Performance and Scalability of the NAS Parallel Benchmarks in Java

Michael Frumkin, Matthew Schultz
Haoqiang Jin, Jerry Yan
NASA Advanced Supercomputing Division
NASA Ames Research Center

{frumkin,hjin,yan}@nas.nasa.gov

The Problem

- Evaluate Java for use in CFD applications
- Evaluate platforms for CFD-in-Java
- Understand FORTRAN -> Java translation

The NAS Parallel Benchmarks

- Block Tridiagonal (BT)
- Scalar Pentadiagonal (SP)
- LU Decomposition (LU)

Simulated applications

- 3-dimensional FFT (FT)
- Conjugate Gradient (CG)
- V-cycle multigrid (MG)
- Integer Sort (IS)
- Embarrassingly Parallel (EP)

Kernel benchmarks

Classes: S,W,A-D (grid sizes: 12-1024)

Applications of the NPB

- Evaluation of the hardware for CFD codes
- Evaluation of the compilers and tools
- Demonstration of the programming paradigms (MPI, OpenMP, HPF, Java,…)
- Building blocks for the NAS Grid Benchmarks

FORTRAN, OpenMP, HPF, and Java versions are available from www.nas.nasa.gov/Research

NPB in Java on 5 architectures

- BT, SP, LU, FT, MG, CG, IS
- Performance results on:

- IBM p690 (POWER 4)	32 procs
- SGI Origin (2000-3000)	32 procs
- SUN Enterprise 10000	16 procs
- Intel P4 (Linux)	2 procs
- Apple Xserv (OS X)	2 procs

