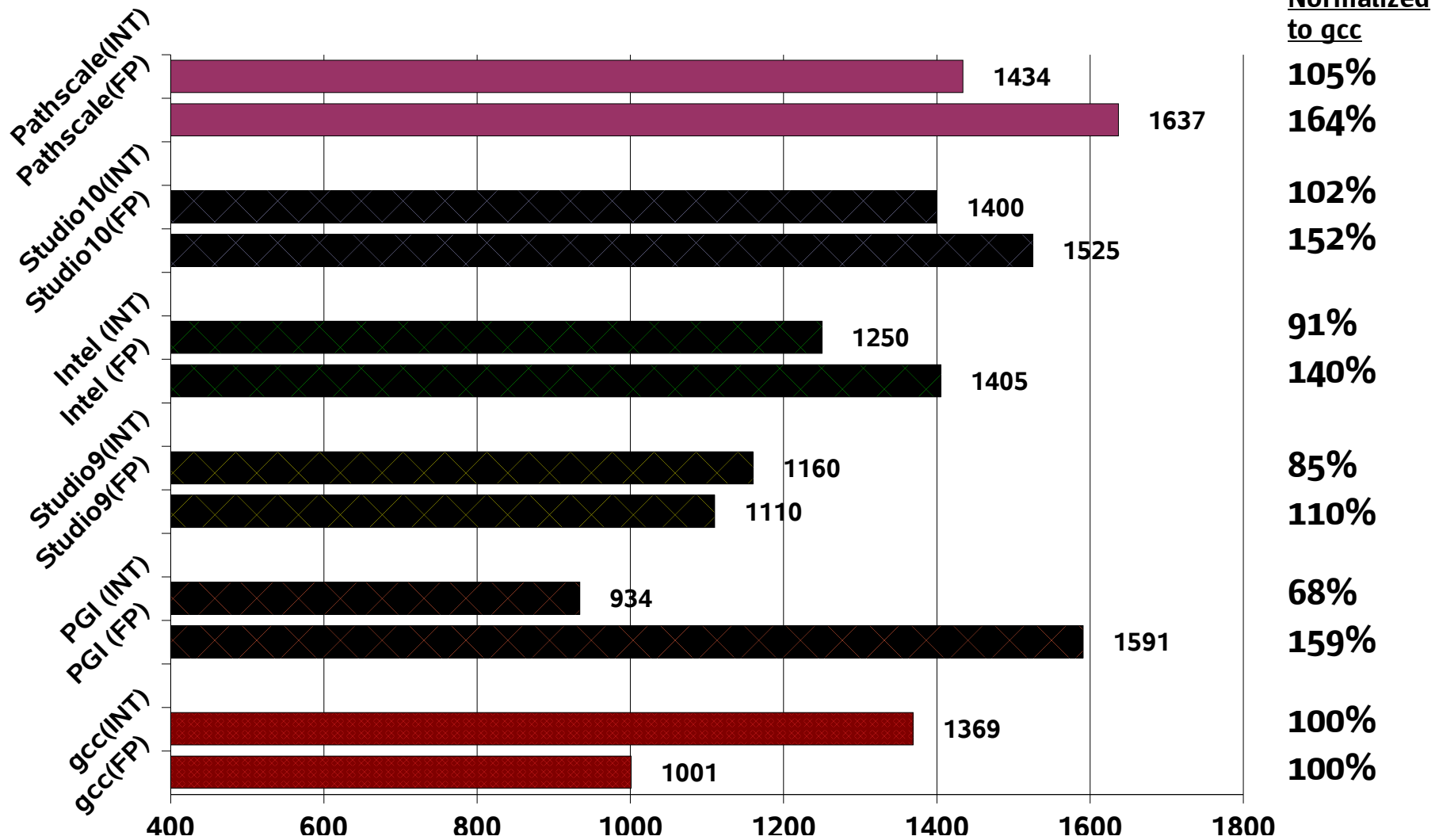
- Major tools upgraded for AMD64 performance
 - C, C++, Fortran compilers
 - Instruction selection tuned for AMD (32/64)
 - C++ template-template parameters (SPARC and x86)
 - Thread local storage in C / C++ (`__thread`)
 - Automatic parallelization (`-xautopar`)
 - Inter-procedural optimization (`-xipo`)
 - Profile-feedback (`-xprofile`)
 - OpenMP 2.0 support in C, C++, Fortran (`-xopenmp`)
 - Loop dependence analysis (`-xdepend`)
 - Loop vectorization (`-xvector`)
 - Restricted parameters (`-xrestrict`)
 - Type-based alias analysis (`-xalias_level`)
 - Memory prefetch (`-xprefetch`)
 - Non-standard floating point computation (`-fns`)
 - Dbx and Performance Analyzer for AMD64-bit apps
 - Performance libs optimized for AMD64 (`-library=sunperf`)

