Porting and Performance of the Community Climate System Model (CCSM3) on the Cray X1

George R Carr Jr
National Center for Atmospheric Science
Climate and Global Dynamics Division
gcarr@ucar.edu

NCAR

Co-Authors

- George R Carr Jr, NCAR/CGD
- Matthew J Cordery, Cray
- Ilene L Carpenter, SGI (formerly Cray)
- John B. Drake, ORNL
- Michael W Ham, ORNL
- Forrest M Hoffman, ORNL
- Patrick H. Worley,ORNL
- … at least another 20 supporters
  - Lawrence Buja, NCAR/CGD

NCAR

Overview

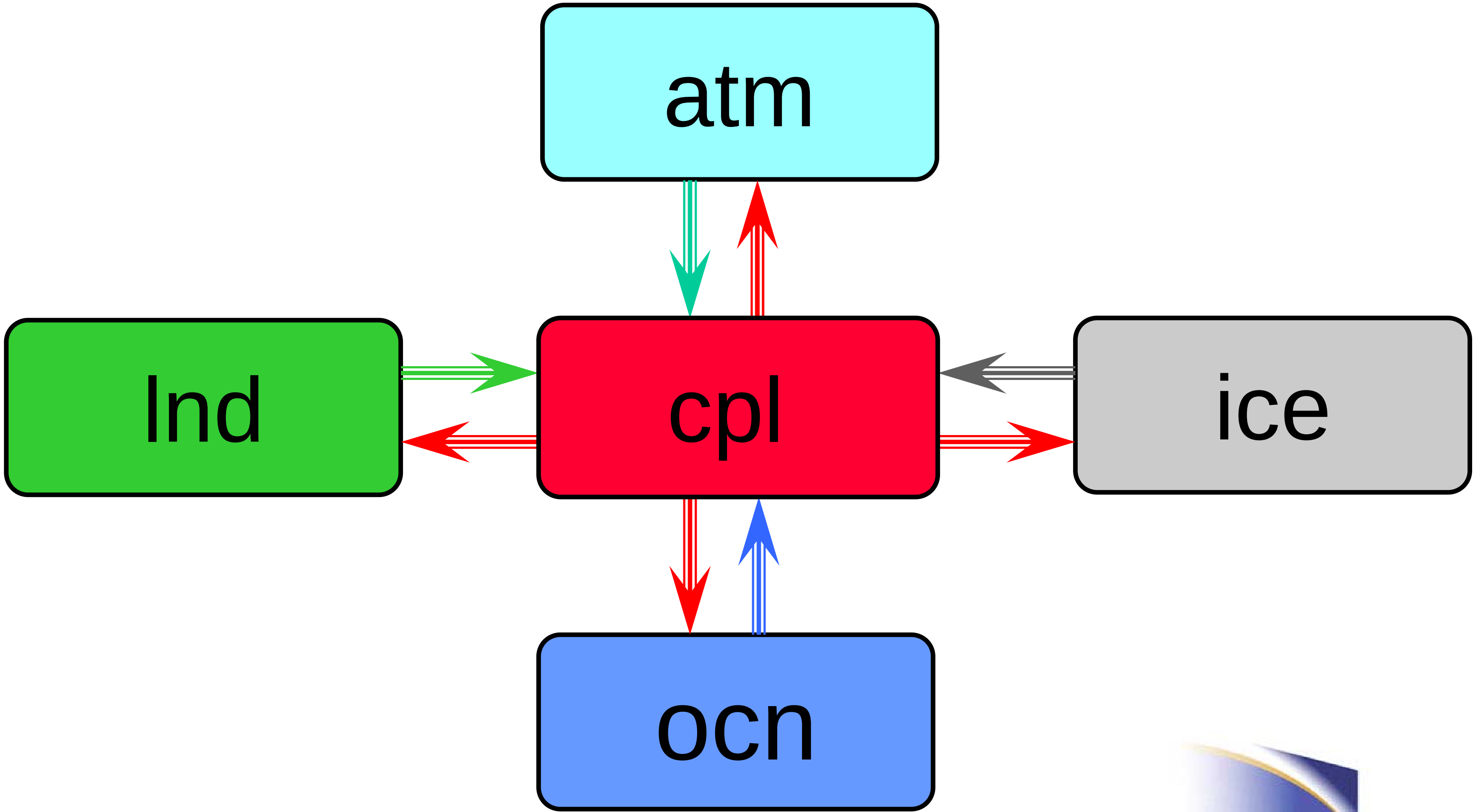
- CCSM3 Introduction
- Cray X1 Introduction and Status
- An Orientation on Performance
- Some Results
- Future Activities

CCSM Introduction

- **CCSM, the Community Climate System Model is a coupled model for simulating the earth's climate system.**
  - **Developed at NCAR with significant collaborations with DOE, NASA and the university community**
- **Components in CCSM3 include**
  - **Atmospheric Model – CAM 3.0**
  - **Ocean Model – modified version of POP 1.4.3**
  - **Sea Ice Model – CSIM5**
  - **Land Model – CLM2**
  - **Coupler - CPL6**

