DOE Evaluation of the Cray X1

Mark R. Fahey

James B. White III (Trey)

Center for Computational Sciences

Oak Ridge National Laboratory

Acknowledgement

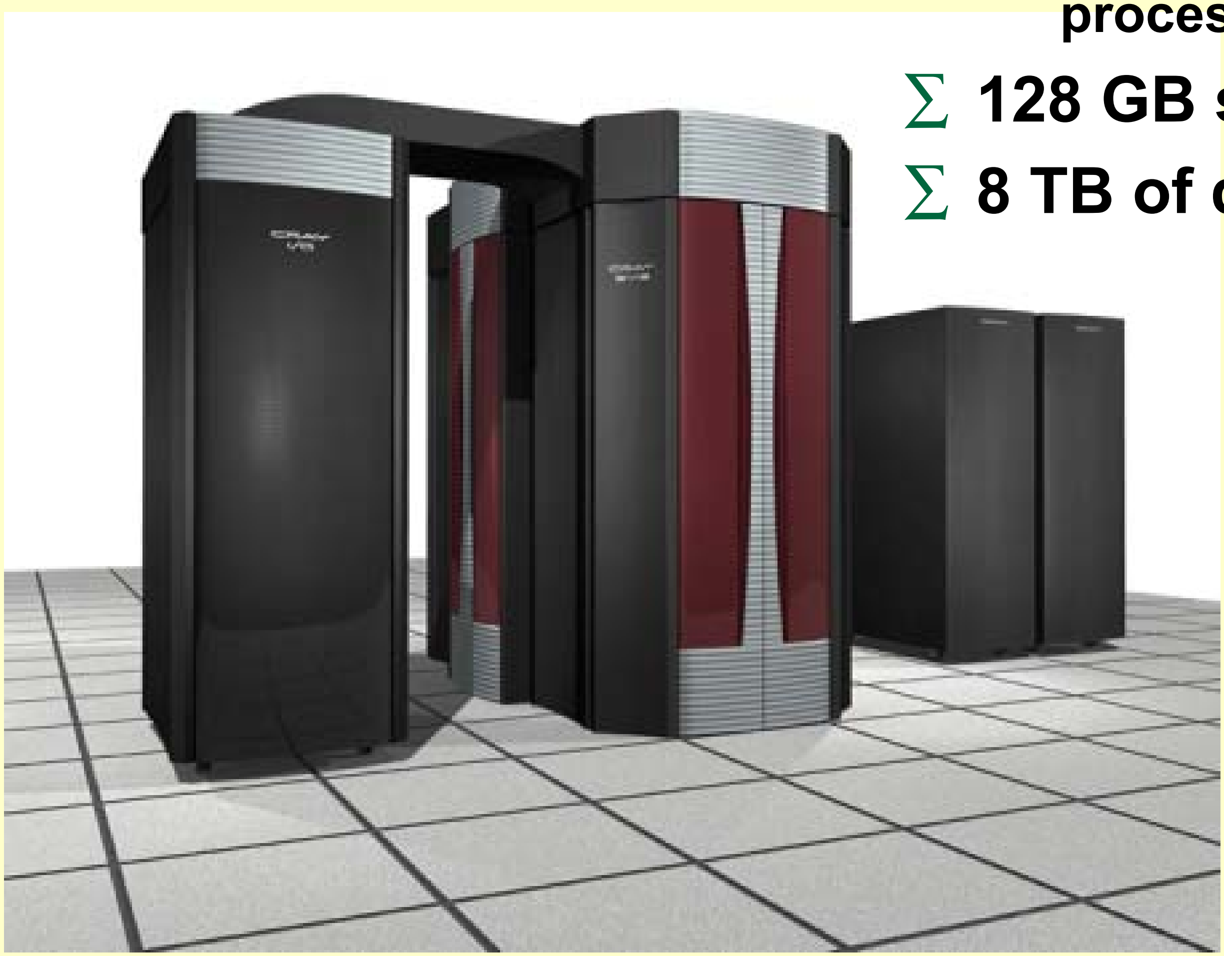
Research sponsored by the Mathematical, Information, and Computational Sciences Division, Office of Advanced Scientific Computing Research, U.S. Department of Energy, under Contract No. DE-AC05-00OR22725 with UT-Battelle, LLC.