AMD64 Compilers

| | OS Support | Vector SIMD Support | Peels/ Aligns Loops | Inter- Procedural Analysis | OpenMP | Function Inlining | Profile Guided Feedback | 32 bit & 64- bit support | Parallel Debugger | Auto Loop Unrolling | Mc Model= medium |
|-------------------|----------------|---------------------|---------------------|----------------------------|--------|-------------------|-------------------------|--------------------------|-------------------|---------------------|------------------|
| Sun Studio | Solaris, Linux | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| GNU | Solaris, Linux | ∅ | ∅ | ∅ | ∅ | ✓ | ✓ | ✓ | ∅ | ✓ | ✓ |
| PGI | Solaris, Linux | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Intel | Linux | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ∅ | ✓ | ✓ | ∅ |
| Pathscale | Linux | ✓ | ∅ | ✓ | ∅ | ✓ | ✓ | ∅ | ∅ | ✓ | ∅ |

State-of-the-art Sun Studio Feature Set:
 debugger, perftools, OpenMP, IDE user I/F

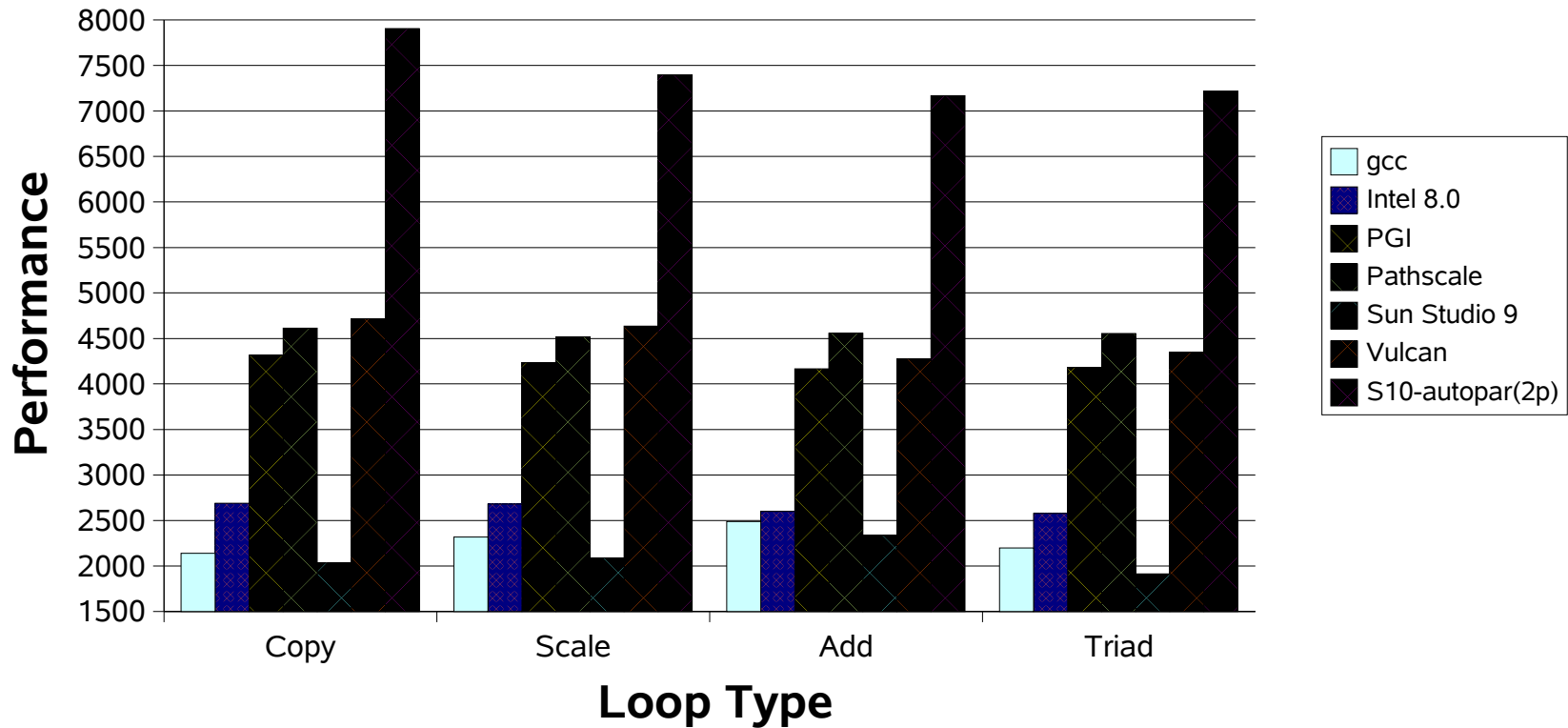
SPEC Performances



- 2.4GHz, AMD Opteron 150
- Studio 10 Early Access, SpecINT=1301, SpecFP=1365, GCC, PGI, PathScale from www.spec.org

STREAM autopar

STREAM Benchmark



Reliable Computing Interval Arithmetic



Floating Point Precision?

Rump's Counterexample: Inherent Instability

- Increasing precision does not necessarily expose errors or increase accuracy
- The same incorrect answer can be obtained using single, double, and quadruple precision
- For $a = 77617$ and $b = 33096$ the equation

$$f = (333.75 - a^2)b^6 + a^2(11a^2b^2 - 121b^4 - 2) + 5.5b^8 + \frac{a}{2b}$$

- produces
 - 32-bit: $f = 1.172604$
 - 64-bit: $f = 1.1726039400531786$
 - 128-bit: $f = 1.1726039400531786318588349045201838$

- The correct result is

$$f = -0.827396059946821368141165095479816...$$