NCAR

CCSM Models



NCAR

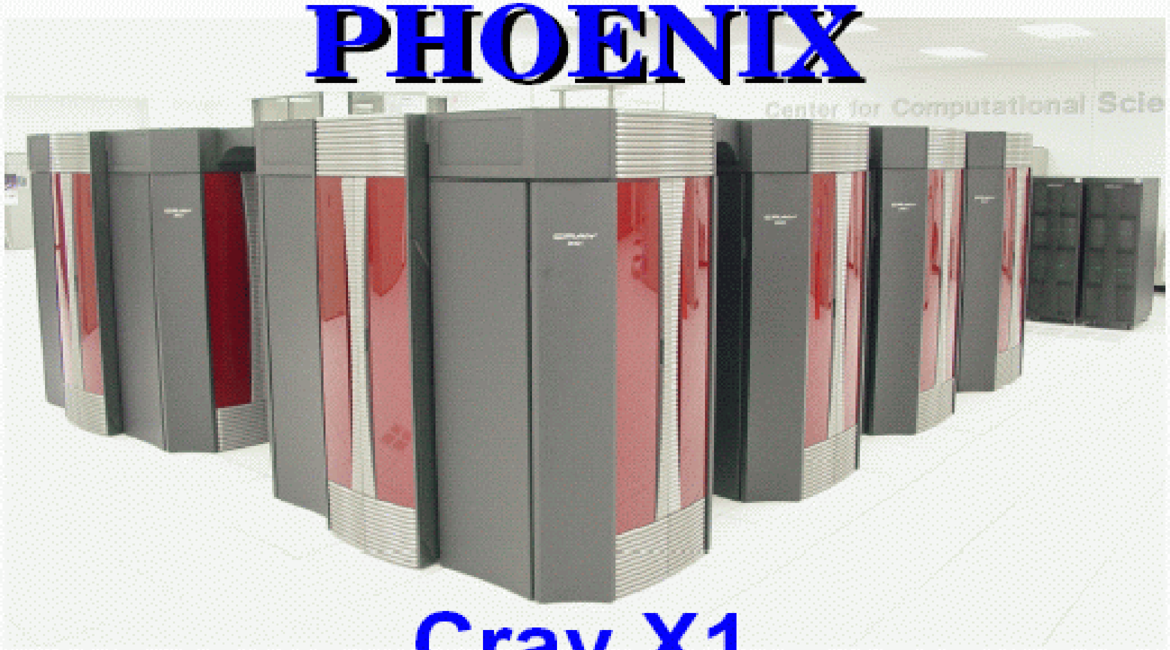
Supported Machines

- IBM P3, P4 - Cat 1
- SGI Origin - Cat 2
- Xeon Linux Clusters (GigE and Myrinet) - recently validated T31, will be Cat 1
- Cray X1 - recently validated T31, will be Cat 1
- SGI Altix - Ready for T31 Validation
- Earth Simulator - Validated on Pre-release
- Opteron Linux Clusters (Myrinet) - work begun
- Xeon Linux Clusters (InfiniBand) - work begun

See CCSM support URL for changes

NCAR

PHOENIX



Cray X1

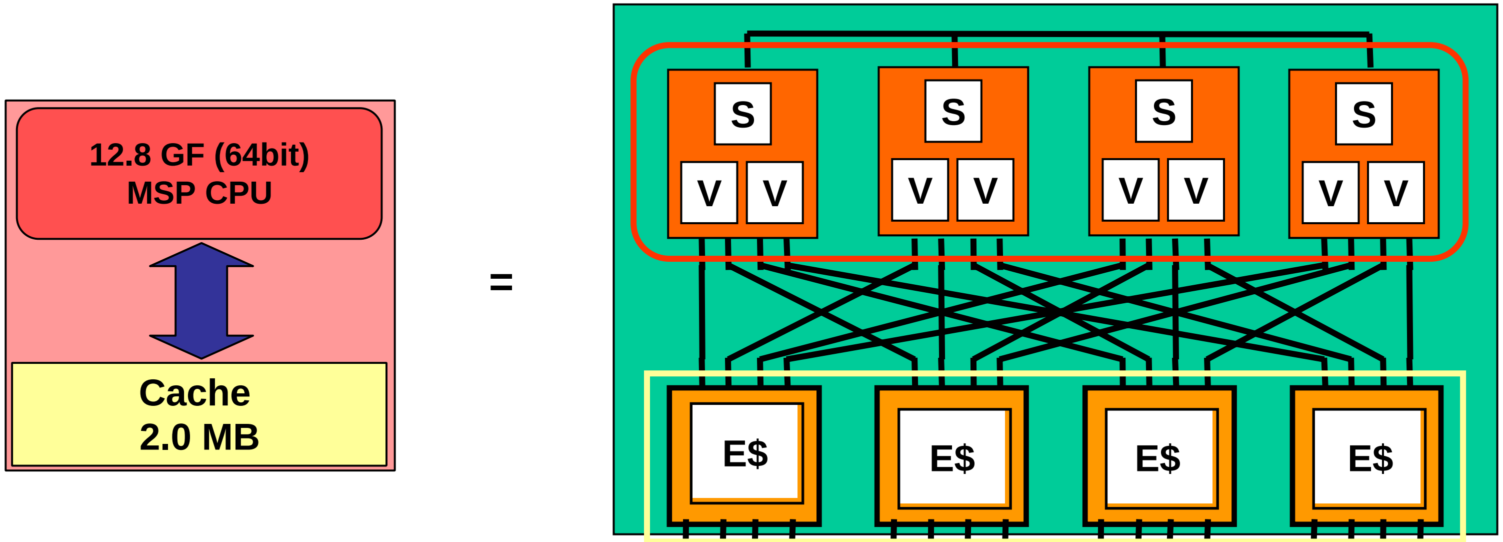
- 128 SMP nodes
- 4 Multi-Streaming Processors (MSPs) per node
- 4 Single Streaming Processors (SSPs) per MSP
- Two 32-stage, 64-bit wide vector units running at 800 MHz and one 2-way superscalar unit running at 400 MHz per SSP
- 2 MB E-cache per MSP
- 16 GB of memory per node

