# NAS Parallel Benchmarks


Subhash Saini and Jack Dongarra

Numerical Aerodynamic Simulation Program
NASA Ames Research Center
Mail Stop T27A-1
Moffett Field, California 94035-1000


February 19963

# Challenges of Parallel Computing Benchmarking, 1990

- **Different architectures:** vector, SIMD distributed memory, and MIMD distributed.
- **Different programming models:** vector loops, data parallel, and message passing.
- **Different programming languages:** Fortran 77, CM Fortran, Lisp, and Intel NX.
- Different algorithms are optimal on different systems.
- Different implementations are optimal on different systems.

# Challenges of Parallel Computing Benchmarking, 1990 (Cond.)

- Vendors advertised extravagant peak performance figures not achievable on most real-world applications.
- Some scientists exaggerated their success in using the systems.

**Overall picture: chaos**

# Problems with Existing Benchmarks

- Popular source code benchmarks such as the Livermore Loops, LINPACK, and the original NAS Kernels are inappropriate for the performance evaluation of highly parallel machines.
  - Dubious meaning of single processor performance numbers.
  - Small computation and memory requirements.
  - Irrelevant for large parallel applications.

# Problems with Existing Benchmarks

- Popular source code benchmarks such as the Livermore Loops, LINPACK, and the original NAS Kernels are inappropriate for the performance evaluation of highly parallel machines.
  - Dubious meaning of single processor performance numbers.
  - Small computation and memory requirements.
  - Irrelevant for large parallel applications.

# The NAS Parallel Benchmarks

- **Key idea:** "pencil and paper" specification
  - eight problems specified in a a technical document.
- Only the computational task is specified; the implementation can be tailored for a particular computer system.
- The problems do not favor any particular architecture.
- Five "kernels" benchmarks (general numerical problems).
- Three "simulated CFD application" benchmarks.
- Sample source codes of workstation implementations are provided as a starting point for a parallel implementation.

# Kernels and Pseudo-Applications

- The NAS Parallel Benchmarks consist of two parts:

  a. Kernels

  b. Pseudo-applications

  - Kernels alone are insufficient to assess the performance potential of a parallel machine on "real" scientific applications
  - **Difficulty:** A certain data structure may be very efficient on a certain system for one of the isolated kernels, and yet this data structure may be inappropriate if incorporated into a larger application

# The NAS Kernel Benchmarks

- **EP:** Accumulates certain statistics of Gaussian random numbers. Virtually no interprocessors communication.
- **MG:** Performs a simple multigrid calculations. Highly structured, short and long distance communication.
- **CG:** Uses the conjugate gradient method to compute the smallest eigenvalue of a large sparse SPD matrix.
- **FT:** Solves a 3-D PDE using FFTs. Heavy long distance communication.
- **IS:** Performs a sort operation that is important in “particle” codes.

# The NAS Pseudo-Application Benchmarks

- **LU:** Regular-sparse, block (5x5) lower and upper triangular system solution. Typified at NASA Ames by the code INS3D-LU.
- **SP:** Solution of multiple, independent systems of non diagonally dominant, scalar pentadiagonal equations.
- **BT:** Solution of multiple, independent systems of non diagonally dominant, block tridiagonal equations with a (5x5) block size.

**SP and BT are typified at NASA by the code ARC3D.**

# The Latest NPB Results

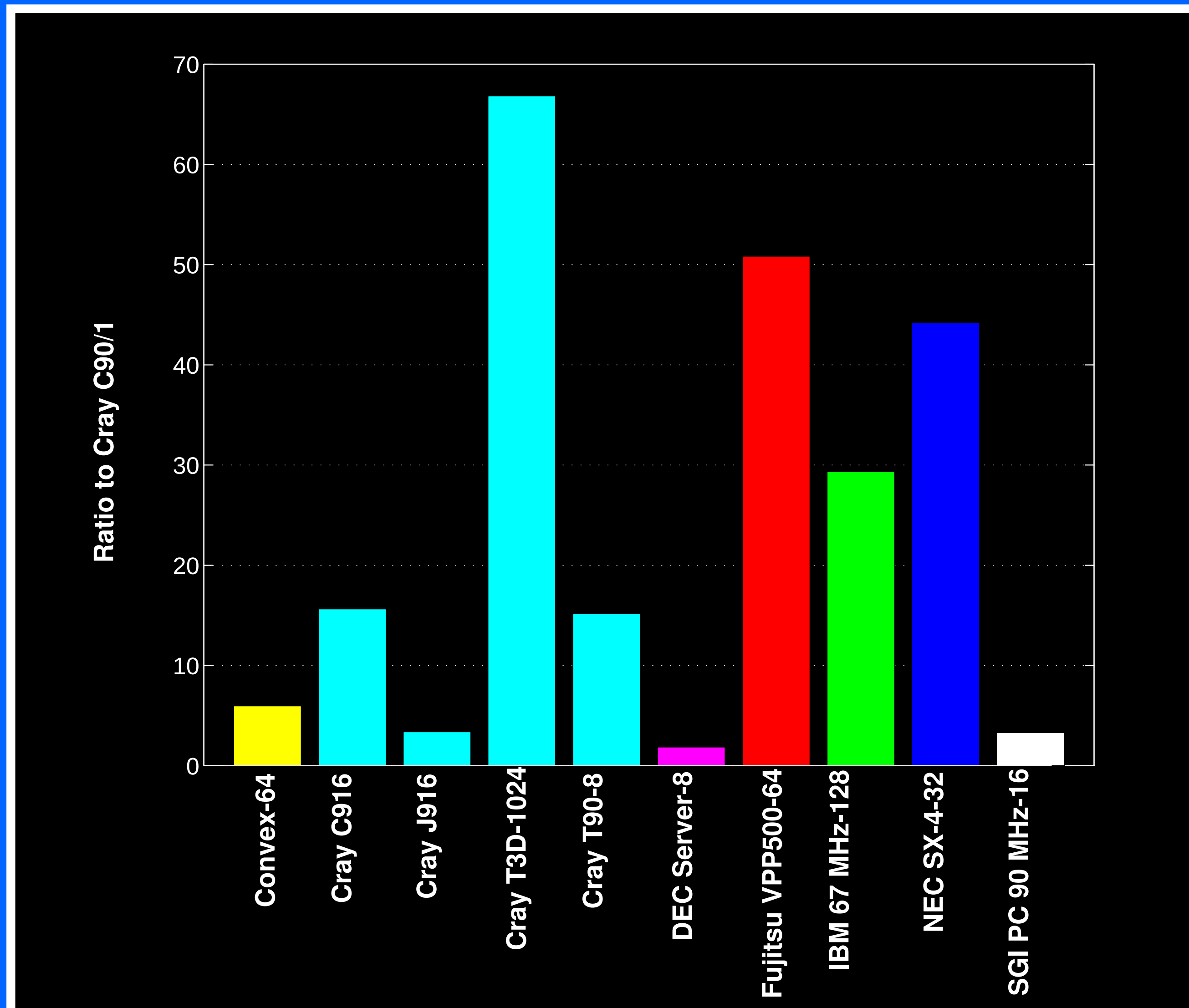
- Timings listed are the best timings that have been submitted to date.
- These timings are all from the respective vendors.
- Complete report available using Mosaic at URL:

```
http://www.nas.nasa.gov/NAS/NPB
```

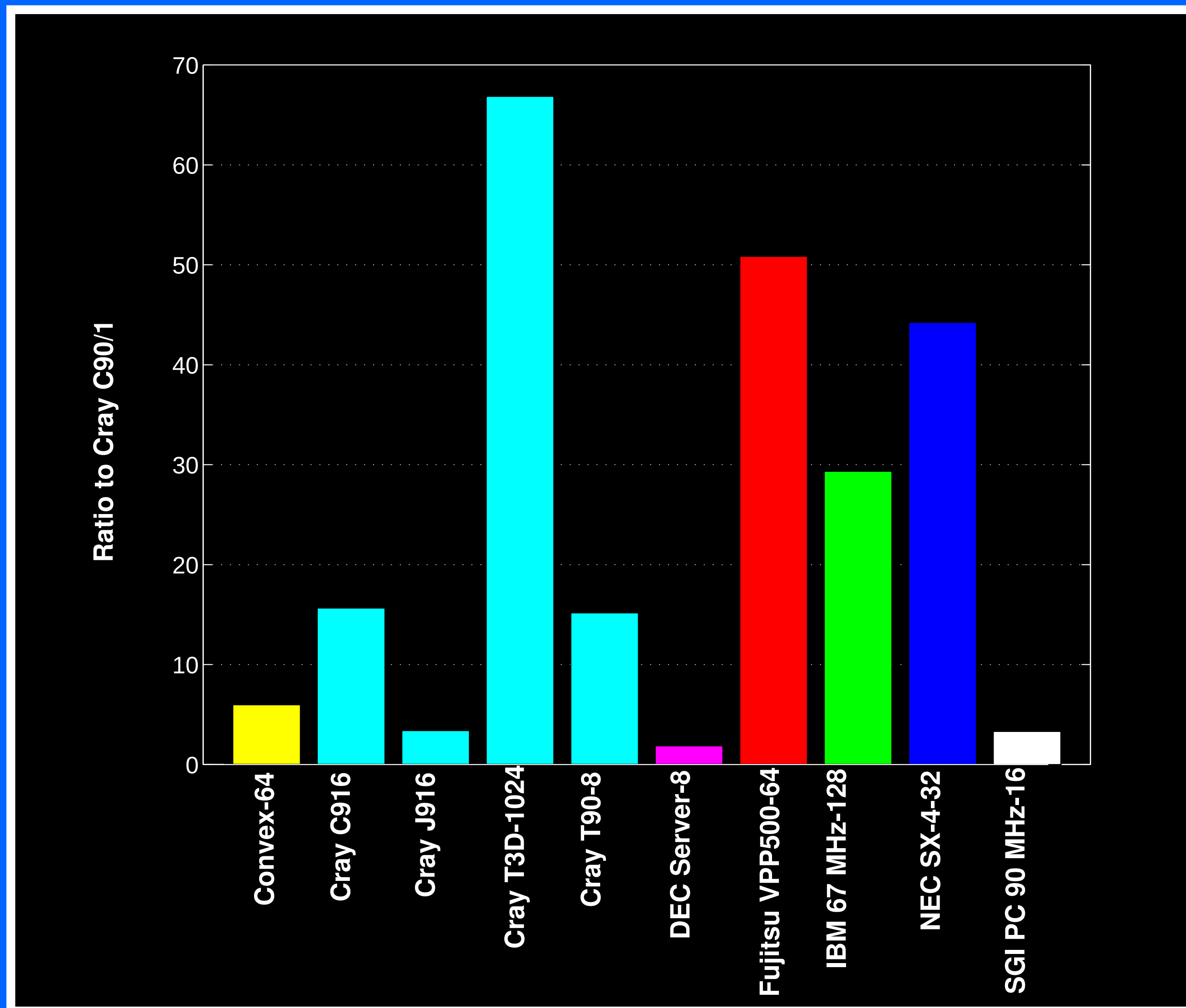
- Results are also available from Park Bench at URL:

```
http://www.netlib.org/parkbench
```