Outline

- CCS X1
- Evaluation Overview
- Applications
    - Climate
    - Fusion
    - Materials
    - Biology

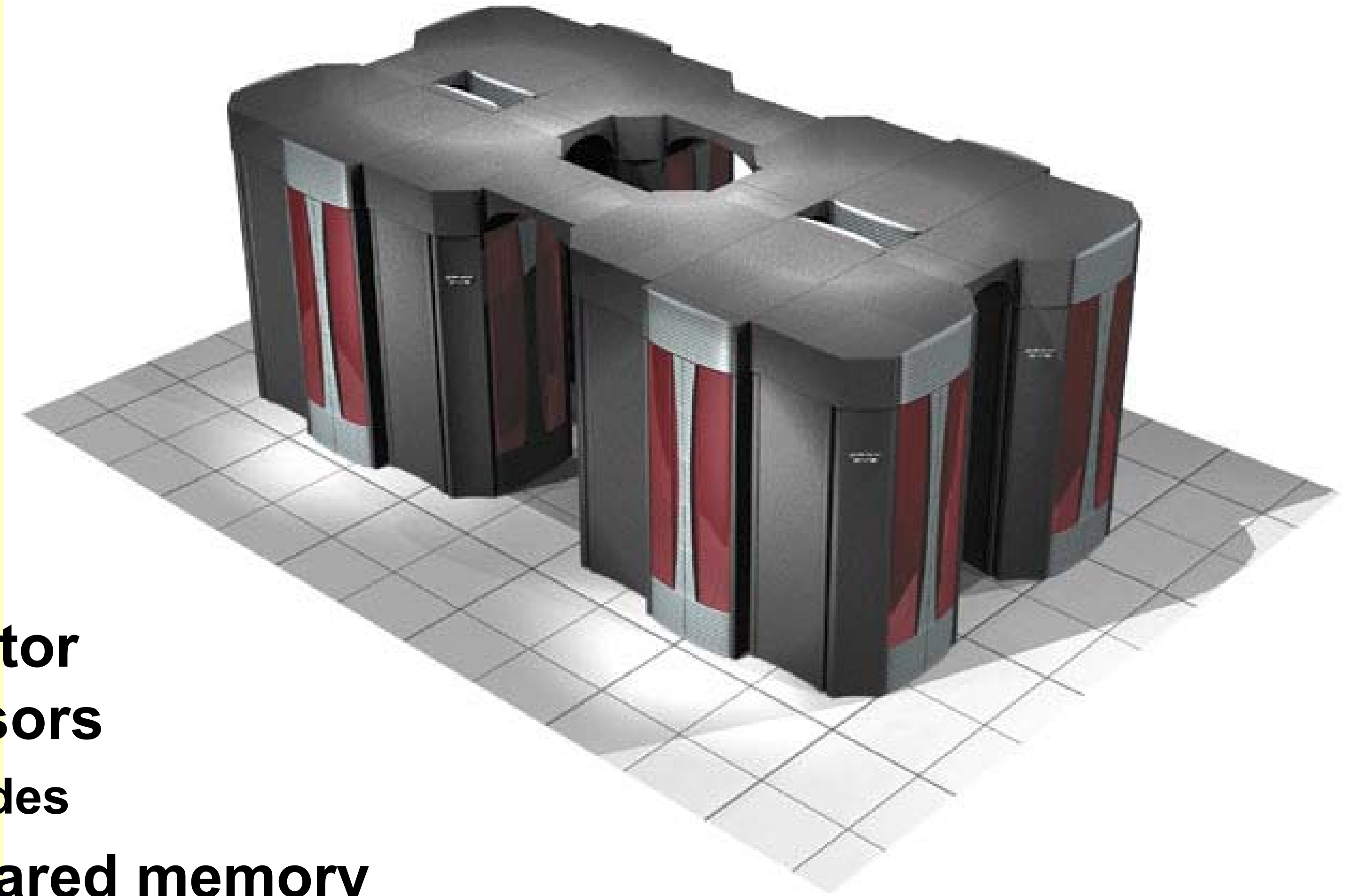
Phase 1 – March 2003

- 32 Vector Processors
  - 8 nodes, each with 4 processors
- 128 GB shared memory
- 8 TB of disk space



400 GigaFLOP/s

Phase 2 – September 2003



- 256 Vector Processors
  - 64 nodes
- 1 TB shared memory
- 20 TB of disk space

3.2 TeraFLOP/s

X1 evaluation

- Compare performance with other systems
    - Applications Performance Matrix
- Determine most-effective usage
- Evaluate system-software reliability and performance
- Predict scalability
- Collaborate with Cray on next generation

Hierarchical approach

- System software
- Microbenchmarks
- Parallel-paradigm evaluation
- Full applications
- Scalability evaluation