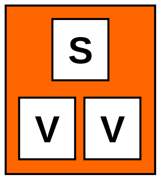
---

512 processors (MSPs), 2048 GB of memory, and 6400 GFlop/s peak

NCAR

Multistreaming Processor



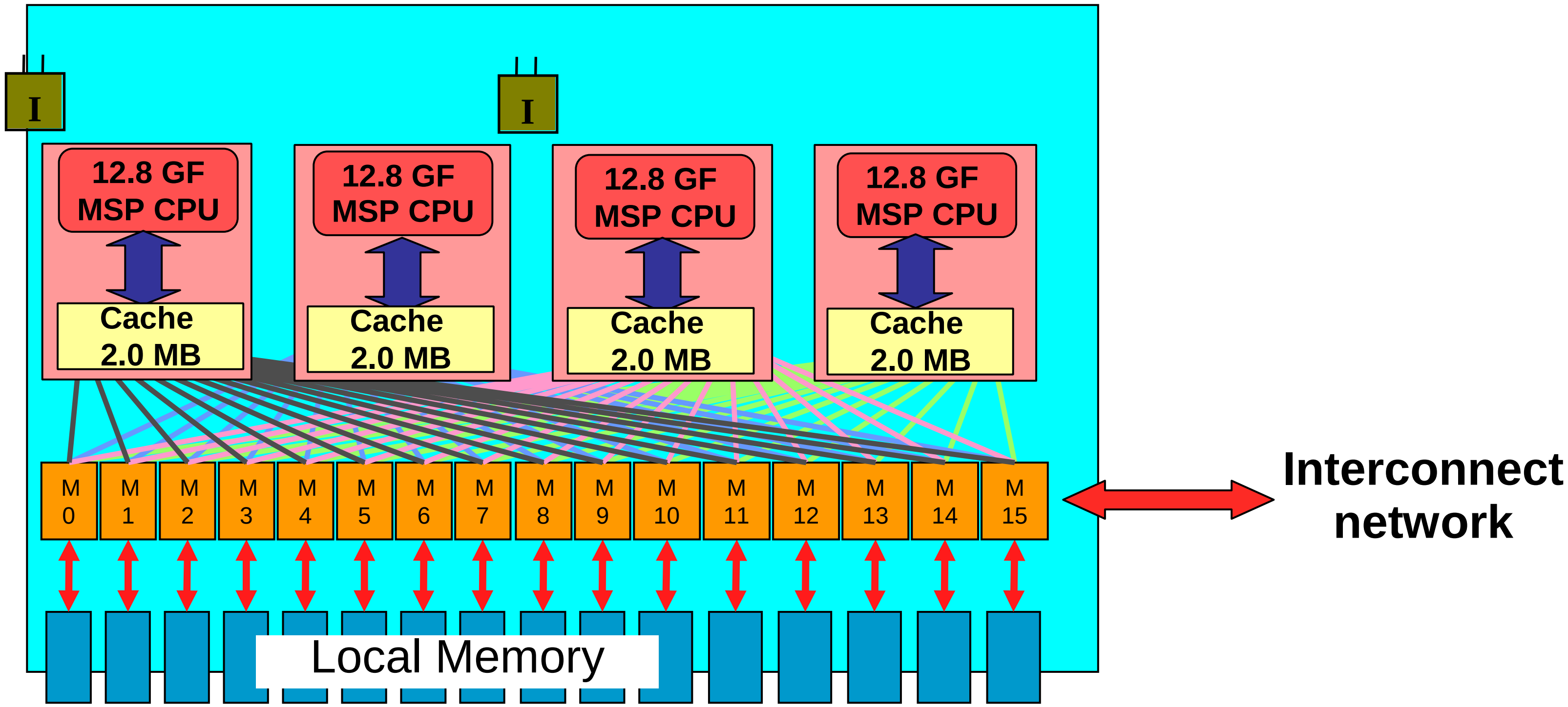


SSP – Single-Streaming Processor

- **Two vector pipe units**
- **One 4-way superscalar processor**

NCAR

Cray X1 Processor Node Module



Approximate Timelines

=> December 2003:
- Component model vectorization

=> April, 2004:
- Merge of vector versions into development branch, including basic support for the X1
- CAM/CLM2 standalone model (spectral Eulerian dycore) validated on the Earth Simulator and X1

=> June, 2004:
- CCSM validated on Earth Simulator and achieves required percentage of vectorization
- CCSM3 released, including basic support for X1

NCAR

Current X1 Project Highlights

- T31x3 Climate Validation Completed
- Choice of MSP (MPI only) orientation at this time due to OMP restrictions with MPMD.
- Some run configuration load balancing
- Regression test process begun
- Some *VERY* early performance numbers produced
- Functionality NOW, performance soon, portability required

NCAR

The Good

- Worked through initial problems with
  - Compiler
  - Kernel panic
  - Configuration issues (netcdf)
  - Scripts setup
  - Great support from ORNL and Cray
- All CCSM3 tests pass: T31, T45, and T85
- 75 year T31x3 climate validated

NCAR

Remaining X1 Issues

- **Model requires a particular (old) version of system software (compilers and MPI libraries).**
- **Model time in POP and CSIM4 suddenly becomes corrupted after approximately 10 simulation years.**
- **Performance variability is being explored.**
- **Answers change slightly (round-off level) when using dynamic CAM load balancing.**
- **Some performance timers in coupler are broken.**
- **Need to harden run scripts for ORNL environment.**
- **Long term archiving script is not yet set up for ORNL.**
- **Ice model validation may need to be revisited.**
- **Production script enhancements needed to speed up build process.**