Computing Paradigms

- Floating Point Numbers with Finite Precision
 - Representing measurement uncertainty is inconvenient
 - Rigorously propagating uncertainty and error is difficult
 - Numerical disconnection from continuous Mathematics/Science/Engineering ...
 - “Speed at any price – Unsafe at any speed”

Computing Paradigms

- Floating Point Numbers with Finite Precision
 - Representing measurement uncertainty is inconvenient
 - Rigorously propagating uncertainty and error is difficult
 - Numerical disconnection from continuous Mathematics/Science/Engineering ...
 - “Speed at any price – Unsafe at any speed”
- Performance =
Rate of progress toward the correct answer

Computing Paradigms

- Floating Point Numbers with Finite Precision
 - Representing measurement uncertainty is inconvenient
 - Rigorously propagating uncertainty and error is difficult
 - Numerical disconnection from continuous Mathematics/Science/Engineering ...
 - “Speed at any price – Unsafe at any speed”
- Performance =
 - Rate of progress toward the correct answer
- Interval Arithmetic = Reliable Computing
 - Interval $[a,b]$: a continuous set bounded by its endpoints
 - Arithmetic operations $\{+,-,x,/ \}$ on intervals must produce an interval result of all possible values; cset is the smallest
 - Intervals are continuous sets: bound error and uncertainty!

Interval Arithmetic

- What can be done with intervals?
 - Compute interval bounds on the set of all possible solutions
 - Integrate ordinary differential equations
 - Solve nonlinear functions, systems of nonlinear equations
 - Find the global minimum of a nonlinear function
 - Generic interval algo are embarrassingly/massively parallel
- Reliable Computing community
 - <http://interval.louisiana.edu>