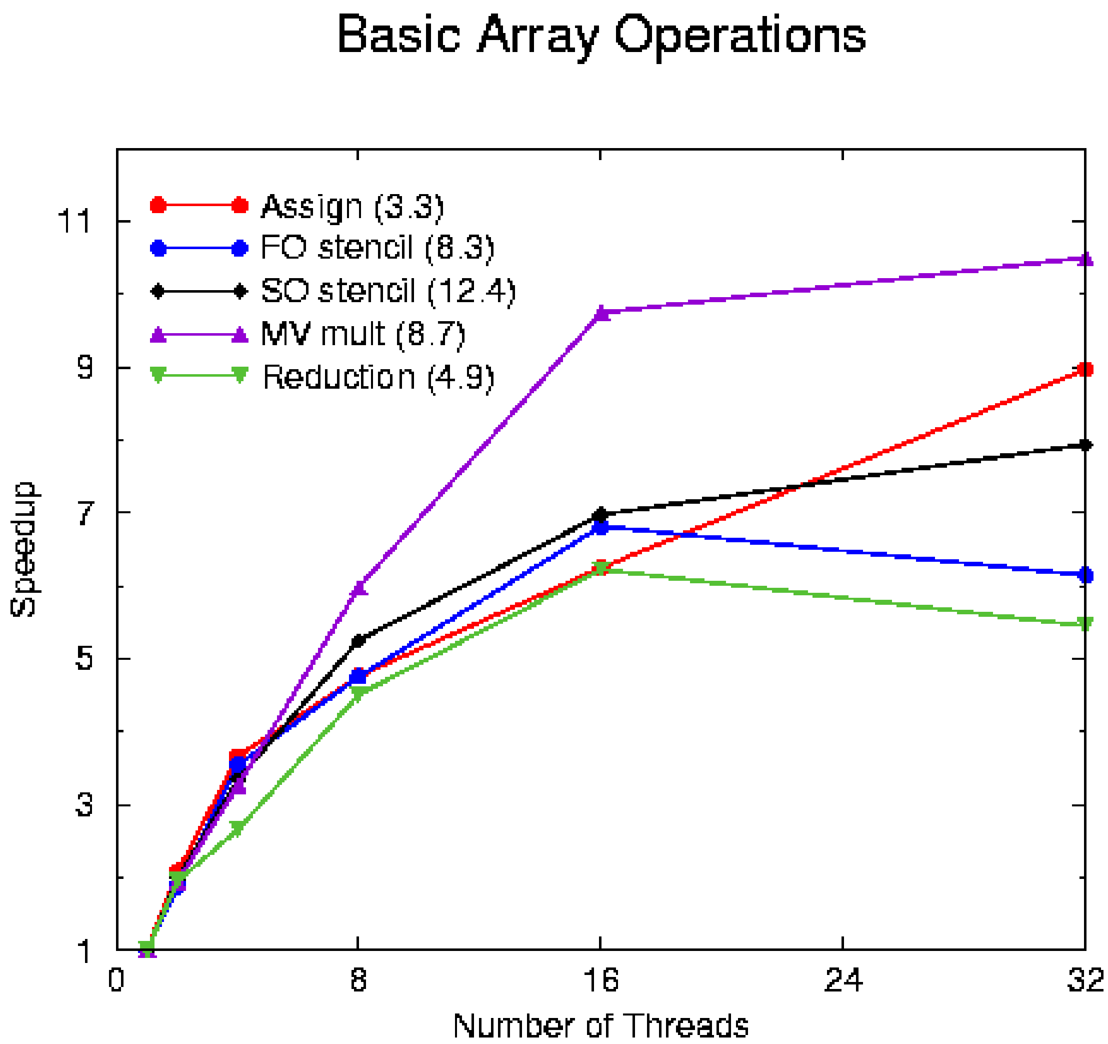
FORTRAN → Java translation

- Literal translation
- Array linearization
- Semiautomatic (using regular expressions)
- Multithreading:
  - in outermost loops in the nests
  - pipelines in LU

Baseline for the translation

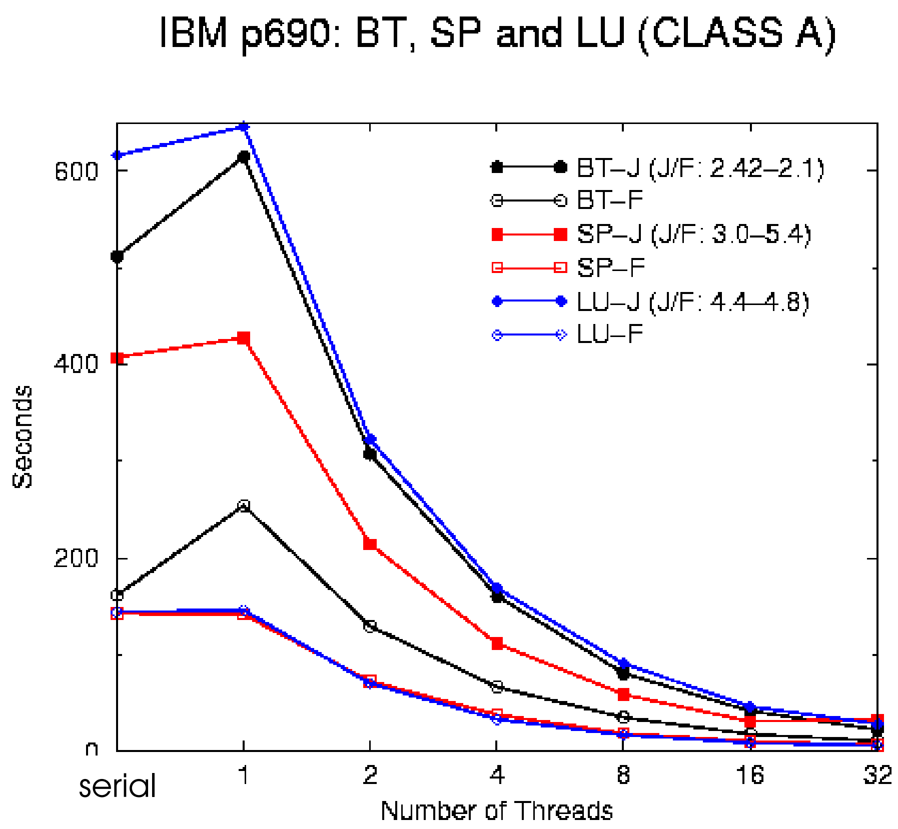
- Basic CFD array operations set baseline for the efficiency of the translated code
  - Assignment
  - First and second order stencil
  - Matrix vector multiplication
  - Reduction sum