NCAR

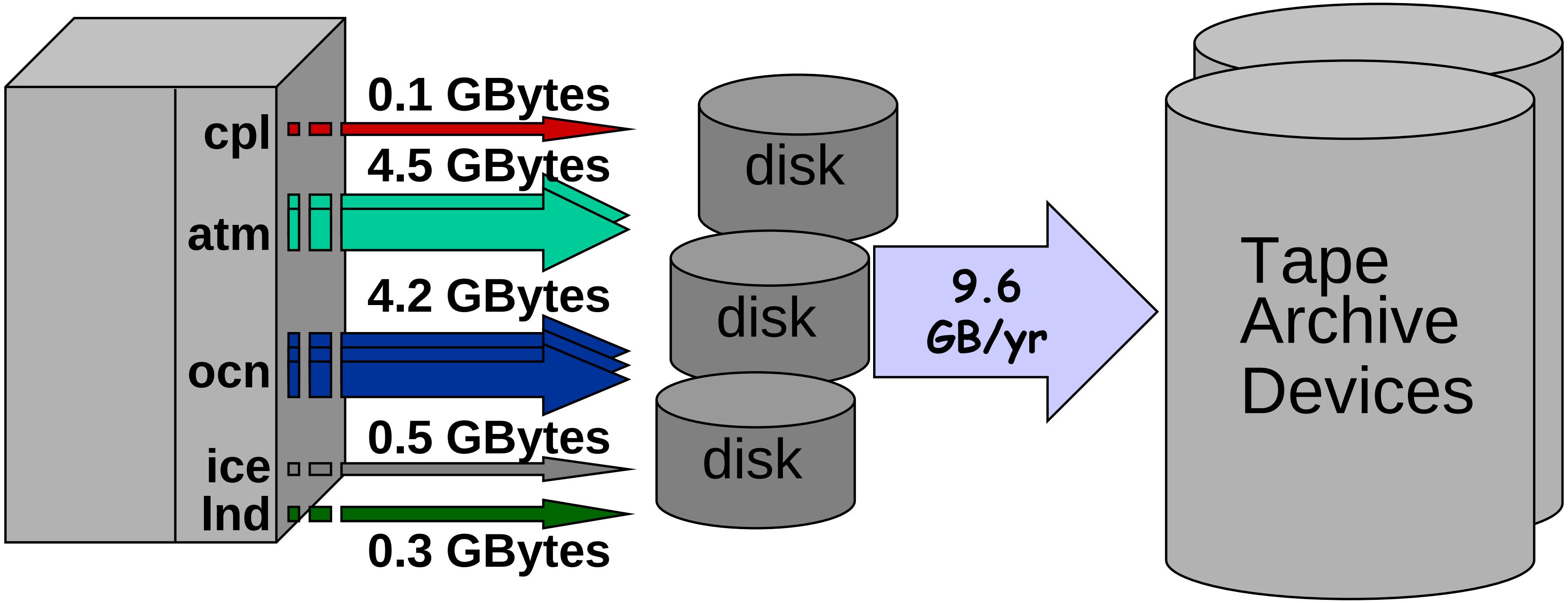
The Production Process

- Compile/Load
- Data Pre-stage - startup data files, restart files
- Job startup - system load, MPI startup, data ingest, data distribution
- Job (the real work) - daily/monthly log entries, monthly results
- Job termination - create restart files
- (optional) Short term archive (usually non-scrubbed disk)
- (optional) Long term archive (tape)
- Monitor progress (manual)
- Submit next job (can be automated in run script)

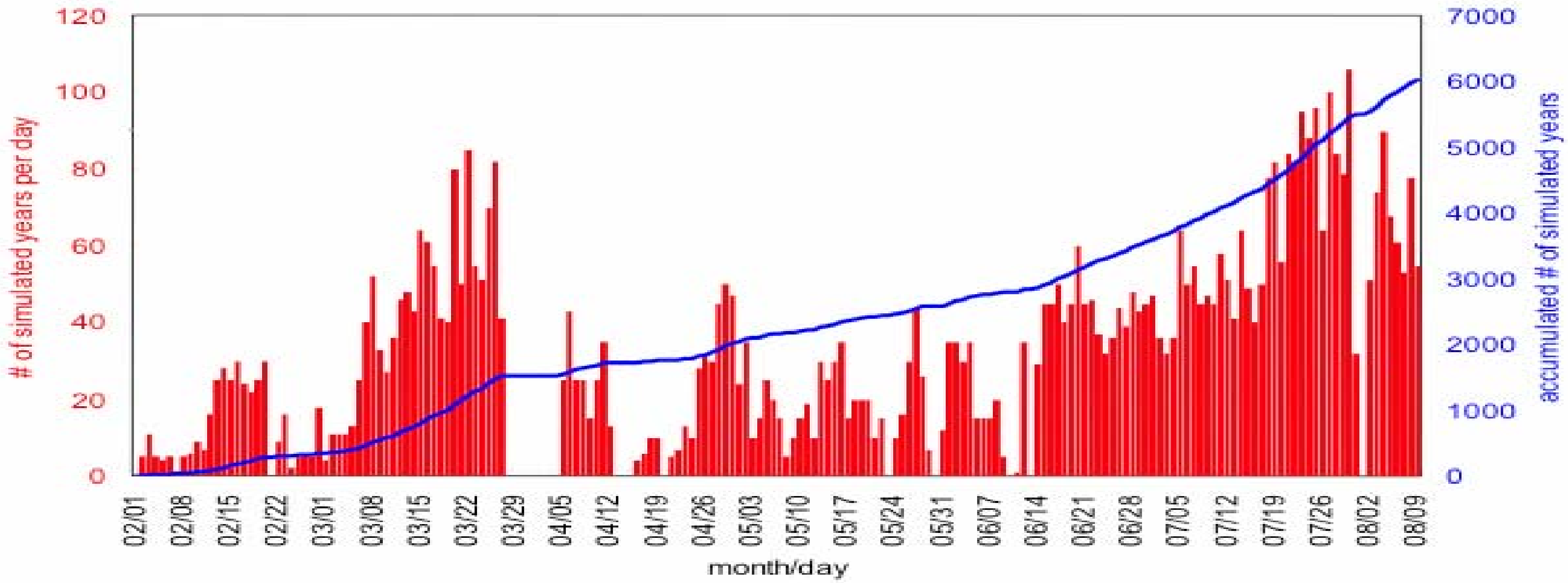
NCAR

CCSM T85 Data Output

T85 IPCC: 9.6 GBytes/year



IPCC ES Production Summary



Roughly 60 Tbytes of history data produced!
At times, could generate data faster than could get it to tape!