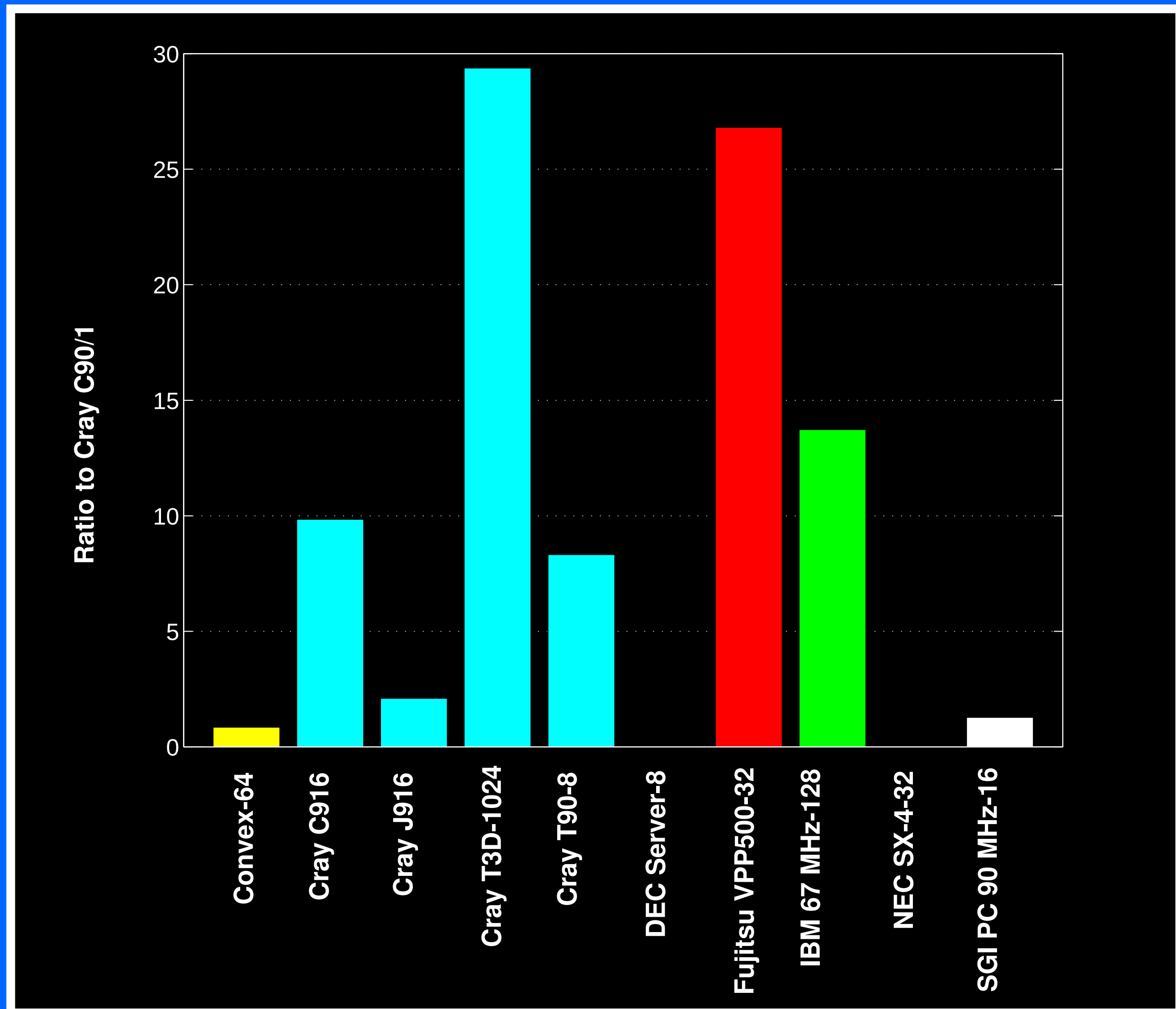
# Performance of Class B EP Benchmark



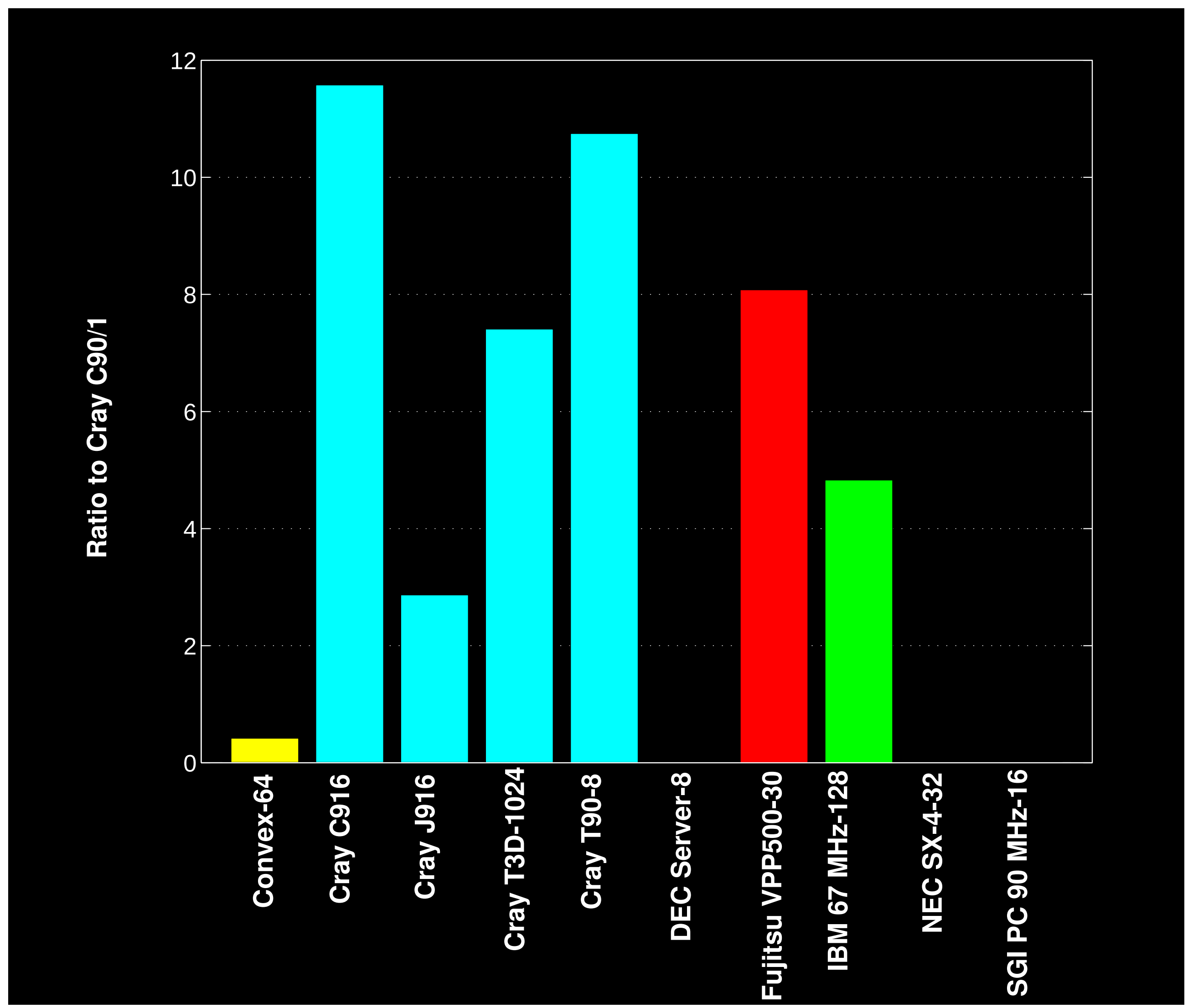
# Performance of Class B EP Benchmark



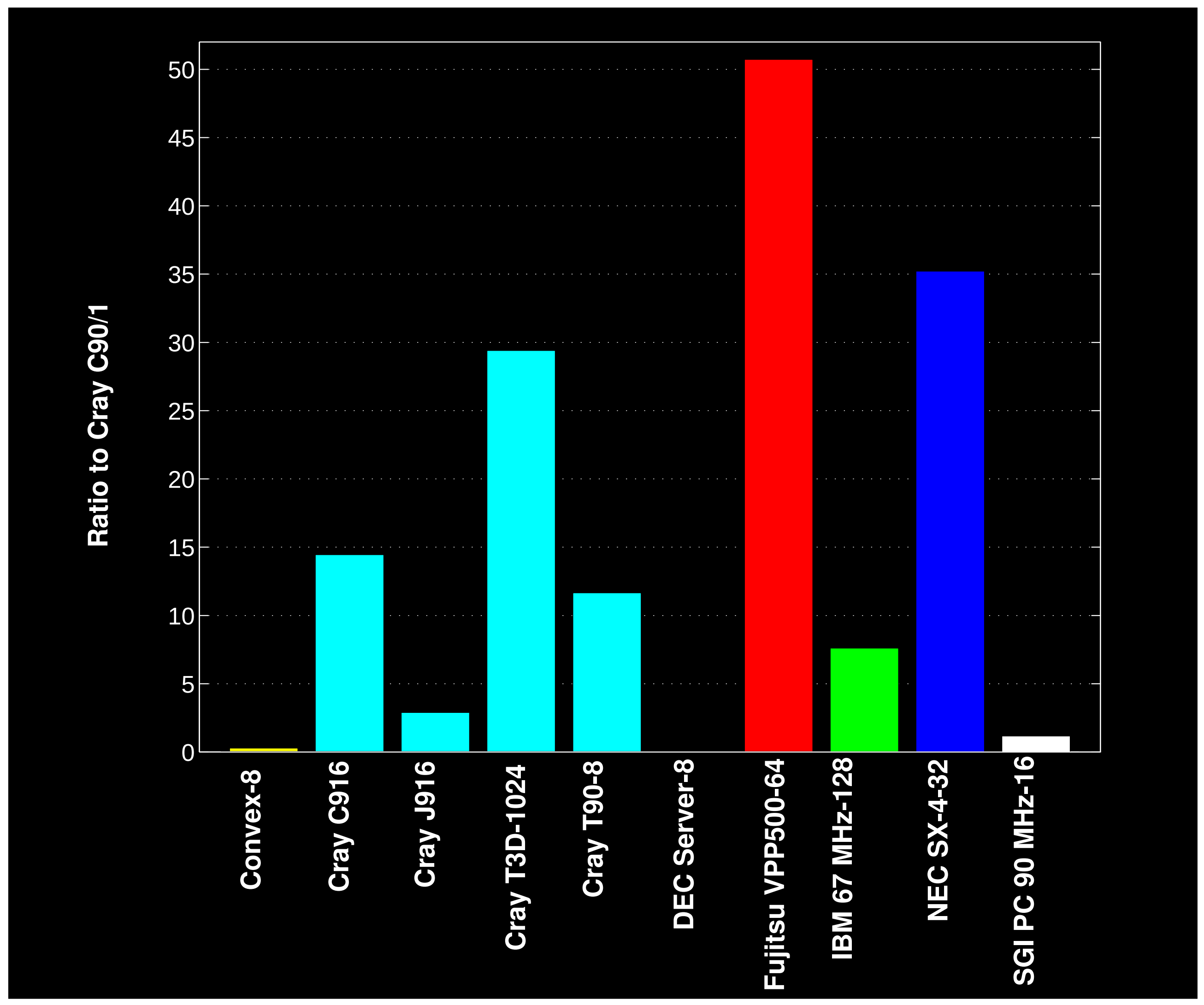
# Performance of Class B MG Benchmark



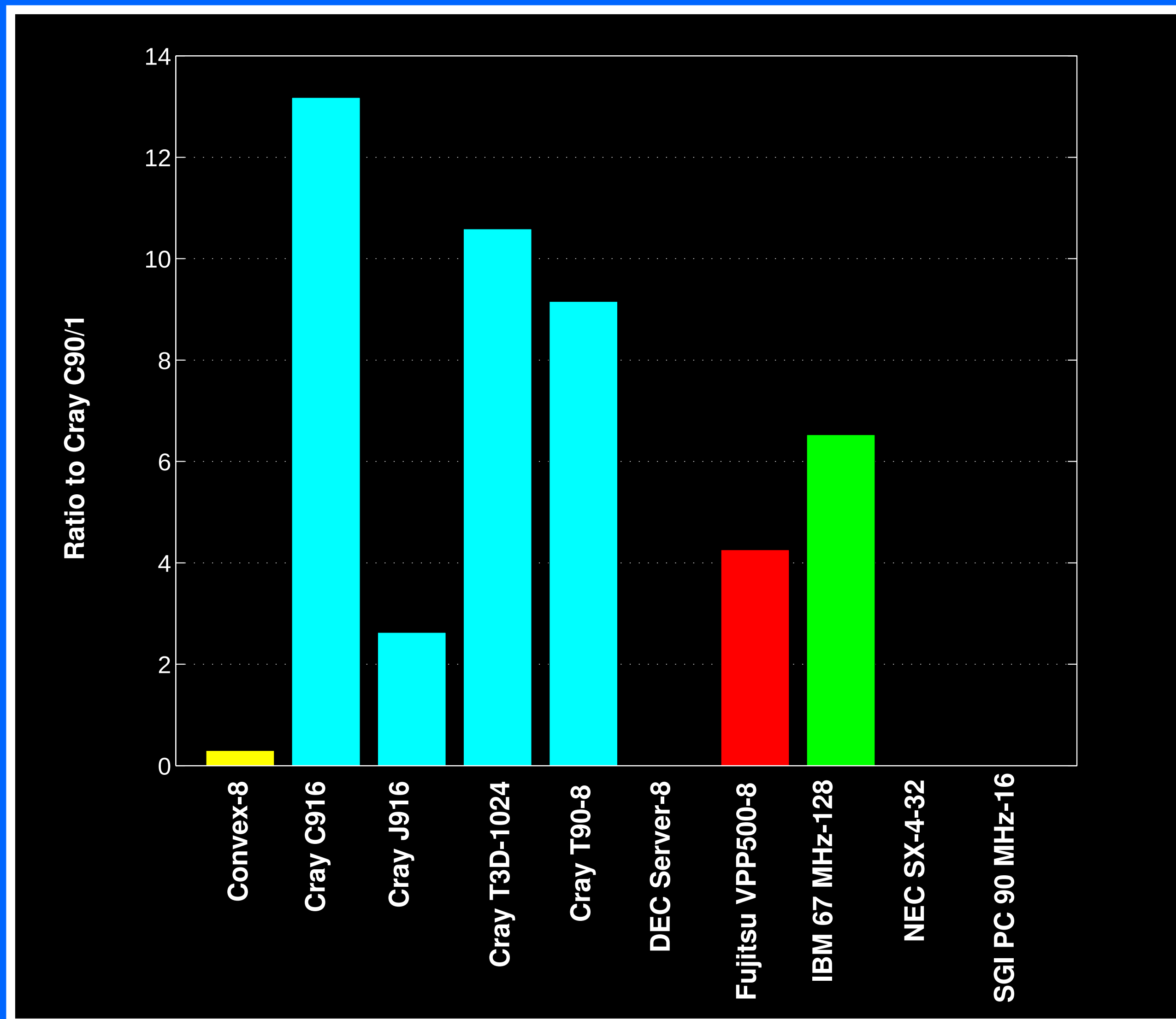
# Performance of Class B CG Benchmark



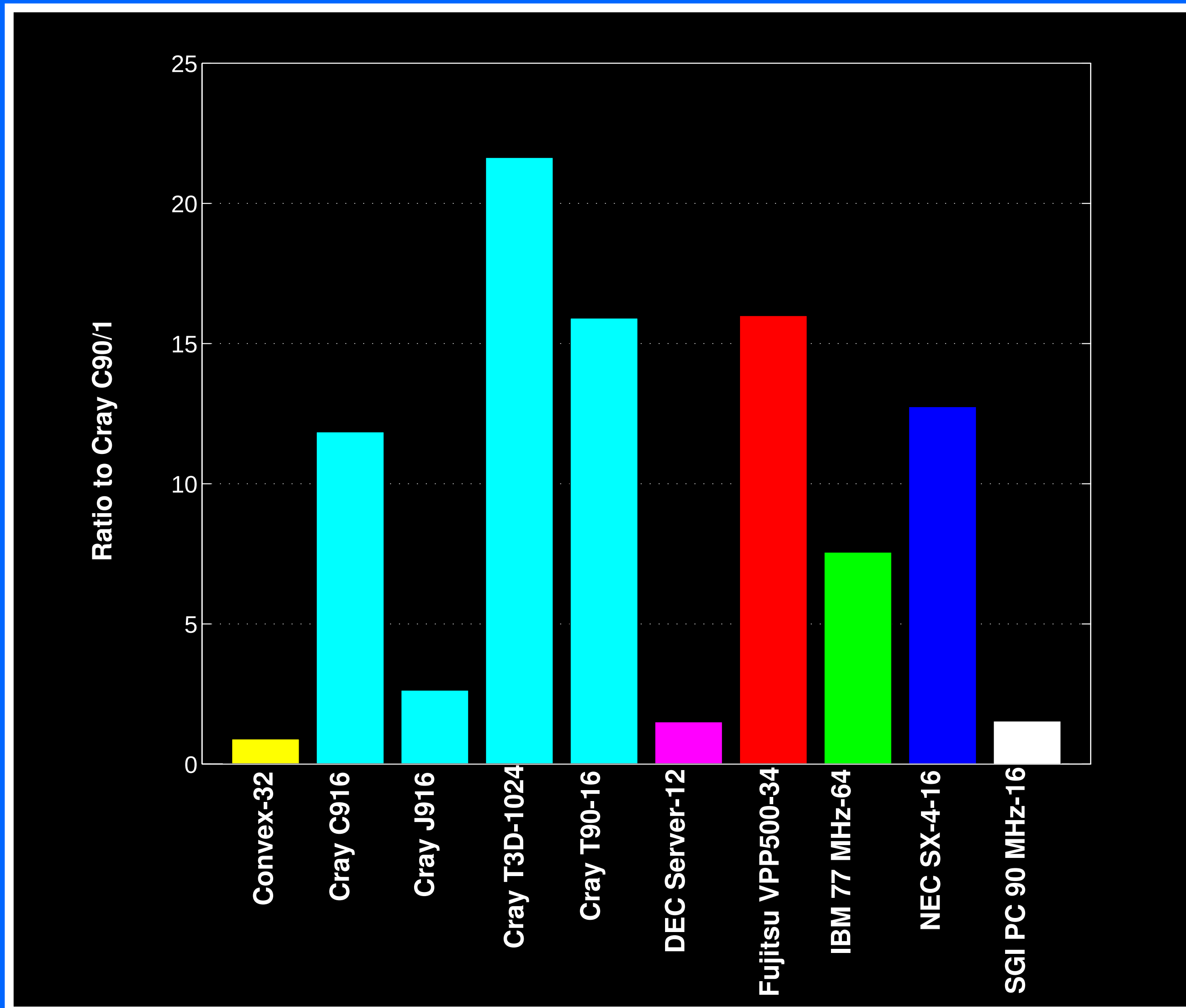
# Performance of Class B FT Benchmark



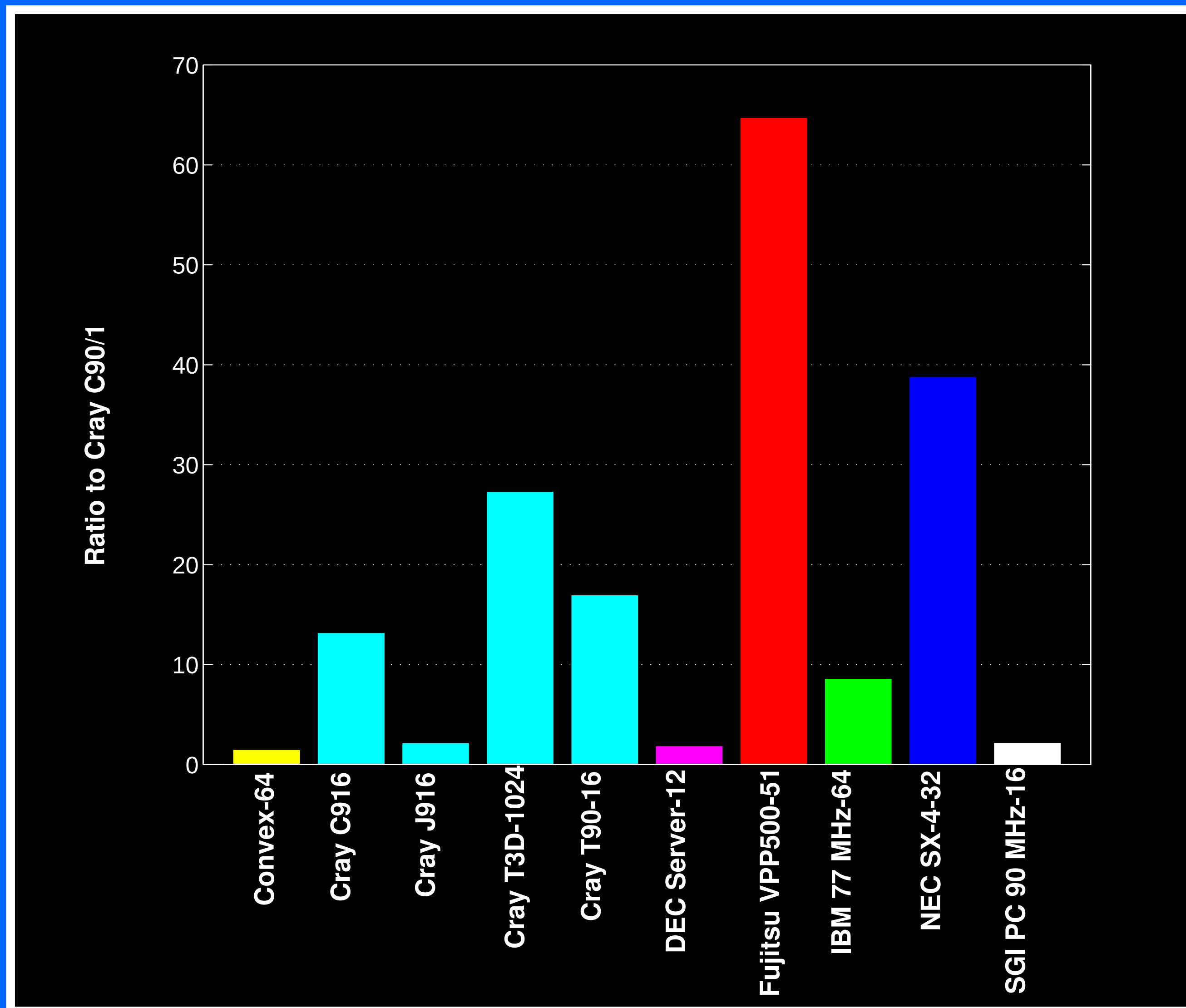
# Performance of Class B IS Benchmark



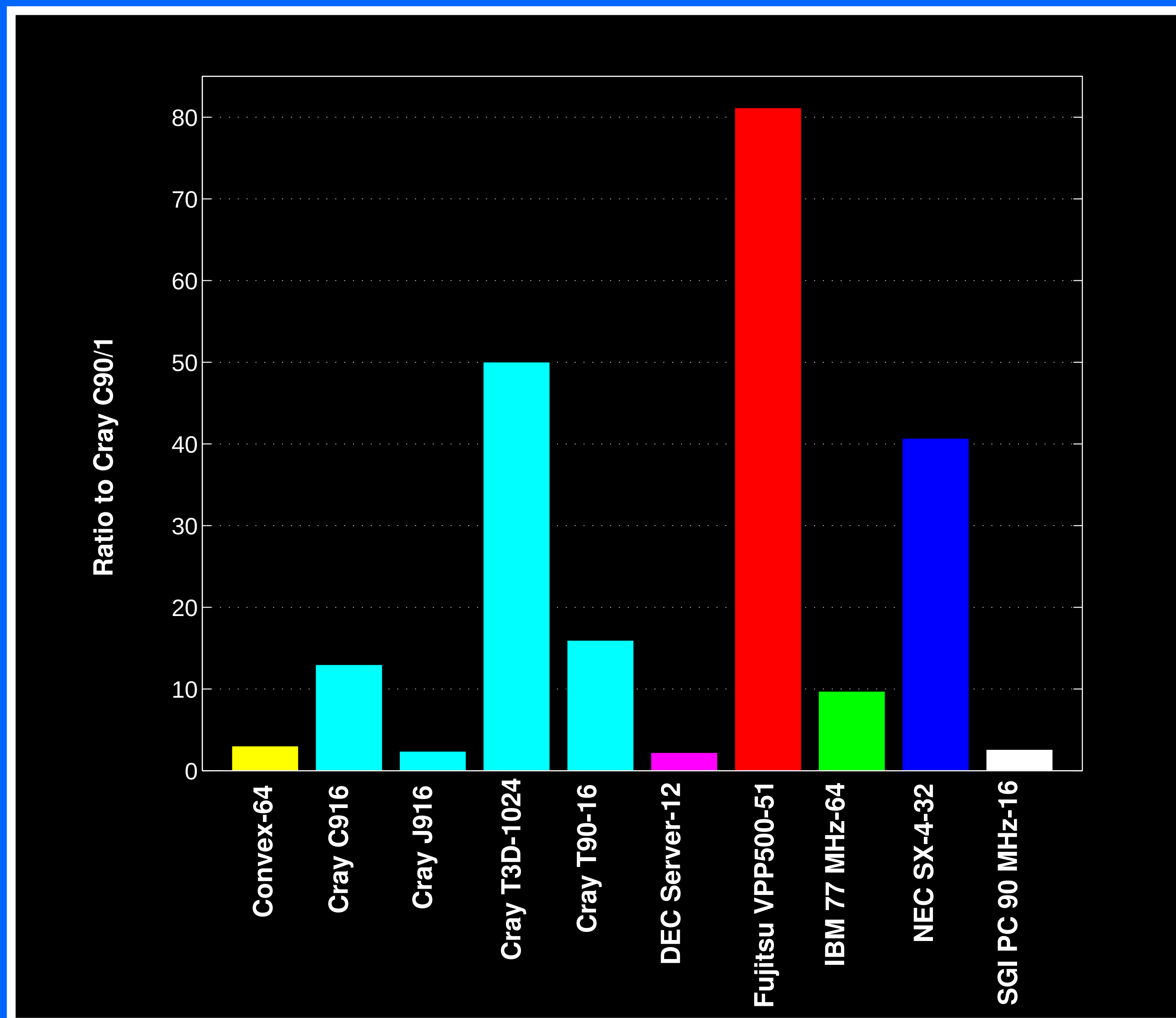
# Performance of Class B LU Benchmark



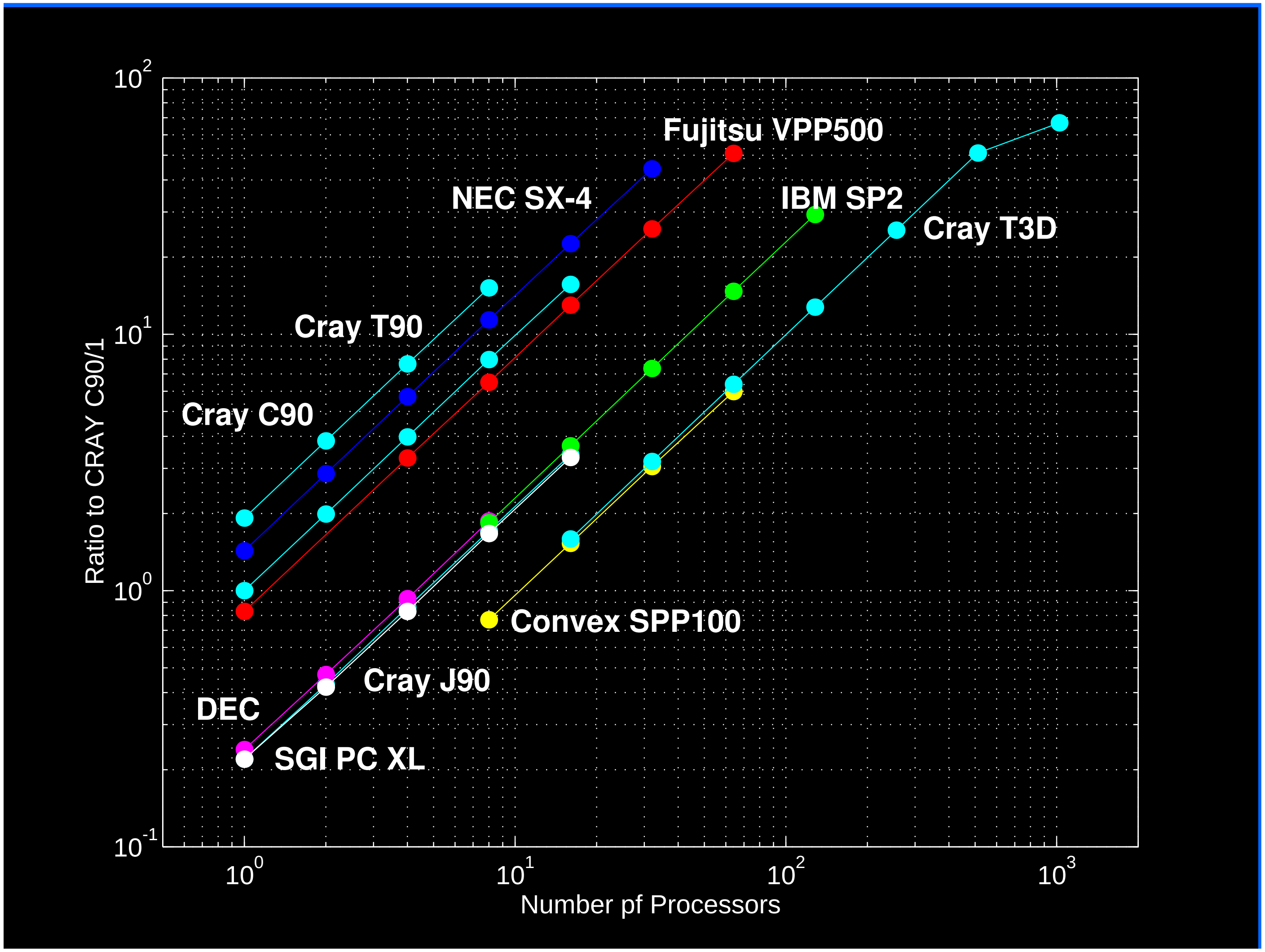