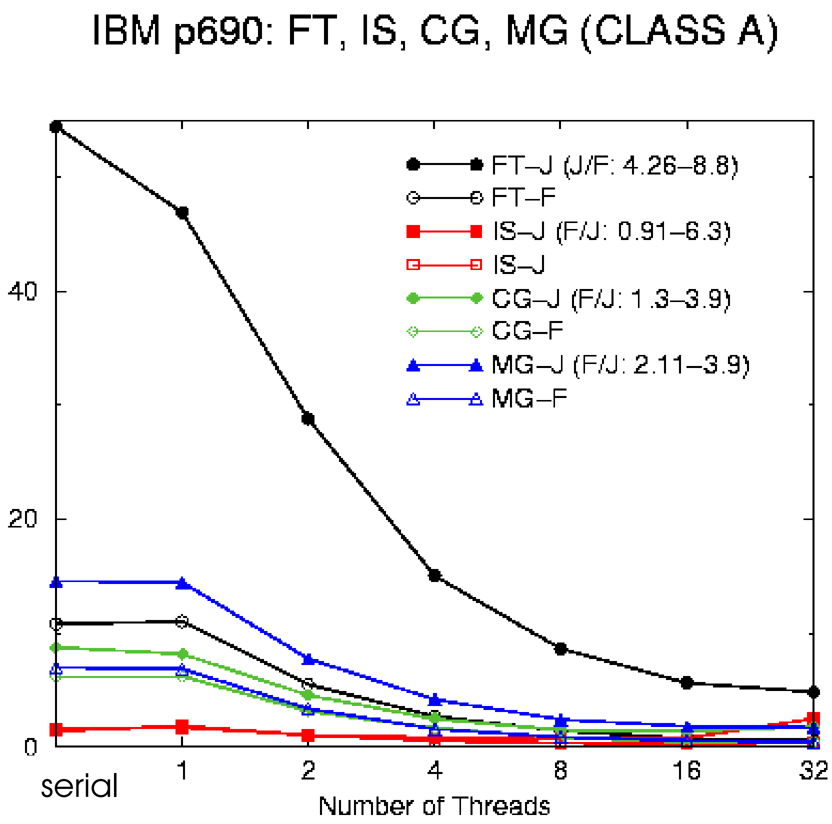
Performance of the basic CFD operations



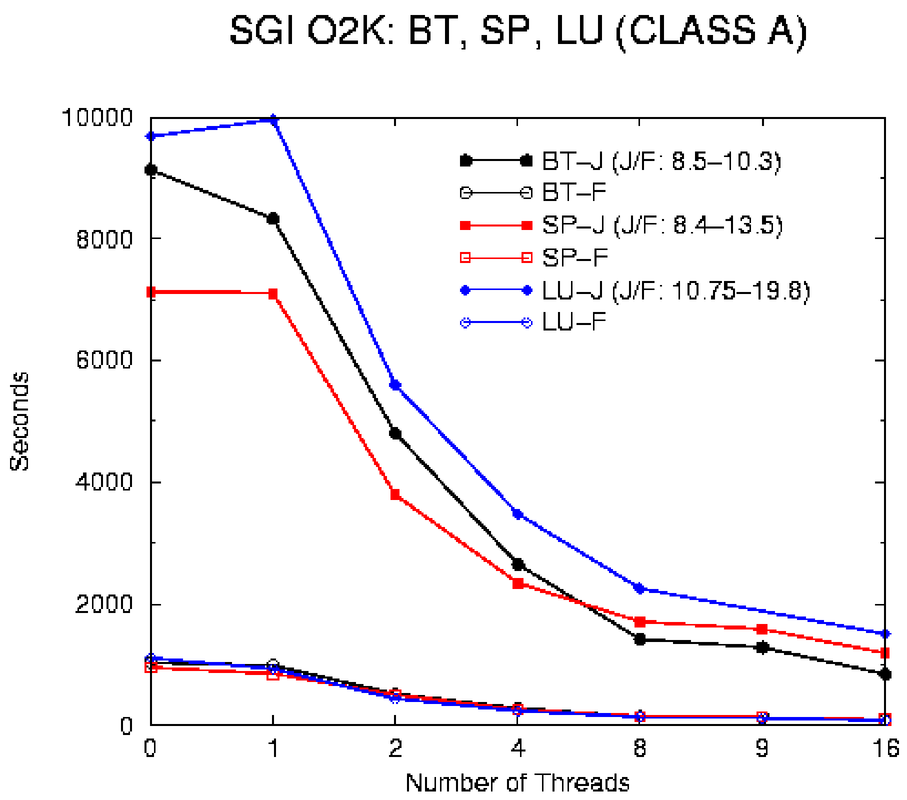
SGI Origin 2000
250 MHz
32 proc, 8 GB
SGI Java 1.1.8
81x81x100 mesh

