JavaNumerics

Report of the Java Grande Forum Numerics Working Group

Ronald F. Boisvert and Roldan Pozo
NIST Information Technology Laboratory

SC'98, Orlando, Nov. 13, 1998

Working Group Charter

- ✓ evaluate the suitability of Java for numerical applications
- ✓ voice community consensus on needed changes to language, environment
- ✓ communicate needs to Sun
- ✓ catalyze development of standard APIs for core numerical functions

Group Organization

- ✓ Co-chairs from NIST
  - Ron Boisvert (boisvert@nist.gov)
  - Roldan Pozo (pozo@nist.gov)
- ✓ Open meetings with Java Grande
  - March, May, August 1998
- ✓ Communication via
  - javagrandeforum@npac.syr.edu
  - http://math.nist.gov/javanumerics/

Meeting Participants

- ✓ Industry (11)
  - IBM, Mantos Consulting, The MathWorks, NAG, Sun, Visual Numerics
- ✓ Academia (8)
  - Loyola Chicago, Syracuse, TU Delft, UC Berkeley, U. Houston, UNC, U. Tenn.
- ✓ Government (2)
  - NIST

... and input from many others

Initial Group Products

- ✔ proposal for extensions to Java to support numerical computing
- ✔ response to Sun proposal for extension of Java floating-point semantics
- ✔ development of proposed APIs
- ✔ development of numerical benchmark

Critical Needs for Numerics

- ✔ complex arithmetic
- ✔ multidimensional arrays
- ✔ lightweight classes
- ✔ operator overloading
- ✔ improved floating-point

Issue: Complex

- ✔ Requirements …
  - efficient as **`floats`** and **`doubles`**
  - convenient as **`floats`** and **`doubles`**
- ✔ A standard class provides neither
  - object overhead unacceptable
  - semantics of = and == are wrong
  - inscrutable code from method calls, e.g. **`a.assign(b.times(c).plus(d))`**

Issue: Multidimensional Arrays

- ✔ Requirements …
  - array operations must be optimizable
  - reliable model of layout needed for programmers to select best algorithm
- ✔ Problems …
  - Java arrays jagged, rows can be aliased
  - standard API would mean unreadable code, inefficiencies

A Common Solution

✓ Lightweight Classes
  - restricted class with value semantics
  - low overhead, can be inlined

✓ Operator Overloading
  - at least for existing operators: arithmetic, compare, assign, subscripts

✓ Admits extensions
  - interval, multiprecision, ...

Issue: Floating-point

- ✔ Requirements …
  - good performance on most processors
  - high performance *sometimes*
  - exact reproducibility *sometimes*
  - access to IEEE features
- ✔ Problems …
  - 80-bit registers on x86 ⇒ 2-10x slower
  - use of fused multiply-add denied
  - optimizations, e.g. associativity, denied

Floating-point Proposals

- Sun: **widefp** (default) & **strictfp**
  - **widefp** admits indiscriminant use of float- and double-extended
  - **strictfp** is current mode
- JGF: default, **strictfp**, **associativefp**
  - default allows wide (15-bit) exponent range, fused multiply-add
  - **associativefp** admits associativity, more?

Additional recommendations

- ✔ "Reproducibility" of math functions
  - Java version of fdlibm for **`strictfp`**
- ✔ Standard class libraries to get, set
  - IEEE floating-point flags
  - IEEE rounding modes
- ✔ Implementation of IEEE functions
  - Darcy has recently developed these

Core Numerical APIs

Strawman proposals now available

- ✔ Complex (VNI)
- ✔ Multidimensional Arrays (IBM)
- ✔ Linear Algebra (MathWorks & NIST)
- ✔ Special Functions (VNI)