# Performance of Class B SP Benchmark



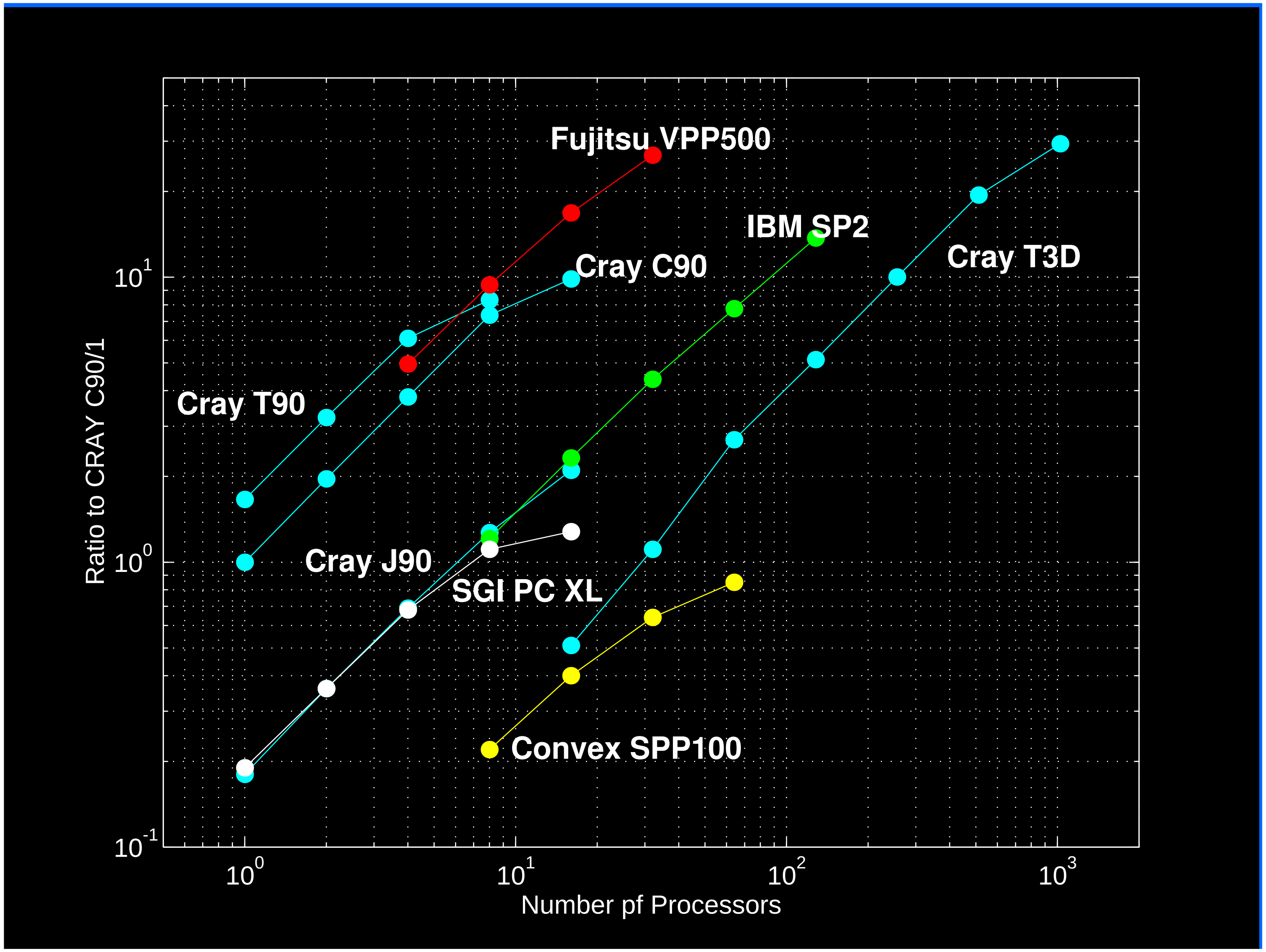
# Performance of Class B BT Benchmark



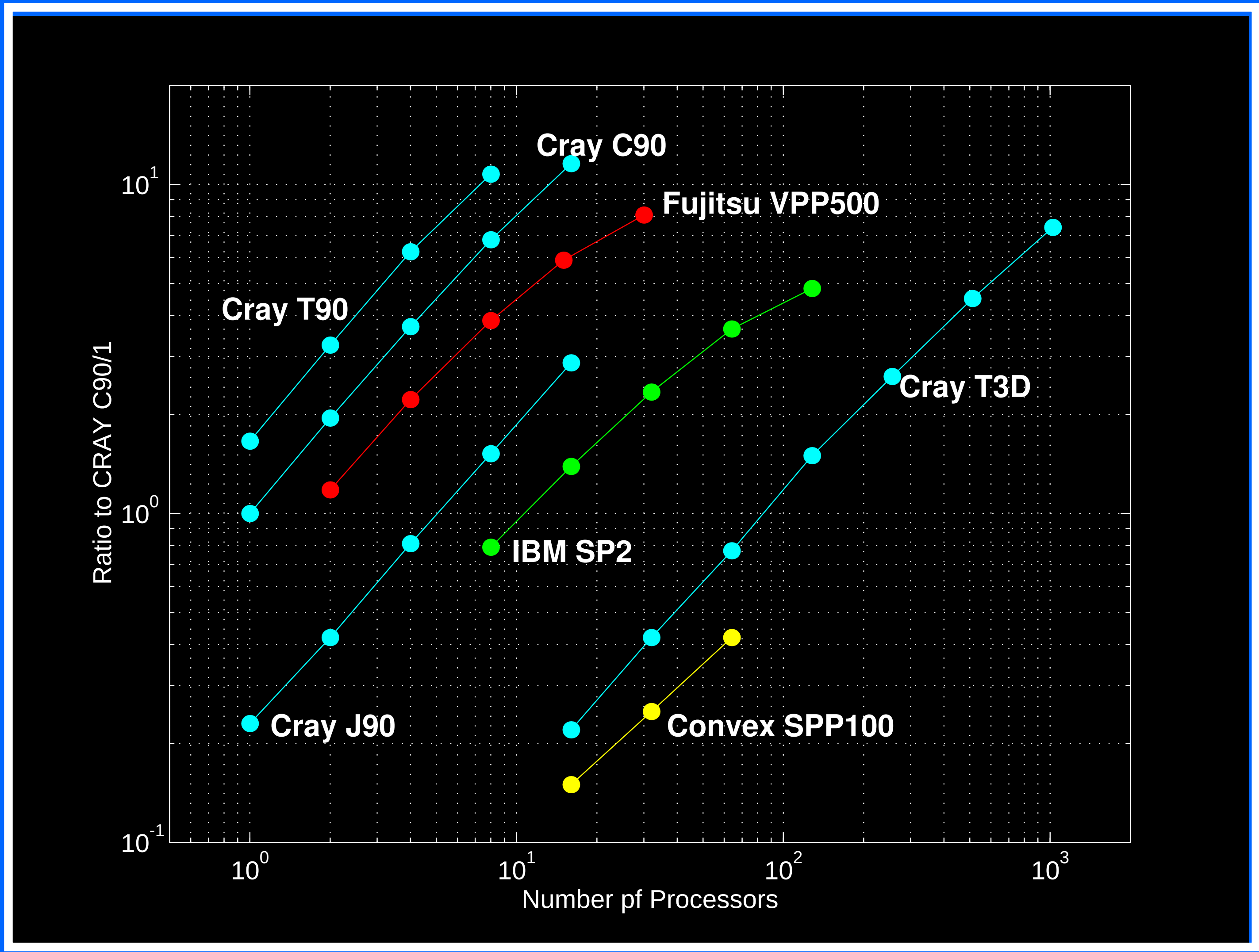
# Scalability of Class B EP Benchmark



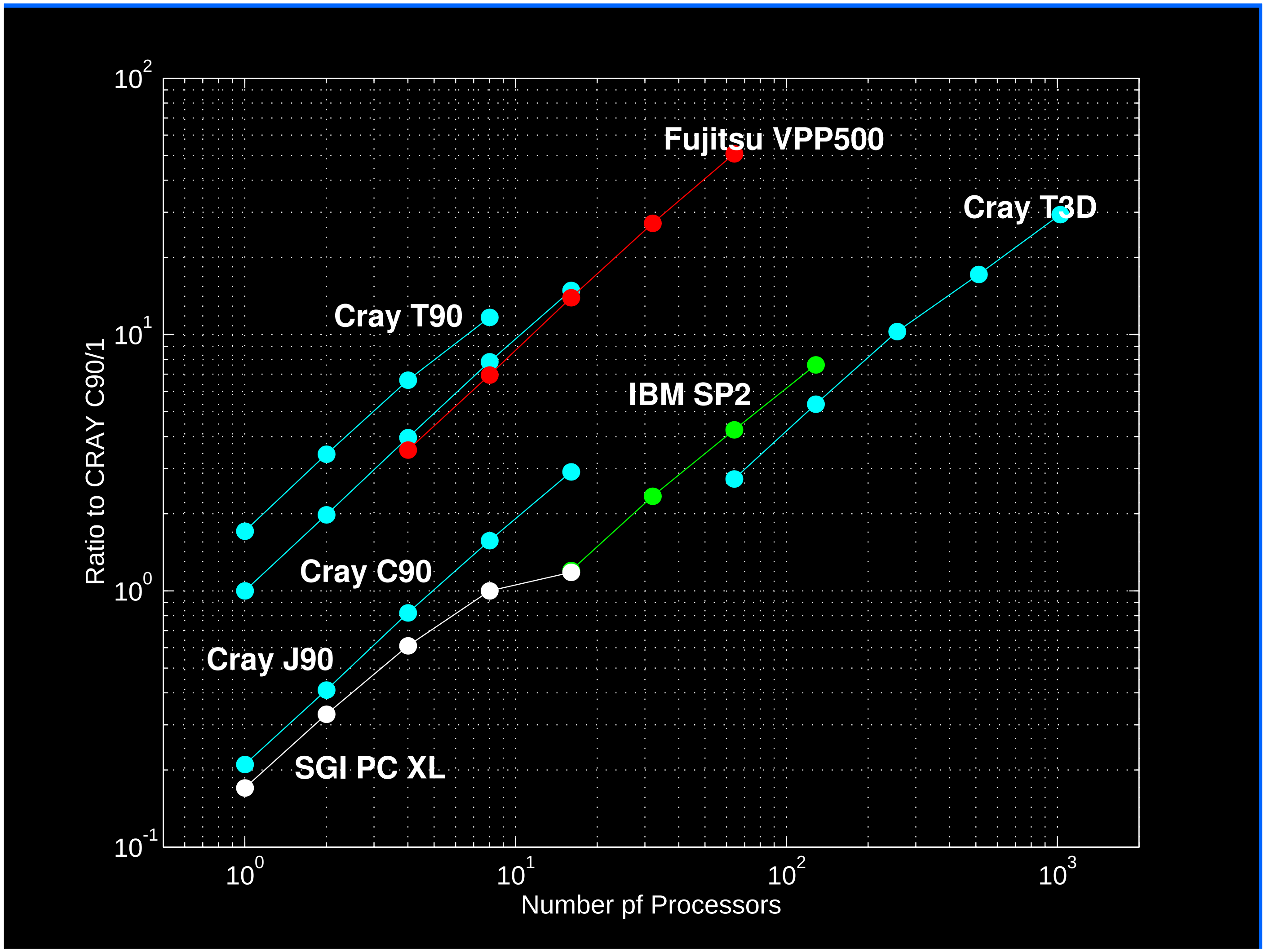
# Scalability of Class B MG Benchmark



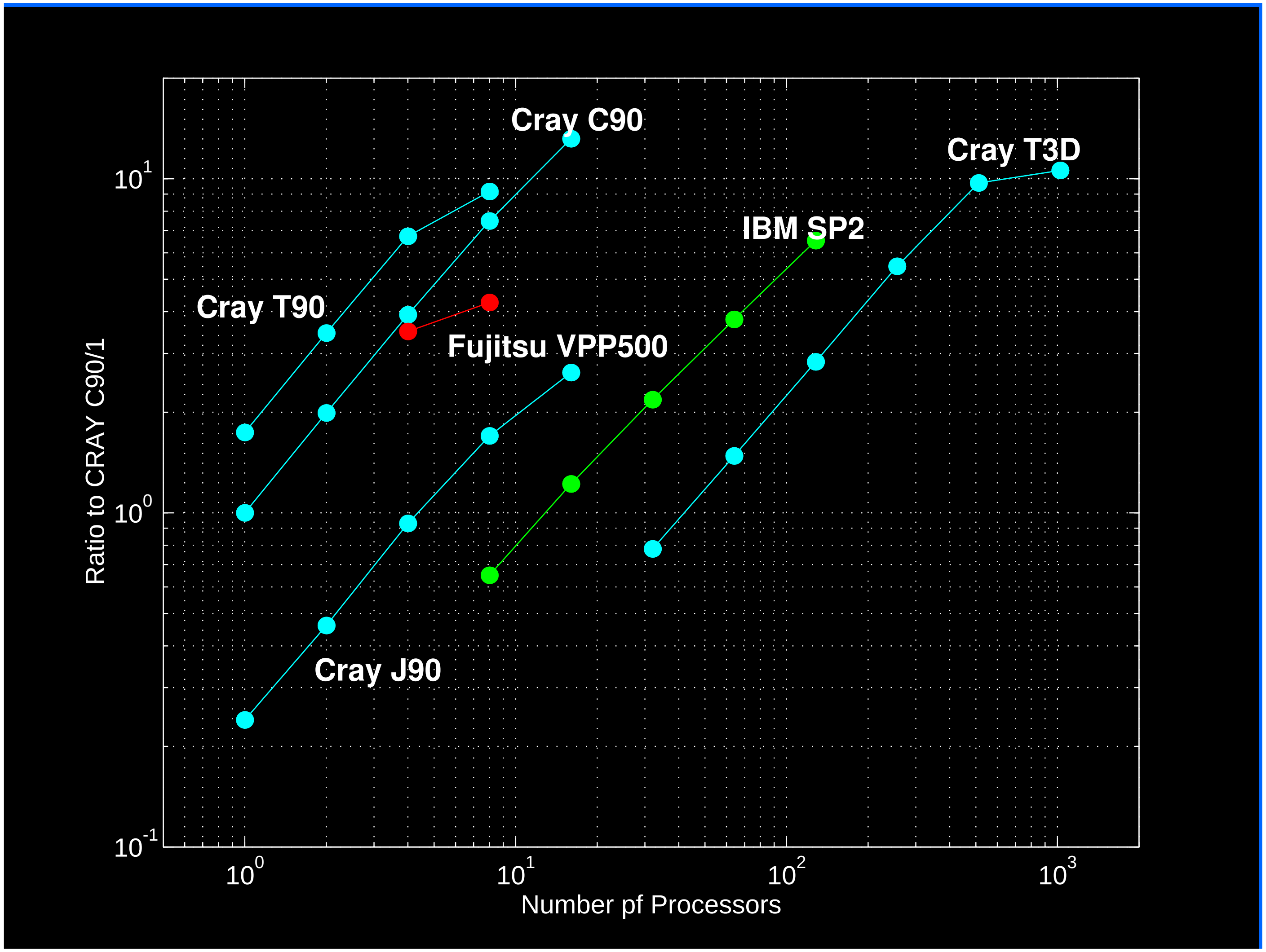
# Scalability of Class B CG Benchmark



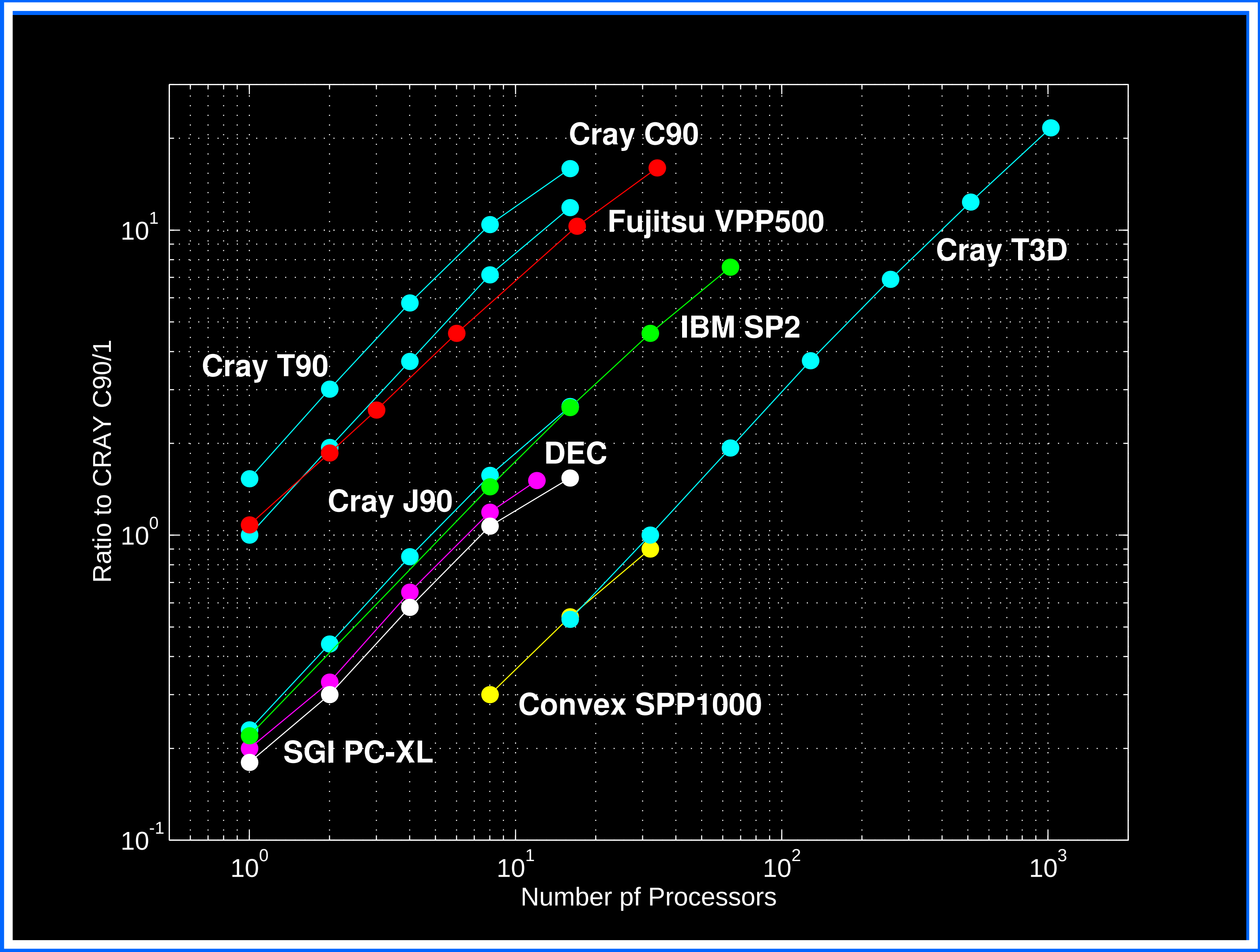
# Scalability of Class B FT Benchmark



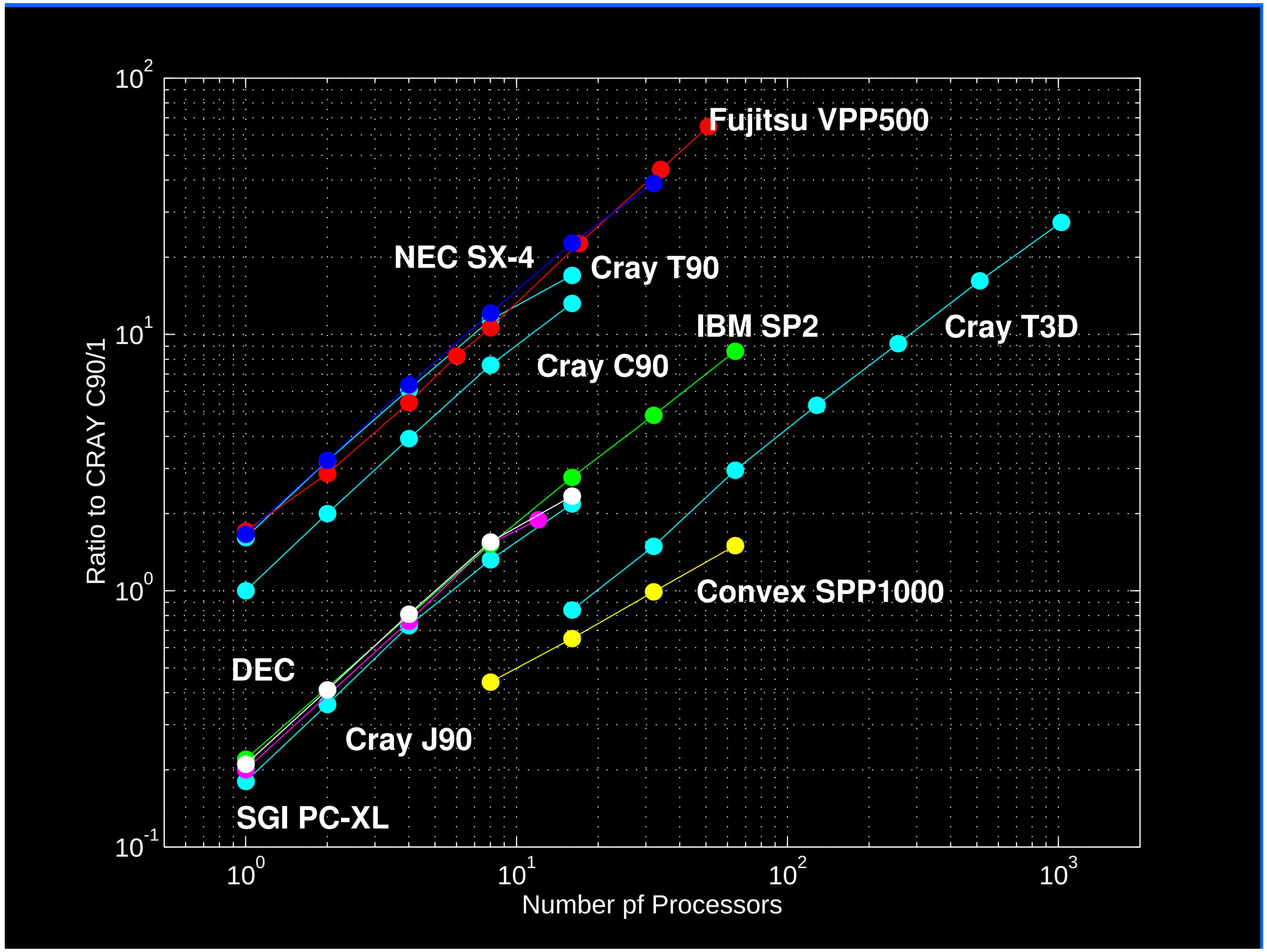
# Scalability of Class B IS Benchmark



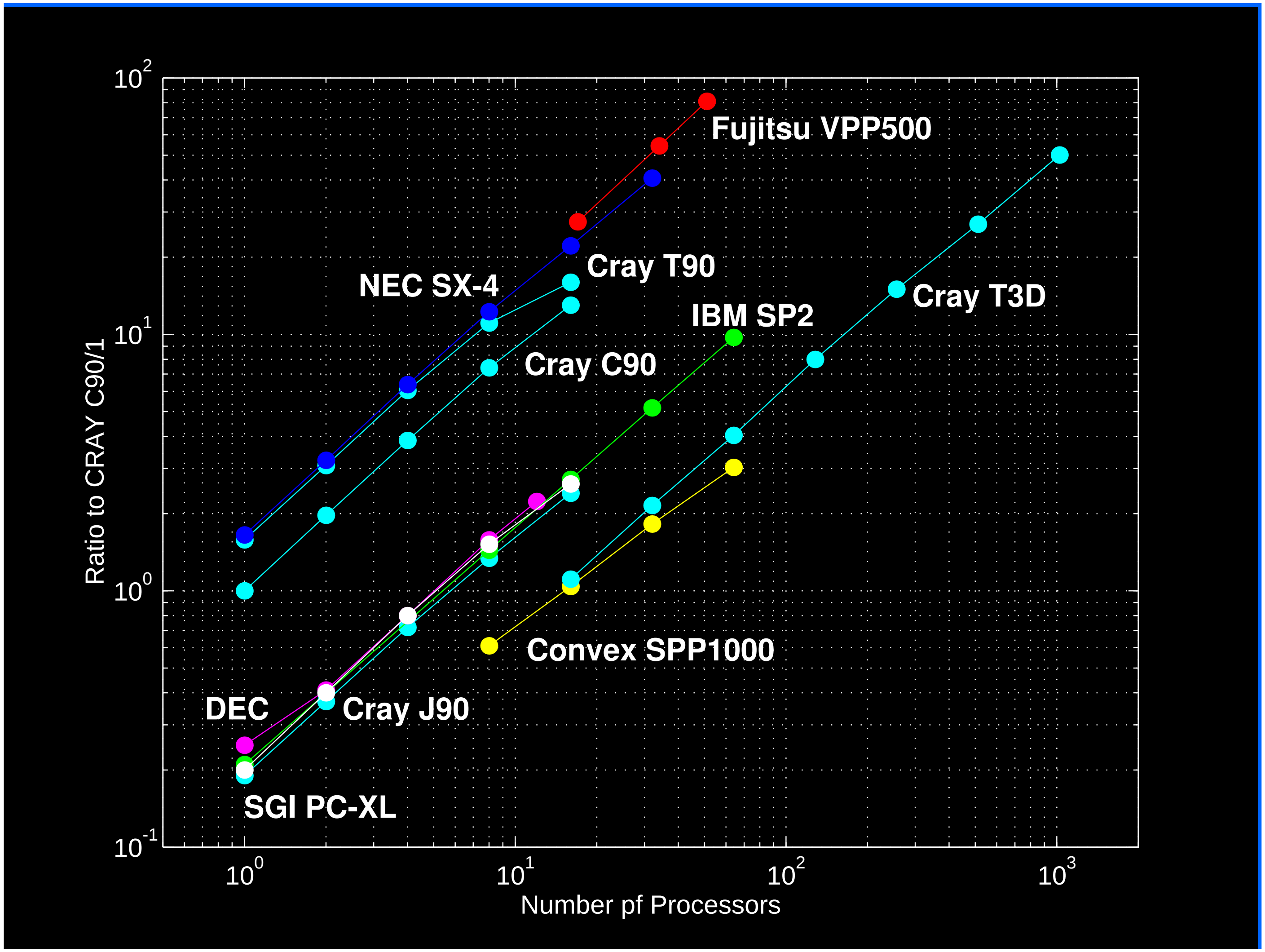