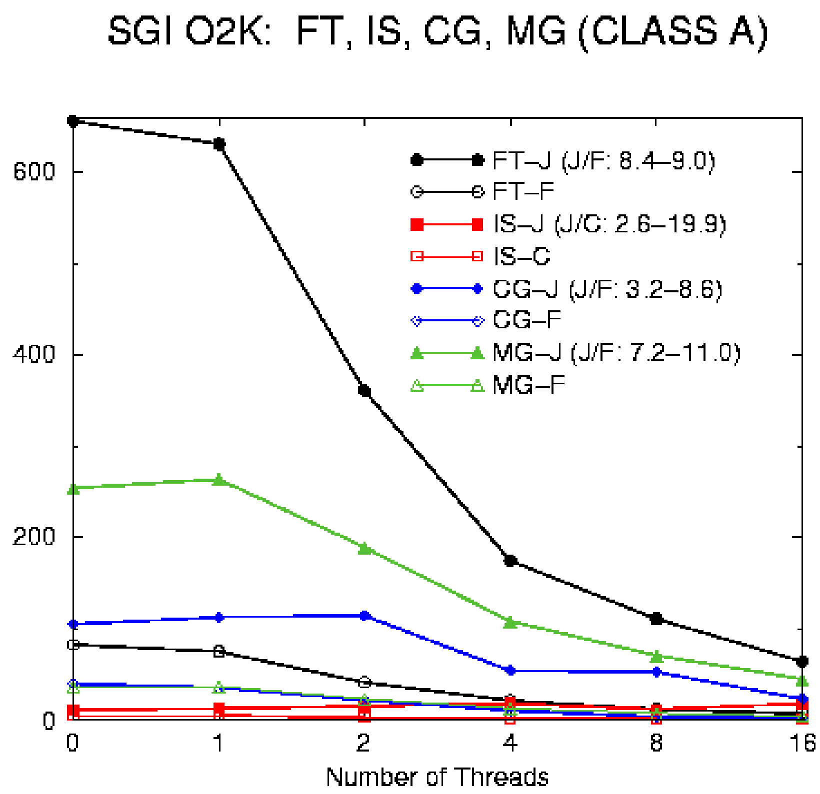
IBM p690 (1.3 GHz, 32 procs, 32 GB), Java 1.3.0