Special thanks to **Dr. Yoshikatsu Yoshida and all his colleagues of the Central Research Institute of Electric Power Industry (CRIEPI)**

NCAR

X1 Validation Observation

- Run time variability and average run time
  - T31x3 validation
    - Showed that a perfectly controlled system could run 7-8 seconds per day (on 36 MSPs)
    - One example: mean 12 seconds, range 7 to 46 seconds, mode of 10. Eight hour run on 36 “CPUs”
  - Seen with IBM. Better than Linux clusters tested. Seen on Origin and Altix also.
  - Possible sources
    - System process/processor migration
    - Job impacts by I/O sub-system
    - Timer issues do not seem to be an issue

NCAR

T31x3 Production Job Log

- (tStamp_write) cpl model date 0509-12-05 00000s wall clock 2004-10-06 17:45:08 avg dt 8s dt 8s
- (tStamp_write) cpl model date 0509-12-06 00000s wall clock 2004-10-06 17:45:16 avg dt 8s dt 8s
- (tStamp_write) cpl model date 0509-12-07 00000s wall clock 2004-10-06 17:45:28 avg dt 8s dt 12s
- (tStamp_write) cpl model date 0509-12-08 00000s wall clock 2004-10-06 17:45:36 avg dt 8s dt 8s
- (tStamp_write) cpl model date 0509-12-09 00000s wall clock 2004-10-06 17:45:44 avg dt 8s dt 8s
- (tStamp_write) cpl model date 0509-12-10 00000s wall clock 2004-10-06 17:45:52 avg dt 8s dt 8s
- (tStamp_write) cpl model date 0509-12-11 00000s wall clock 2004-10-06 17:45:59 avg dt 8s dt 8s
- (tStamp_write) cpl model date 0509-12-12 00000s wall clock 2004-10-06 17:46:07 avg dt 8s dt 8s
- (tStamp_write) cpl model date 0509-12-13 00000s wall clock 2004-10-06 17:46:15 avg dt 8s dt 8s
- (tStamp_write) cpl model date 0509-12-14 00000s wall clock 2004-10-06 17:46:23 avg dt 8s dt 8s
- (tStamp_write) cpl model date 0509-12-15 00000s wall clock 2004-10-06 17:46:31 avg dt 8s dt 8s
- (tStamp_write) cpl model date 0509-12-16 00000s wall clock 2004-10-06 17:47:13 avg dt 8s dt 42s
- (tStamp_write) cpl model date 0509-12-17 00000s wall clock 2004-10-06 17:47:27 avg dt 8s dt 14s
- (tStamp_write) cpl model date 0509-12-18 00000s wall clock 2004-10-06 17:47:35 avg dt 8s dt 8s

NCAR

Performance: Of Two Minds

- Capability
  - How fast can we run this important job?
  - Can we run this really big problem at all?
- Capacity
  - How much combined work can we get done each day?

NCAR

Performance: Of Two Minds

- Capability
  - How fast can we run this important job?
  - Can we run this really big problem at all?
- Capacity
  - How much combined work can we get done each day?
  - THIS IS THE ONE THAT DRIVES ME MOST DAYS!