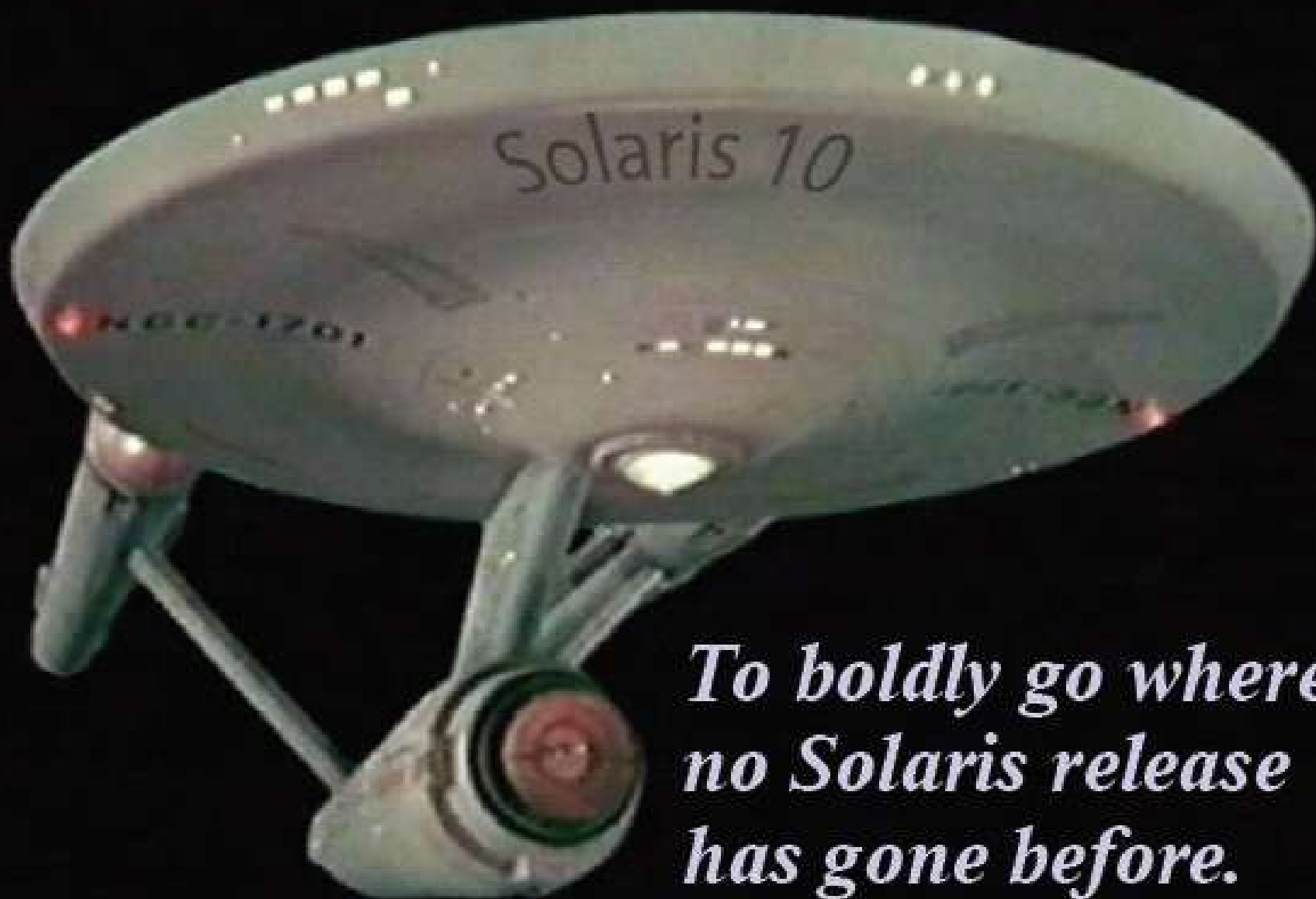
Interval Arithmetic

- What can be done with intervals?
 - Compute interval bounds on the set of all possible solutions
 - Integrate ordinary differential equations
 - Solve nonlinear functions, systems of nonlinear equations
 - Find the global minimum of a nonlinear function
 - Generic interval algo are embarrassingly/massively parallel
- Reliable Computing community
 - <http://interval.louisiana.edu>
- Intervals @ Sun
 - Available with Fortran/C/C++ since Forte 6 compilers
 - Special HW support on SPARC ISA
 - Tutorials at
http://developers.sun.com/prodtech/cc/numerics_index.html
 - Contact our Evangelist at Sun Labs: bill.walster@sun.com