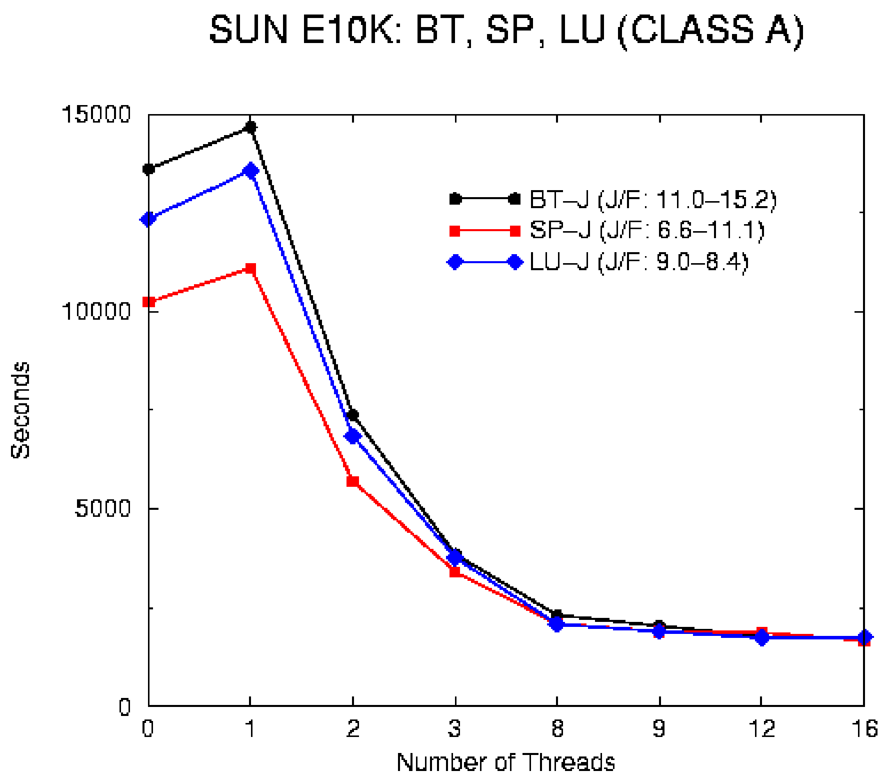


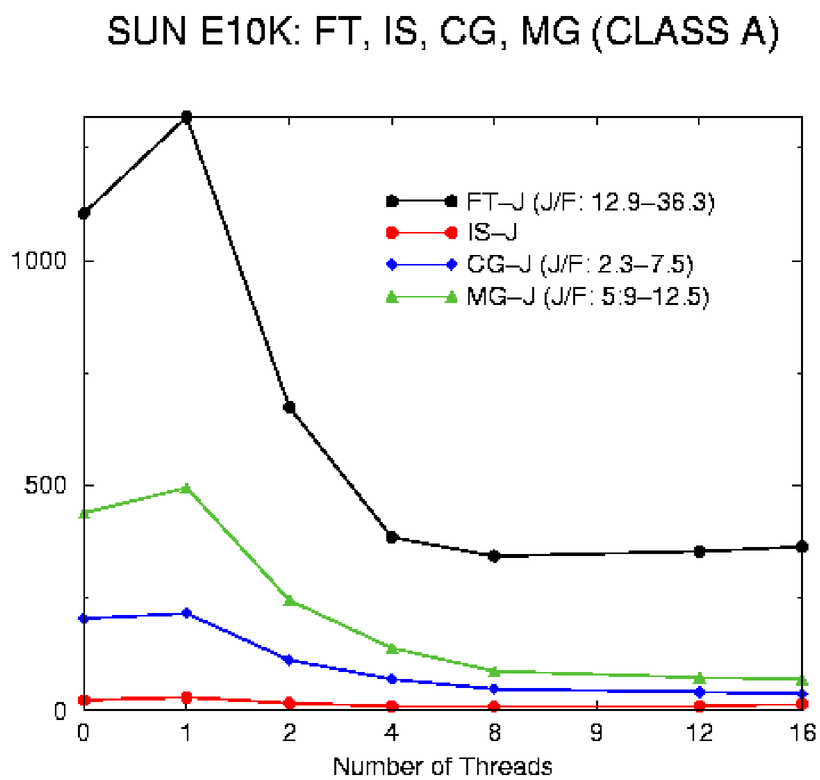
SGI Origin 2000 (250 MHz, 32 procs, 8 GB), Java 1.1.8