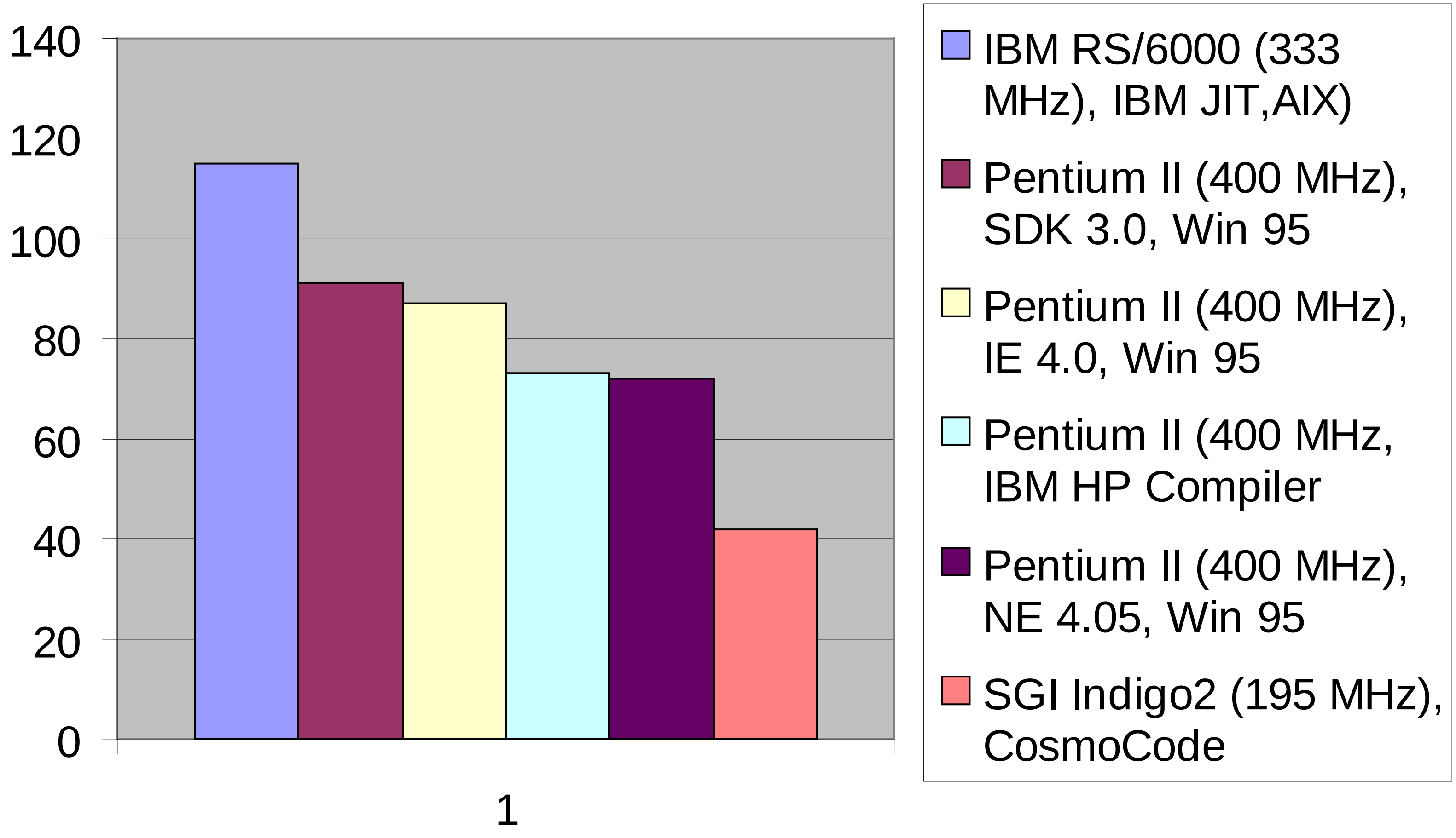
coming: intervals, FFTs, multiprecision, ...