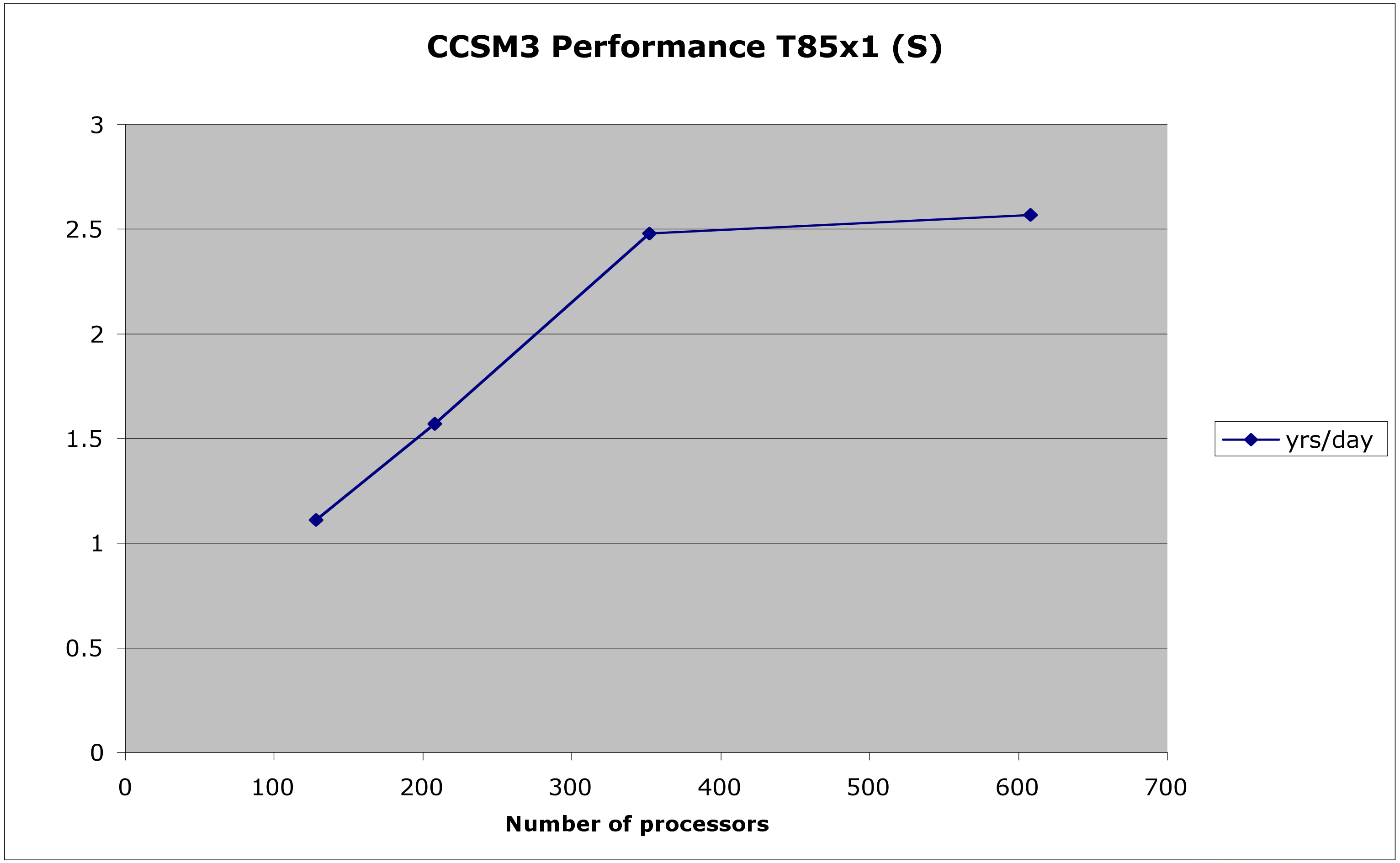
NCAR

Performance Metrics

- Simulated years per wall clock day
  - Optimize for single job maximum performance
- Simulated years per wall clock day per "cpu"
  - Optimize for system aggregate performance