# Scalability of Class B LU Benchmark



# Scalability of Class B SP Benchmark



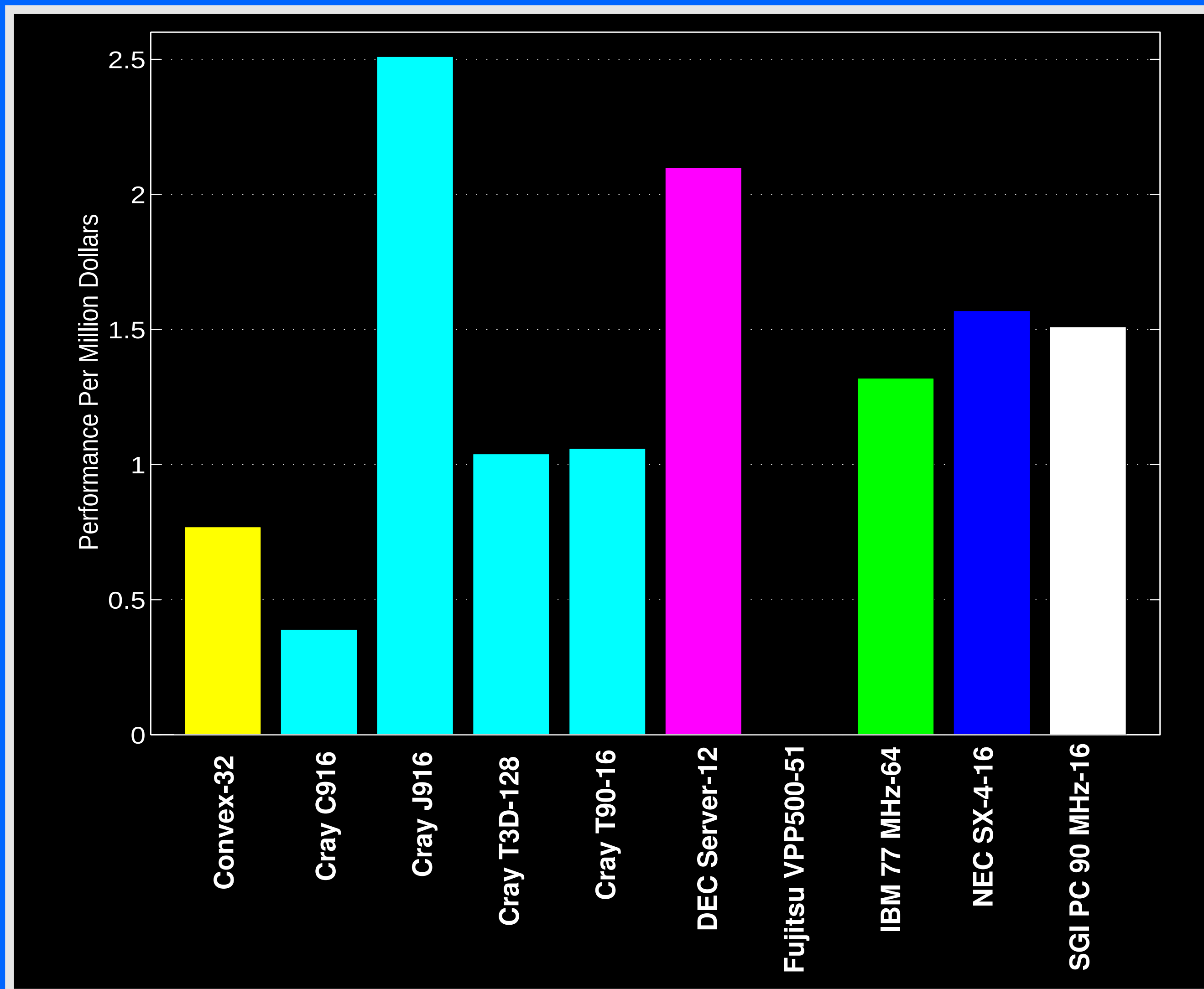
# Scalability of Class B BT Benchmark



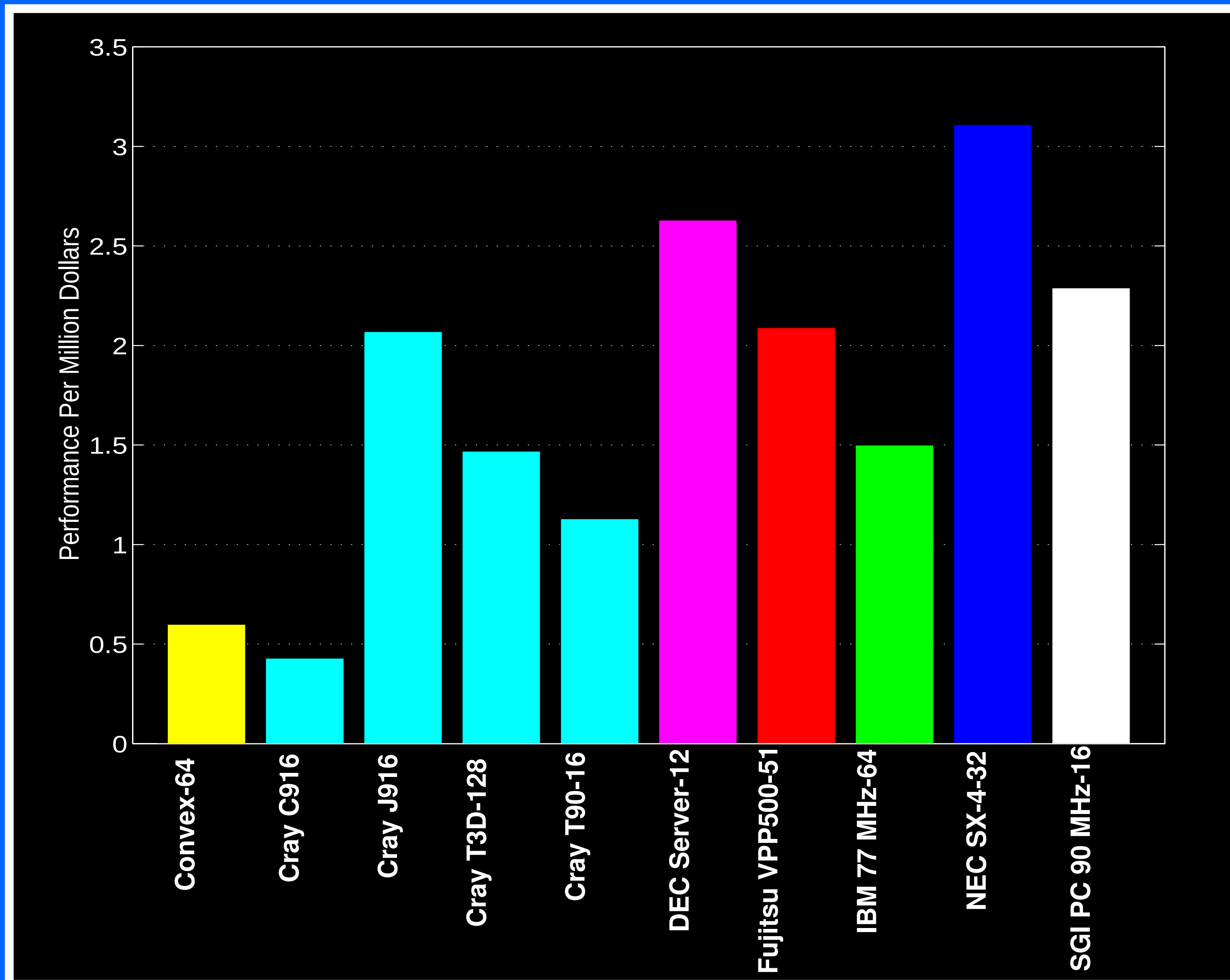
# Sustained Performance Per Dollar

- The relative performance of the systems must take into account the differences in price of these systems.
- Compute sustained performance per million dollars, i.e., performance (ratio to Cray C90/1) divided by the list price in millions.
- List price includes minimal system (memory per node, disk space, etc.) required to run full Class B size NPB.