SciMark Benchmark

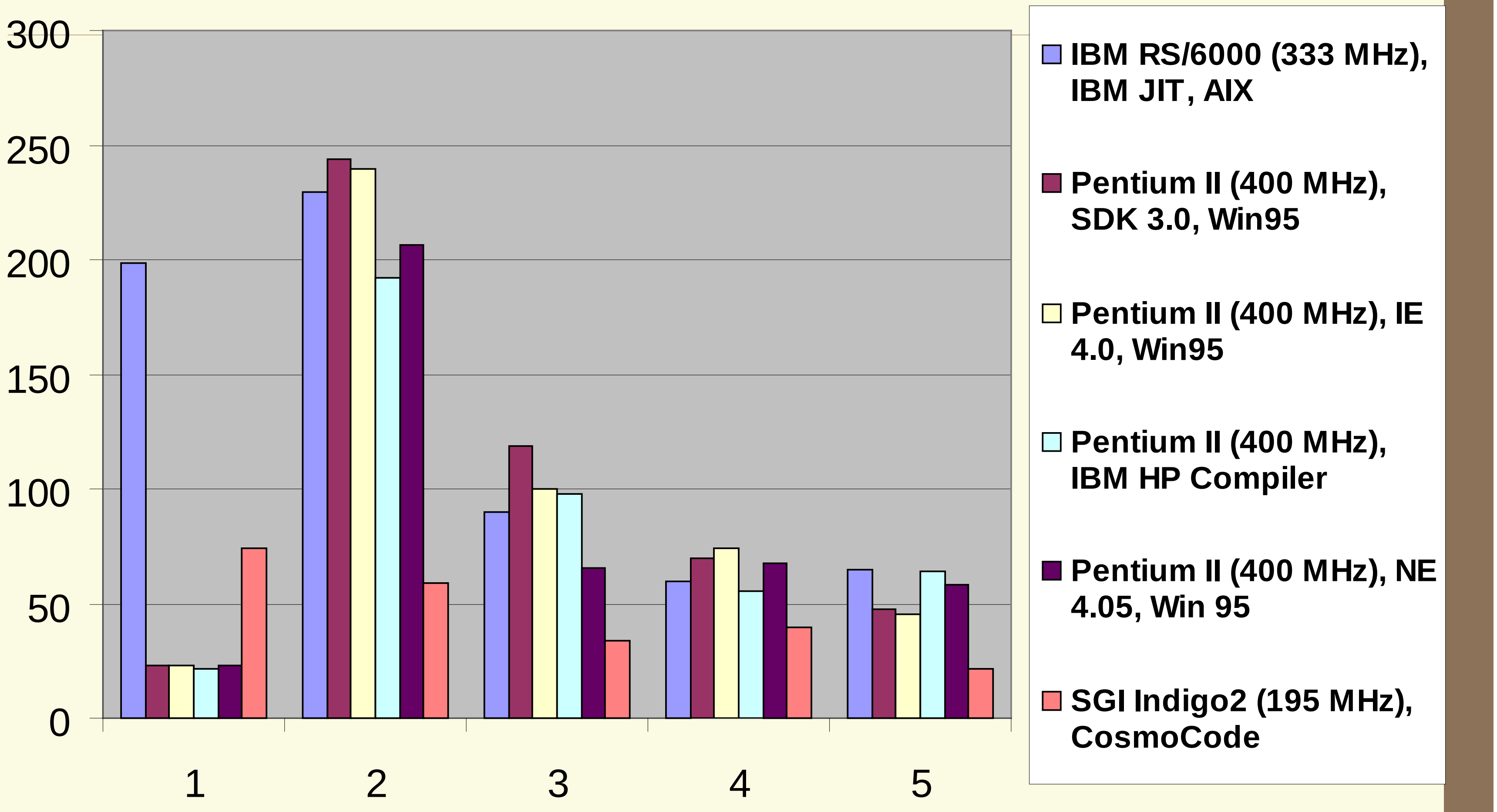
Roldan Pozo, Bruce Miller and Karin Remington, NIST

- ✔ Java benchmark applet for numerics
- ✔ composite of results for five kernels
  - FFT (4K, complex)
  - Gauss-Seidel relaxation (100x100 grid)
  - Monte Carlo integration (25K samples of exp(-x^2))
  - sparse matrix multiply (order 5K, 25K random nonzeros)
  - dense LU factorization with pivoting (100x100)
- ✔ normalized to SPARC 10 / Netscape 4.04

SciMark Composite Results



SciMark Component Results



Next Steps

- ✔ refine proposals for improvements
- ✔ continue to lobby Sun
- ✔ serious review of proposed APIs
- ✔ development of new APIs
- ✔ large-scale benchmark

We Need You!

- ✓ http://math.nist.gov/javanumerics
- ✓ javagrandeforum@npac.syr.edu
- ✓ boisvert@nist.gov
- ✓ pozo@nist.gov