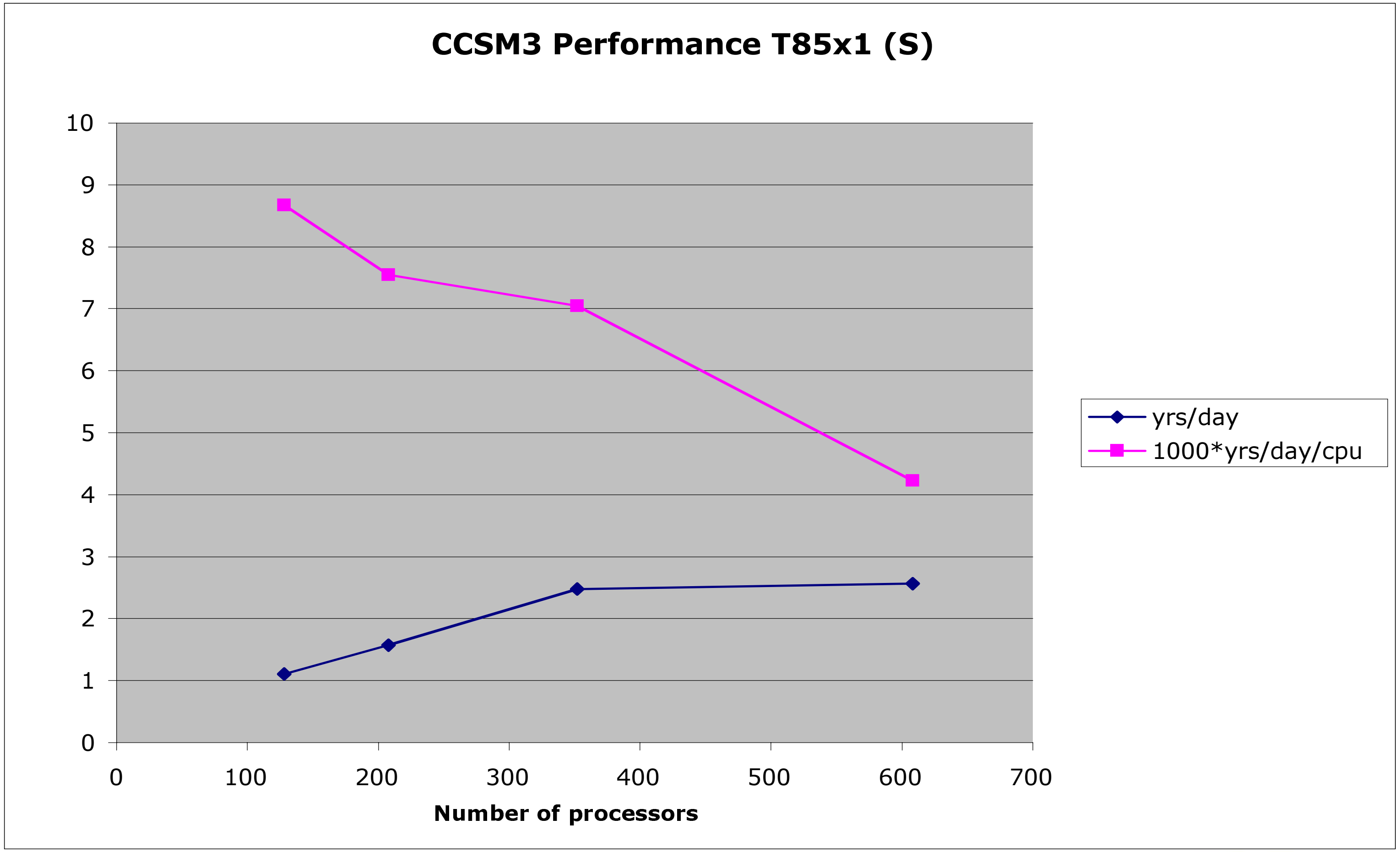
NCAR

Raw Performance



NCAR

Raw Performance vs Efficiency



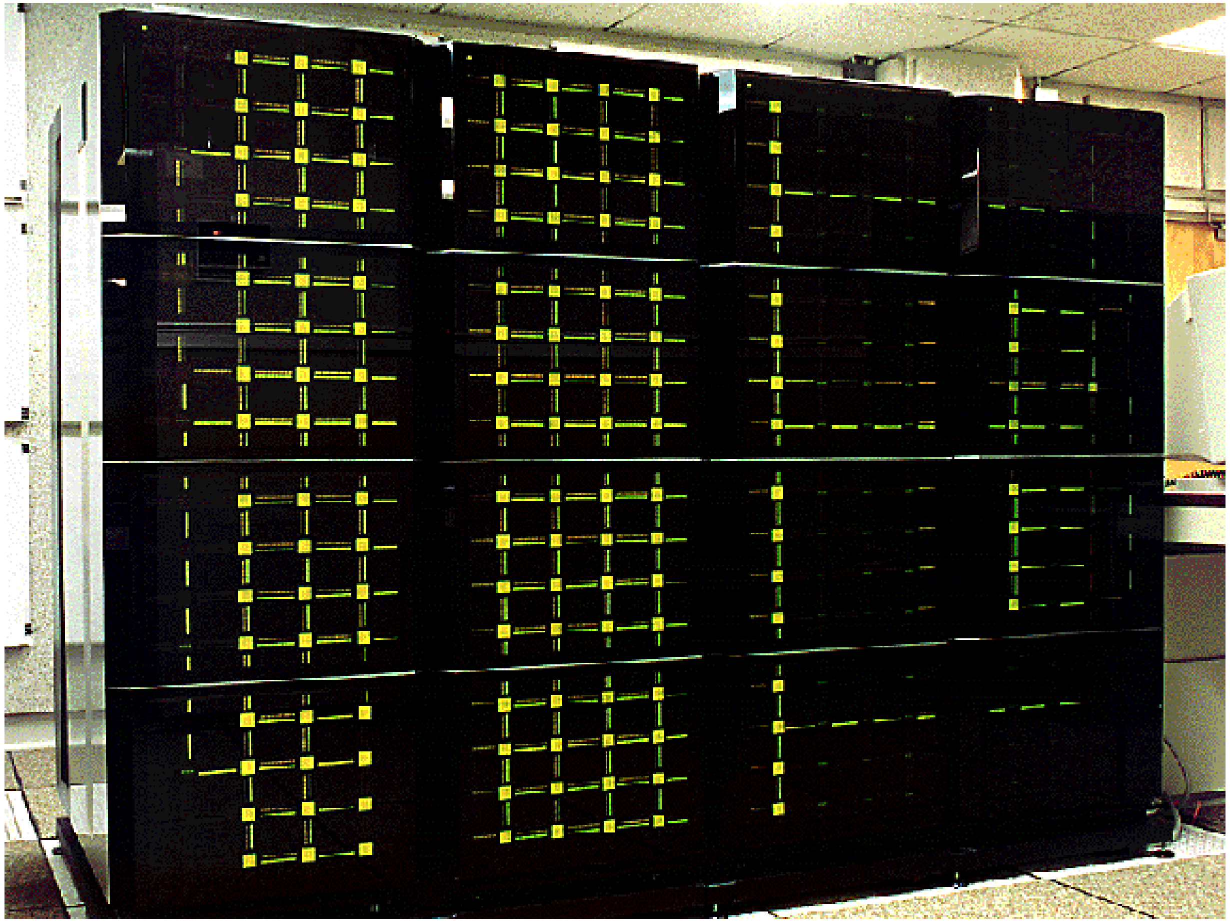
NCAR

I/O Issues

- More a CCSM issue than just X1
  - Want to look at I/O cacher options
    - Better overlap I/O and computation
    - Better insulate computation from I/O congestion
  - Better control over log file output
    - Reduction in number of calls and syncs
    - Compile and runtime controls