- Finally, be aware that list prices are similar to peak performance in that they are guaranteed not to be exceeded.

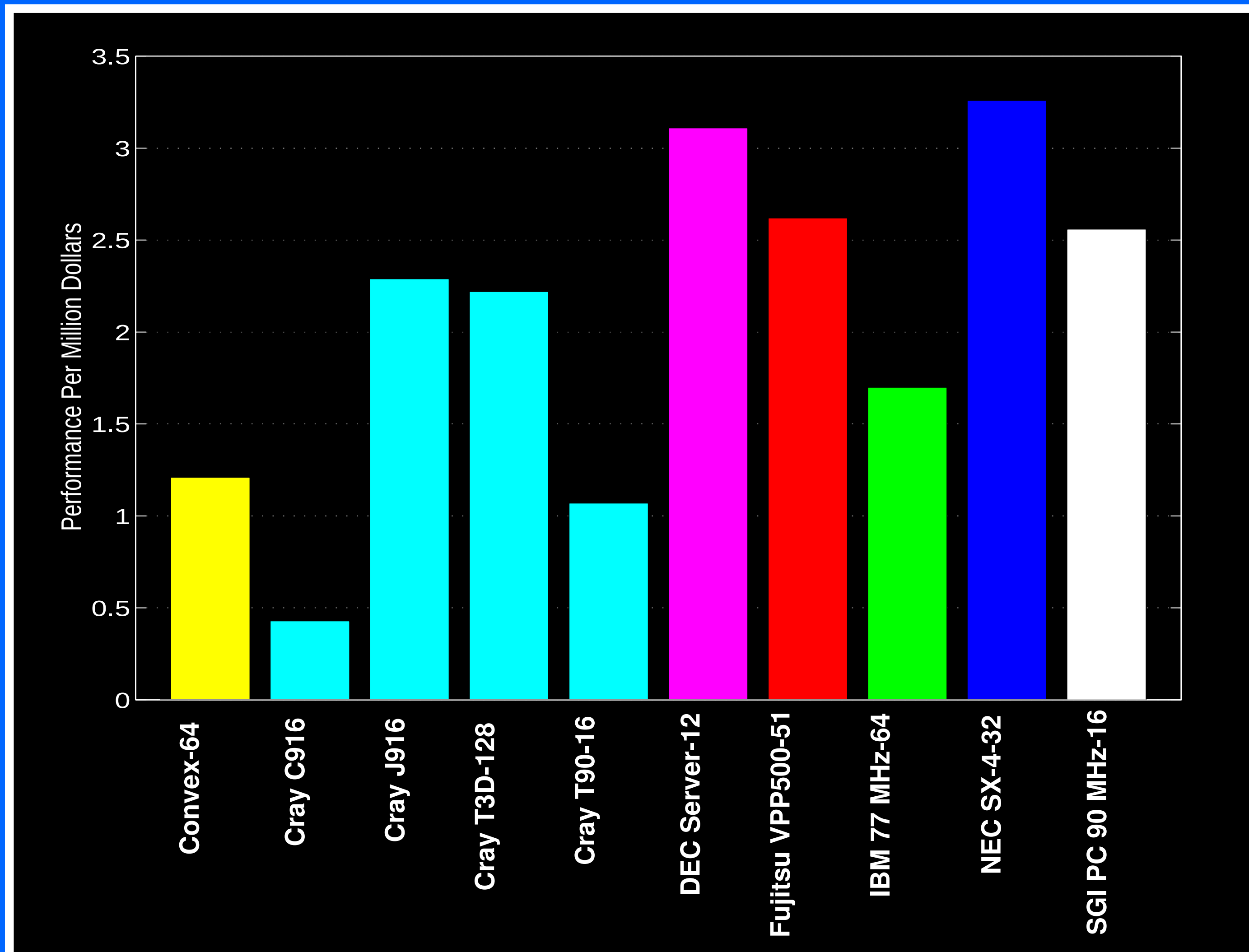
# Performance Per Million Dollars - LU



# Performance Per Million Dollars - SP



# Performance Per Million Dollars - BT



# Future Plans for NPB

- "Class C" problem sizes.
- Standard MPI implementations of five benchmarks (MG, FT, LU, SP and BT) to be released as NPB 2.0.
- Standard HPF implementations of some benchmarks.
- A new set of benchmarks, including

  1. an unstructured grid benchmark,