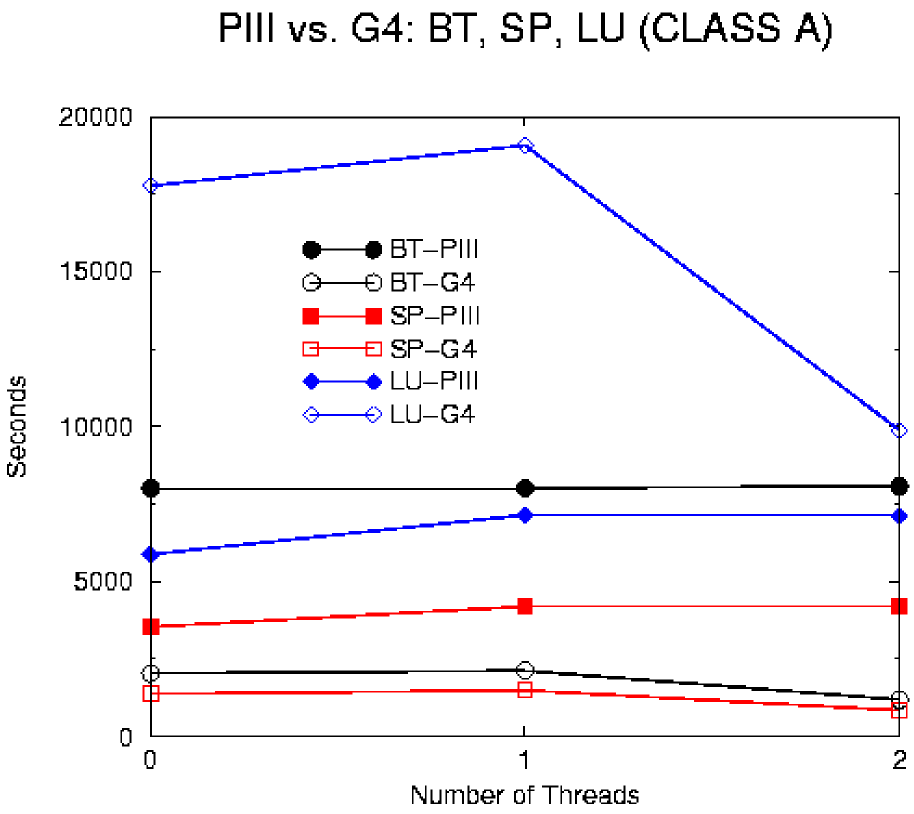


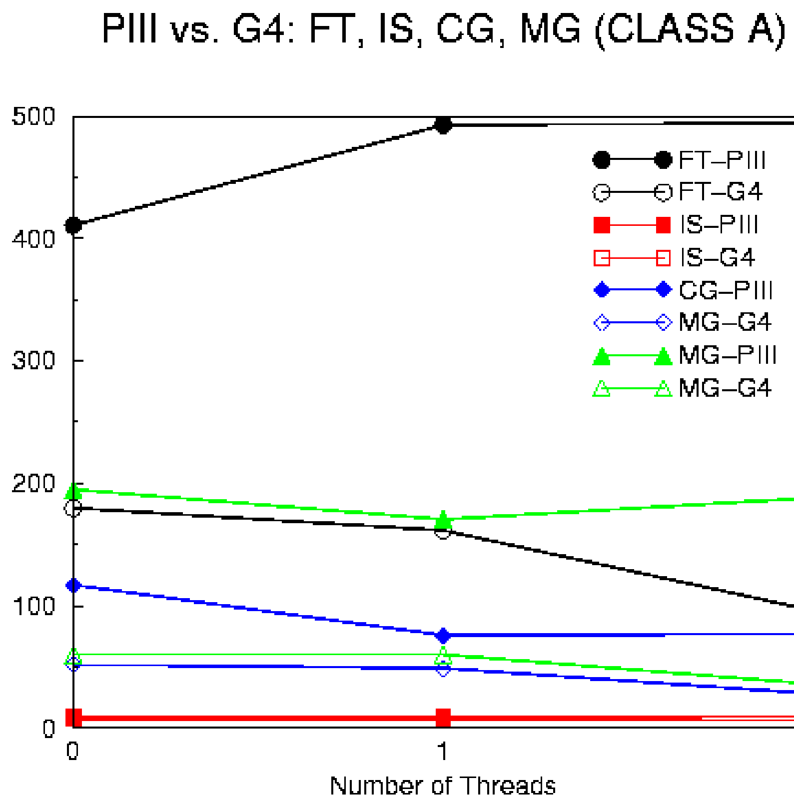
Sun E-10000 (333 MHz, 16 proc, 8 GB), Java 1.1.3