Solaris 10





*To boldly go where
no Solaris release
has gone before.*

Solaris 10: A Generation Ahead

Extreme

Performance

*Dynamic Tracing
Network
Entry systems*

Unparalleled Security

*Process Rights Management
Crypto Infrastructure*

Optimal Utilization

Solaris Containers

Relentless Availability

*Predictive Self Healing
Next-gen file services*

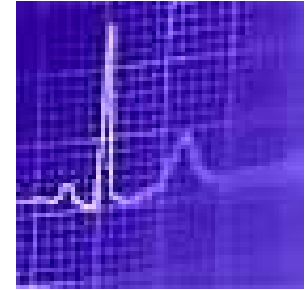


Platform Choice

*New UltraSPARC IV,
New AMD Opteron*

Dynamic Tracing

Real-time Analysis and Diagnosis



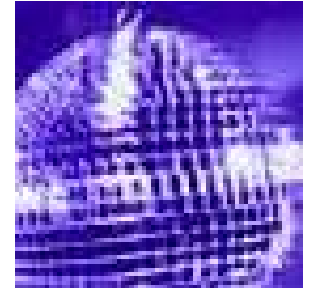
- Safe and comprehensive
 - Global view into systems, apps
 - 30,000+ probe points
- Extensible; easy to use
 - Scriptable for re-use
 - Provider model allows for new probes
- Built for live use on production systems
- Reduced costs
 - Solutions in minutes or hours, not days or weeks
 - Optimization: cases of 3-30x speedups already seen
- Have a look at www.sun.com/bigadmin/content/dtrace

“...it's like they *saw inside my head* and gave me The One True Tool.”

-Slashdot post, November '03

Solaris ZFS

Next-Generation Filesystem



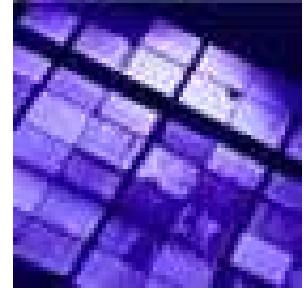
- Complete redesign
 - Pooled storage: no more volume mgmt
 - End-to-end data integrity: checksums at all levels
 - Everything is transactional
 - Dynamic metadata
 - 128-bit addressing: $256 * 10^{15} * \text{ZB!}$
- Benefits for users and sysadmins
 - Simple administration, self-healing
 - Copy-on-write design allows unlimited snapshots
 - Data security: ACL, encryption, checksum authentication ...
 - Performances: dynamic striping, multiple block size ...
- Have a look at www.sun.com/software/solaris/10/

Containers and other stuff



- Solaris Containers
 - Effective, HW-independent approach to virtualization
 - 1000s of isolated “zones” under the same OS kernel
 - Dynamic resources assignment

Containers and other stuff



- Solaris Containers
 - Effective, HW-independent approach to virtualization
 - 1000s of isolated “zones” under the same OS kernel
 - Dynamic resources assignment
- System Performance
 - New IP software stack; goal: saturate 10Gb Eth
 - Memory Placement Optimization for NUMA platforms

Containers and other stuff



- Solaris Containers
 - Effective, HW-independent approach to virtualization
 - 1000s of isolated “zones” under the same OS kernel
 - Dynamic resources assignment
- System Performance
 - New IP software stack; goal: saturate 10Gb Eth
 - Memory Placement Optimization for NUMA platforms
- Linux Binary Compatibility
 - Direct support of Linux kernel I/F



Containers and other stuff



- Solaris Containers
 - Effective, HW-independent approach to virtualization
 - 1000s of isolated “zones” under the same OS kernel
 - Dynamic resources assignment
- System Performance
 - New IP software stack; goal: saturate 10Gb Eth
 - Memory Placement Optimization for NUMA platforms
- Linux Binary Compatibility
 - Direct support of Linux kernel I/F
- Platform Choice
 - SPARC, AMD64, traditional x86



Thanks!

carlo.nardone@sun.com

cell: 335 5828197

The Network is the Computer