  2. an adaptive grid benchmark,
  3. a fluid-structures (multidisciplinary) benchmark.

# Questions for the Future

- How much longer will conventional vector mainframes be around?
- What fraction of the total scientific computing market will be captured by RISC workstations?
- What role will networks of workstations play?
- What will emerge as the preferred language for programming highly parallel systems?
- Will highly parallel interprocessor networks be able to keep pace with advances in RISC processors?

# Questions for the Future (Contd.)

- How far will workstation vendors be able to extend the concept of a shared memory, cache-coherent multiprocessor system?
- What will happen to computer technology as we approach the foreseeable limit of CMOS lithography (0.1 micron)?
- What new applications will be found for scientific computing?
- Will scientists lose control of scientific computing?

# Conclusions

- Supercomputers (both vector and highly parallel) need to focus on large, Grand Challenge jobs; otherwise workstations are more cost-effective.
- The CRAY T3D, CRAY J90, the DEC Alpha Server, the IBM SP2, NEC SX-4, and SGI Power Challenge all exceed the CRAY C90 in NPB performance per dollar.
- Performance reporting abuse, though improved from past years, has given the field an unsavory reputation.
- Highly parallel computers must deliver good performance using numerically efficient algorithms.

# Conclusions (Contd.)

- Weak points of current parallel systems:
  - Clumsy programming languages.
  - Fragile and inadequate system software.
- A more rigorous approach is required to clear the final hurdles to mainstream parallel computing.
- Many unanswered questions remain about the future of scientific computing.