System-software evaluation

- Job-management systems
- Mean time between failure
- Mean time to repair
- All problems, with Cray responses
- Scalability and fault tolerance of OS
- Filesystem performance & scalability
- Tuning for HPSS, NFS, and wide-area high-bandwidth networks
- See Buddy Bland's Talk

  "Early Operations Experience with the Cray X1"

  Thursday at 11:00

Microbenchmarking

- **Results of standard benchmarks**
  - **http://www.csm.ornl.gov/~dunigan/cray**
  - **See Pat Worley's talk today at 4:45**
- **Performance metrics of components**
  - **Vector & scalar arithmetic**
  - **Memory hierarchy**
  - **Message passing**
  - **Process & thread management**
  - **I/O primitives**
- **Models of component performance**

Parallel-paradigm evaluation

- MPI-1, MPI-2 one-sided, SHMEM, Global Arrays, Co-Array Fortran, UPC, OpenMP, MLP, …
- Identify best techniques for X1
- Develop optimization strategies for applications

Scalability evaluation

- Hot-spot analysis
  - Inter- and intra-node communication
  - Memory contention
  - Parallel I/O
- Trend analysis for selected communication and I/O patterns
- Trend analysis for kernel benchmarks
- Scalability predictions from performance models and bounds

Full applications

- Full applications of interest to DOE Office of Science
  - Scientific goals require multi-tera-scale resources
- Evaluation of performance, scaling, and efficiency
- Evaluation of ease/effectiveness of targeted tuning

Identifying applications

- Draft evaluation plan
- Prototype Workshop at ORNL Nov. 5-6
- Feb 3-5, 2003: Fusion
- Feb 6, 2003: Climate
- March 2, 2003: Materials
- May 9, 2003: Biology
- Future DOE-wide workshops

Workshop Goals

- **Set priorities**
  - **Potential performance payoff**
  - **Potential science payoff**
- **Schedule the pipeline**
  - **porting/development**
  - **processor tuning**
  - **scalability tuning**
  - **production runs - *science!***
  - ***small* number of applications in each stage**

Identifying applications

- **Potential application**
  - **Important to DOE Office of Science**
  - **Scientific goals require multi-terascale resources**
- **Potential user**
  - **Knows the application**
  - **Willing and able to learn the X1**
  - **Motivated to tune application, not just recompile**

Climate

- 3 codes
  - CAM, CLM, POP
- Participants from
  - NCAR, LANL, LBNL, ORNL, NASA-Goddard, CRIEPI, Cray, NEC
- Want to optimize for NEC and Cray
  - May require different optimizations

Climate: CAM

- People involved
  - Cray(1), NEC(2), NCAR(1), ORNL(2)
- Porting, profiling ongoing at Cray
- NEC expects single node optimizations for SX-6 complete by early Fall
  - Coordination between NEC and Cray?
- Radiation and Cloud models are focus of most work

Climate: CLM

- Land component of the Community Climate System Model
- Undergoing changes to data structures to make easier to extend and maintain
  - Fortran user-defined types with pointers
- Vectorization involvement
  - NCAR(1), ORNL(2), Cray(1), NEC(1)
  - Coordination with NEC to be worked out
- See Trey White's presentation
  - Wednesday at 8:45

Climate: POP

- Organization involvement
  - LANL(1), Cray(1), NCAR(2), CRIEPI(2)
- Need to coordinate between CRIEPI and Cray
- Significant optimizations already implemented, successful
  - Vectorization and Co-Array Fortran
- Remaining issues
  - Parallel algorithm issues
  - I/O issues
- See Pat Worley's presentation
  - "Early Performance Evaluation of the Cray X1"
  - Today at 4:45

Fusion

- Workshop held Feb 3-5 @ ORNL
- Participants from
  - General Atomics, Princeton Plasma Physics Lab, University of Wisconsin, University of Iowa, Cray, ORNL
- 6 codes
  - M3D and NIMROD (extended MHD)
  - GYRO and GTC (micro turbulence)
  - AORSA and TORIC (RF plasma interactions)
- Concurrent work by different teams
- Too many codes?
  - Provides flexibility when impediments encountered

Fusion: NIMROD

- CCS Teaming with developer and Cray to port and optimize
  - Cray has actively participated
- Uses F90 reshape quite a bit
  - Exploits a known weakness in the compiler
  - Cray filed SPR
- Uses F90 sums extensively inside loops that should be vectorizable
  - Compiler cannot vectorize, arrays are actually pointers
  - Cray filed SPR
- Dramatic effect on performance
  - Cannot predict how fast will be when compiler fixed

Fusion: NIMROD (cont.)