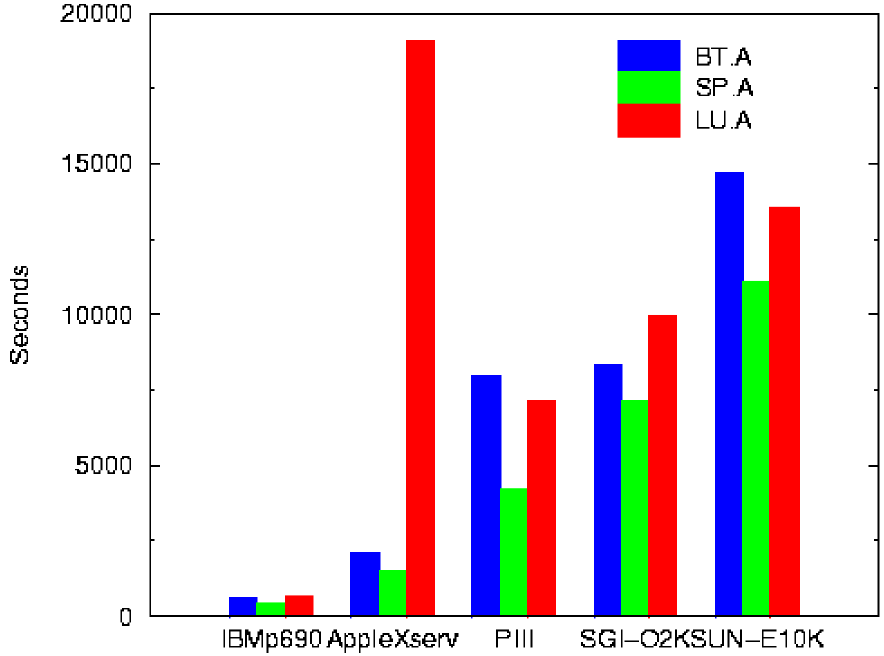
PIII vs. Apple-Xserv (G4), Java 1.3.0



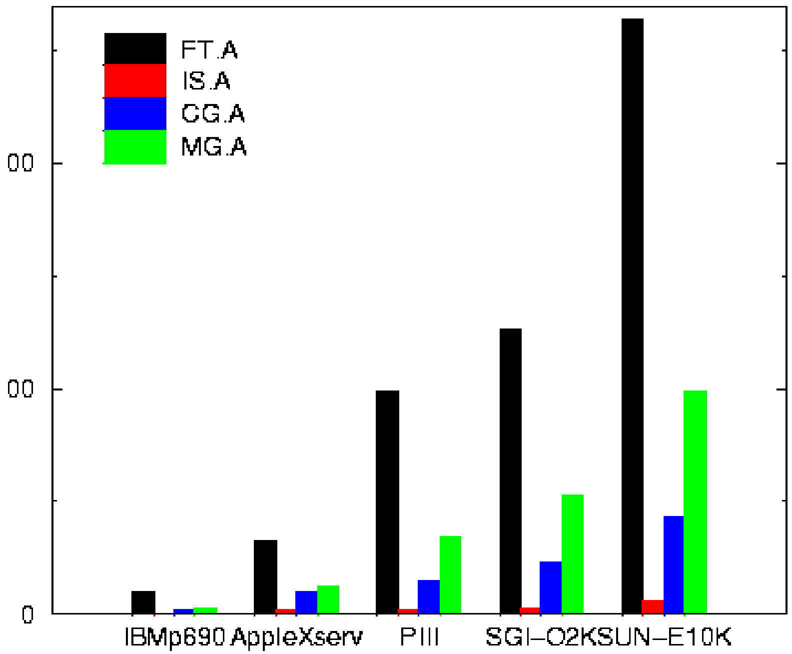


Performance and Scalability Summary

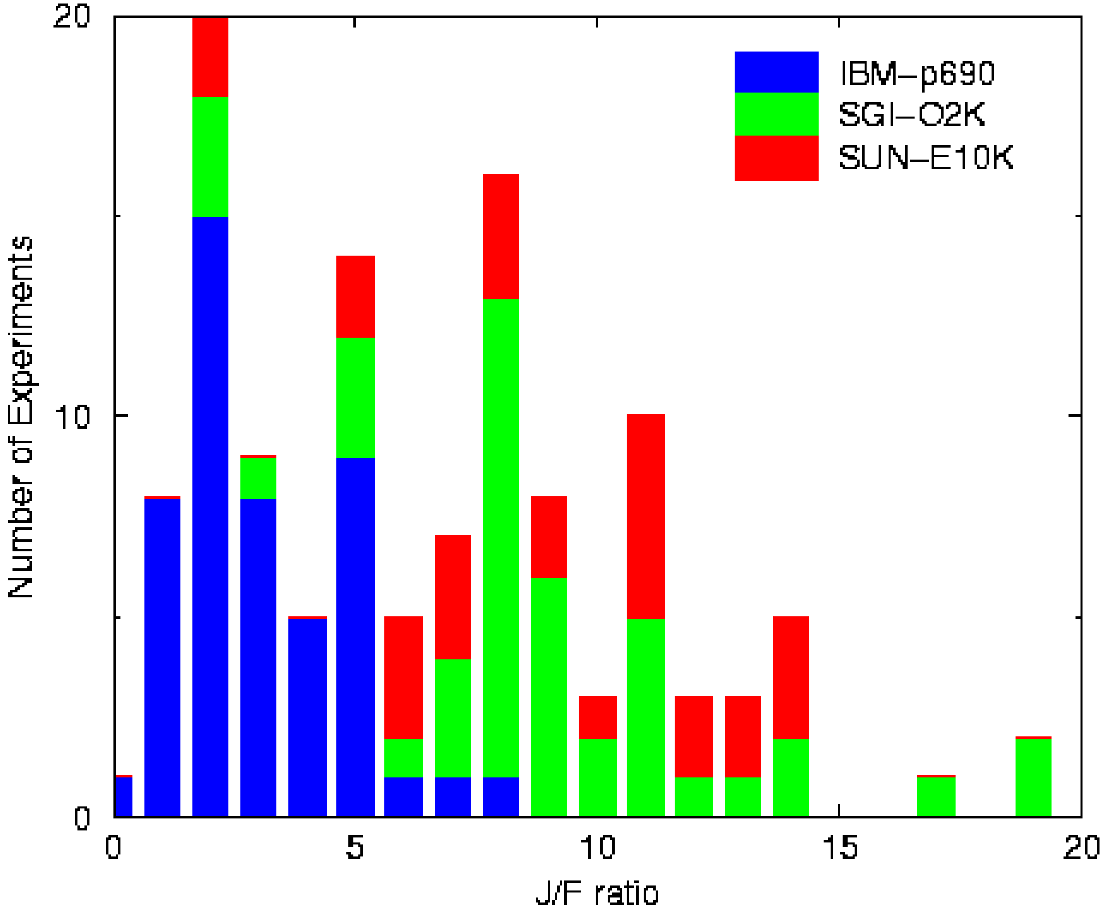
BT, SP, LU: across machines



FT, IS, CG, MG: across machines



J/F Performance Ratio

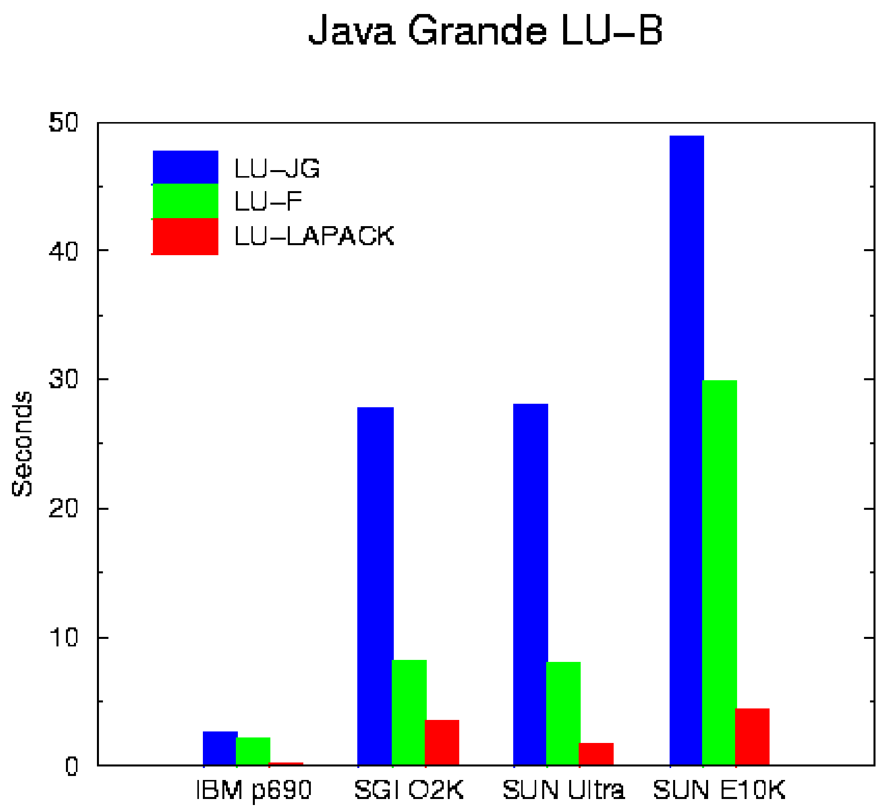


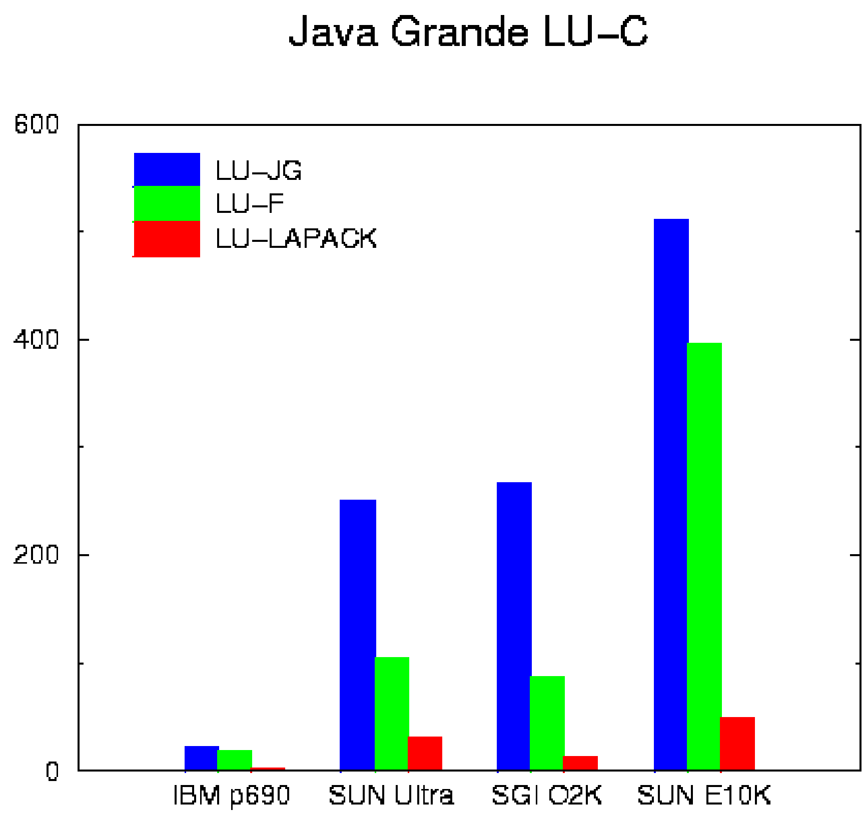
NPB vs. Java Grande Benchmarks

www.epcc.ed.ac.uk/computing/research_activities/java_grande

	J/F ratio	Speedup
JG Benchmarks	1-2	--
NPB3.0-jav	1-19	35%

Grande LU Benchmark





Java Performance Issues

- Array access overhead (boundary checking)
- Data overhead (moving more data around)
- Instructions overhead (J/F instructions ~ 10)

Conclusions

- Compile once – run everywhere
- Acceptable performance
- Expressive language
- Build-in multithreading
- FORTRAN->Java translation is simple

vs.

- Median J/F performance is 5-6
- Scalability is 35% on 10-30 processors