- Data structures are derived types of pointers with allocatable attribute
- Pointers vs allocatable arrays
  - How much performance can be gained by replacing pointers?
    - Would benefit other architectures too
  - Analysis needed before code rewrite can even be discussed
  - Climate Land Model success is important here
    - See Trey White's presentation
    - Wednesday at 8:45

Fusion: GYRO

- Developer and CCS teaming
- Implemented in F90, no derived data-types
- Hand-coded transpose operations using loops over MPI_Alltoall
    - Expected to scale to ES class machine
    - Scaling is ultimately limited by this
- UMFPACK library for field solves

Fusion: GYRO (cont.)

∑ Functional port complete
    - Has identified a couple bugs in code
∑ Several routines easily vectorized by manual loop interchange, and directives
∑ Vectorized sin/cos calls by rewriting code
    - Numerical integration routine
    - Bisection search
∑ Hand optimizations have yielded 5X speedup so far (more work to do)
    - About 35% faster than PWR4 (not enough!)

Fusion: GTC

- Developer ran GTC on SX6
- Cray had previously looked at parts of GTC
- Result:
  - Developer is directly working with Cray
- GTC has been ported to the X1
  - Some optimizations introduced
  - Work ongoing

Fusion: AORSA

- Uses ScaLAPACK
- Cray has ScaLAPACK implementation
  - Not tuned
  - Cray pursuing ScaLAPACK optimizations
- Ported
  - Performance worse than expected
  - Culprit is matrix scaling routine
    - Fix implemented, tests underway
- With Cray Benchmarking group

Fusion: M3D

- M3D uses PETSc
  - Parallel data layout done within this framework
  - Uses the iterative solvers
  - Accounts for 90% of time
- Need to port PETSc to X1
  - Estimate of 6 man-months
  - Require significant changes

Materials

- Primary codes
  - Dynamic Cluster Algorithm, FLAPW, LSMS, Socorro
- Secondary codes
  - LAMMPS, GP, FEFF/TD-DFT, a M-C code
- Majority are C++ and use MPI and/or OpenMP
- Contacts for each code identified

Materials: Dynamic Cluster Alg.

- **MPI, OpenMP, PBLAS, BLAS**
  - significant amount of time spent in dger
  - significant amount of time spent in cgemm
  - On the IBM Power4, the blas2 calls dominate
- **A quick port was performed**
  - Optimizations targeted a couple routines adding a few directives
  - Took a couple days
  - Showed dramatic speedup over IBM Power4
  - For the small problem that was solved
    - time doing calculations became nearly negligible
    - formatted I/O became dominant

Materials: LSMS

- Locally Self-consistent Multiple Scattering
- Code spends most of its time
  - matrix-matrix mulitplications
  - Computing partial inverse
- Communication involves exchanges of smaller matrices with neighbors
- Expected to vectorize well
- Developers are moving to sparse-matrix formulations to scale to larger problems
- With Cray Benchmarking group

Materials: FLAPW

- Full Potential Linearized Augmented Plane Wave (FLAPW) method
  - All-electron method
  - Considered to be most precise electron structure method in solid state physics
- Validation code and as such is important to a large percentage of the materials community
- Cray has started porting it

Biology

∑ **Workshop May 9 (few days ago!)**
∑ **Bioinformatics plan to exploit special features of X1**
  - **Not one or two codes that are more important**
  - **Probably can use primitives in BioLib**
  - **Collaborate with Cray on adding more primitives to BioLib**

∑ **Molecular dynamics based biology codes expected to vectorize**
  - **AMBER is used a lot**
  - **Cray working on AMBER port already, nearly done**

Conclusions

- **Future:**
  - **Chemistry and Astrophysics workshops**
- **Early results are promising**
- **Evaluation continues, much to do**
- **Workshops very productive**
  - **Focused set of codes to port is important**
  - **Identify users/teams**

References

- System software evaluation
  - Buddy Bland's talk Thursday at 11:00
- Results of standard benchmarks
  - http://www.csm.ornl.gov/~dunigan/cray
  - Pat Worley's talk today at 4:45
- Optimization Experiment with CLM
  - Trey White's talk Wednesday at 8:45

Contacts

- Trey White
  - whitejbiii@ornl.gov
- Mark Fahey
  - faheymr@ornl.gov
- Pat Worley
  - worleyph@ornl.gov
- Buddy Bland
  - blandas@ornl.gov

- consult@ccs.ornl.gov