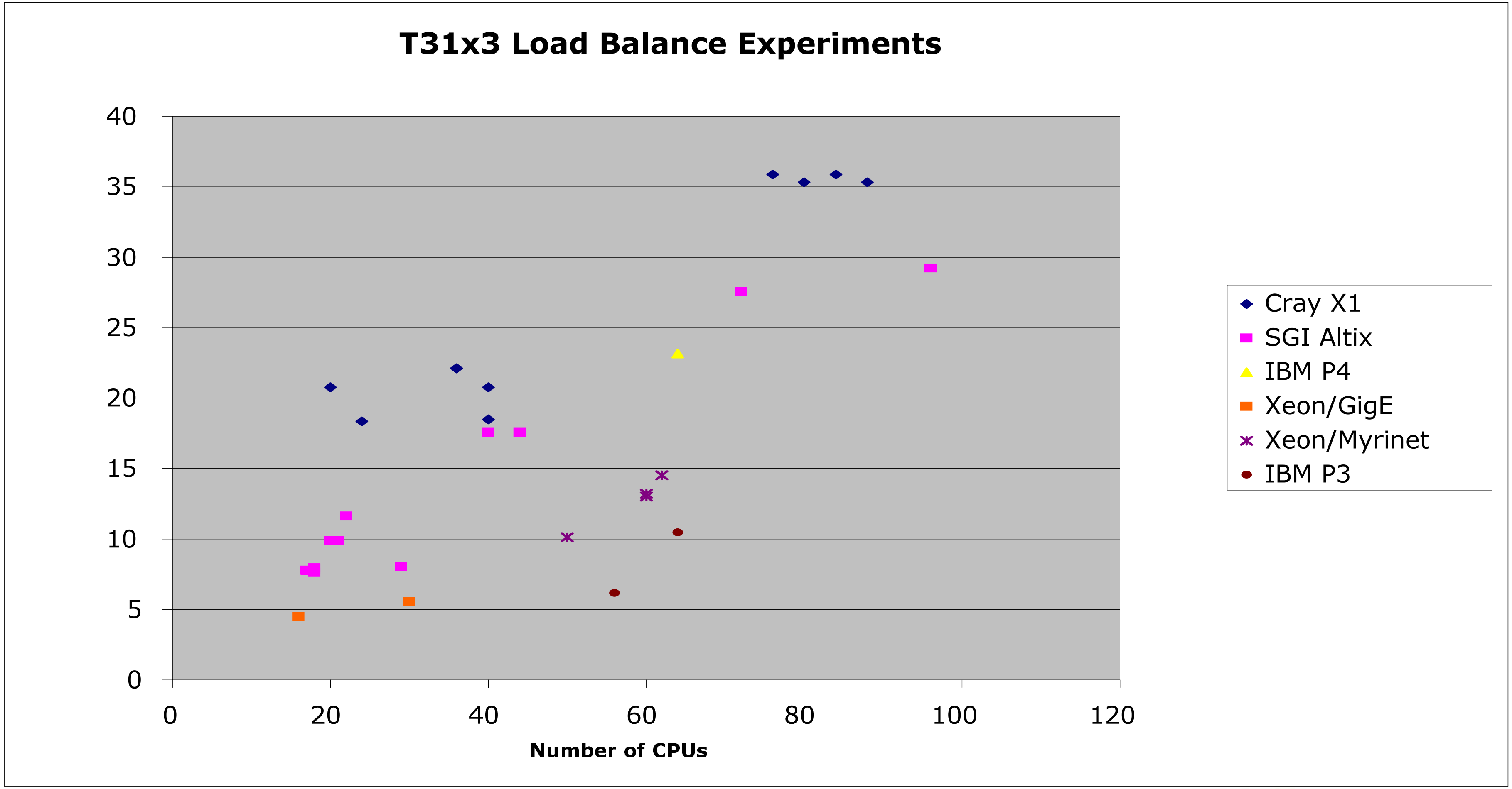
NCAR

Real Lights … I/O Cacher



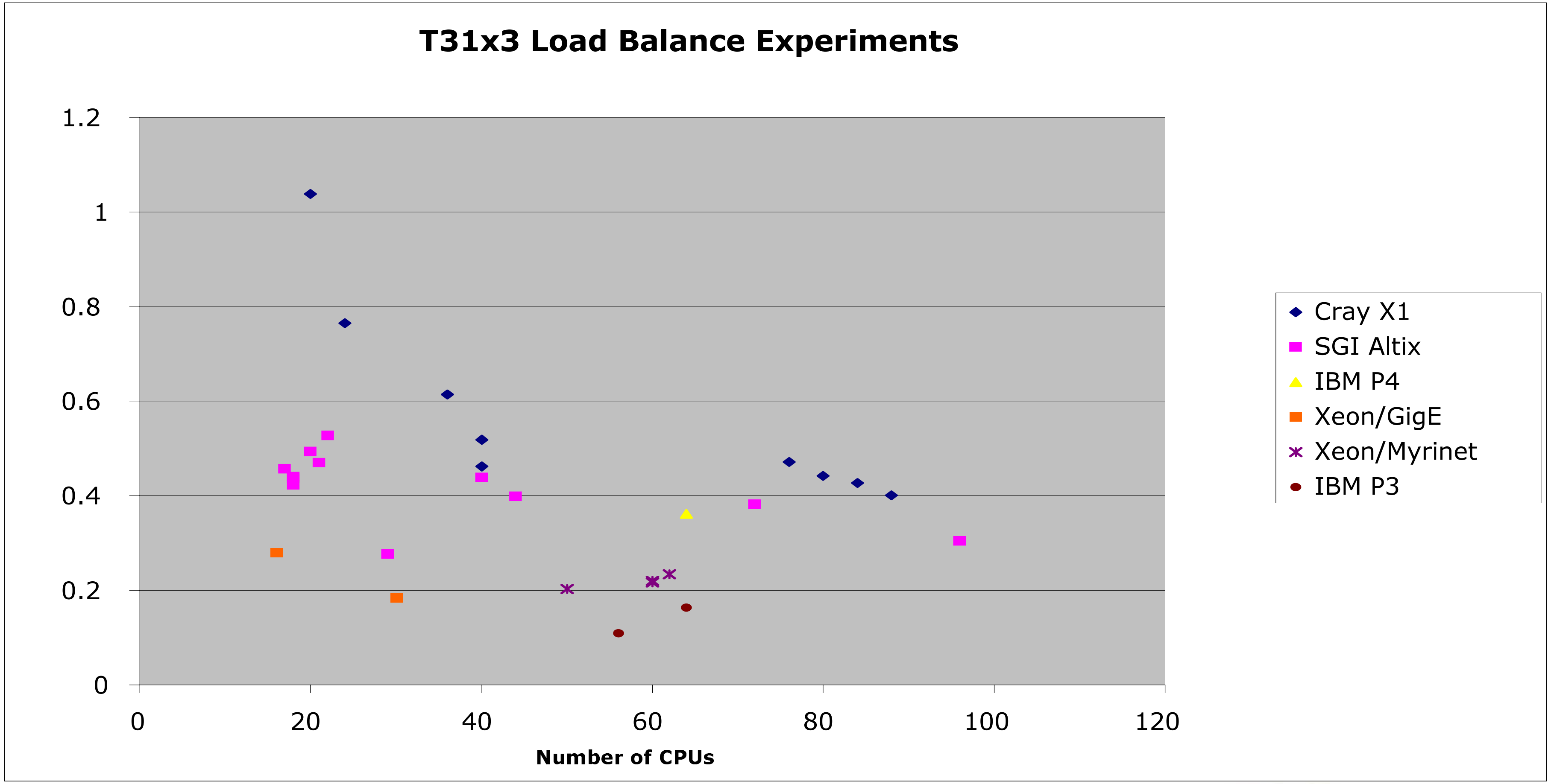
NCAR

T31 Performance

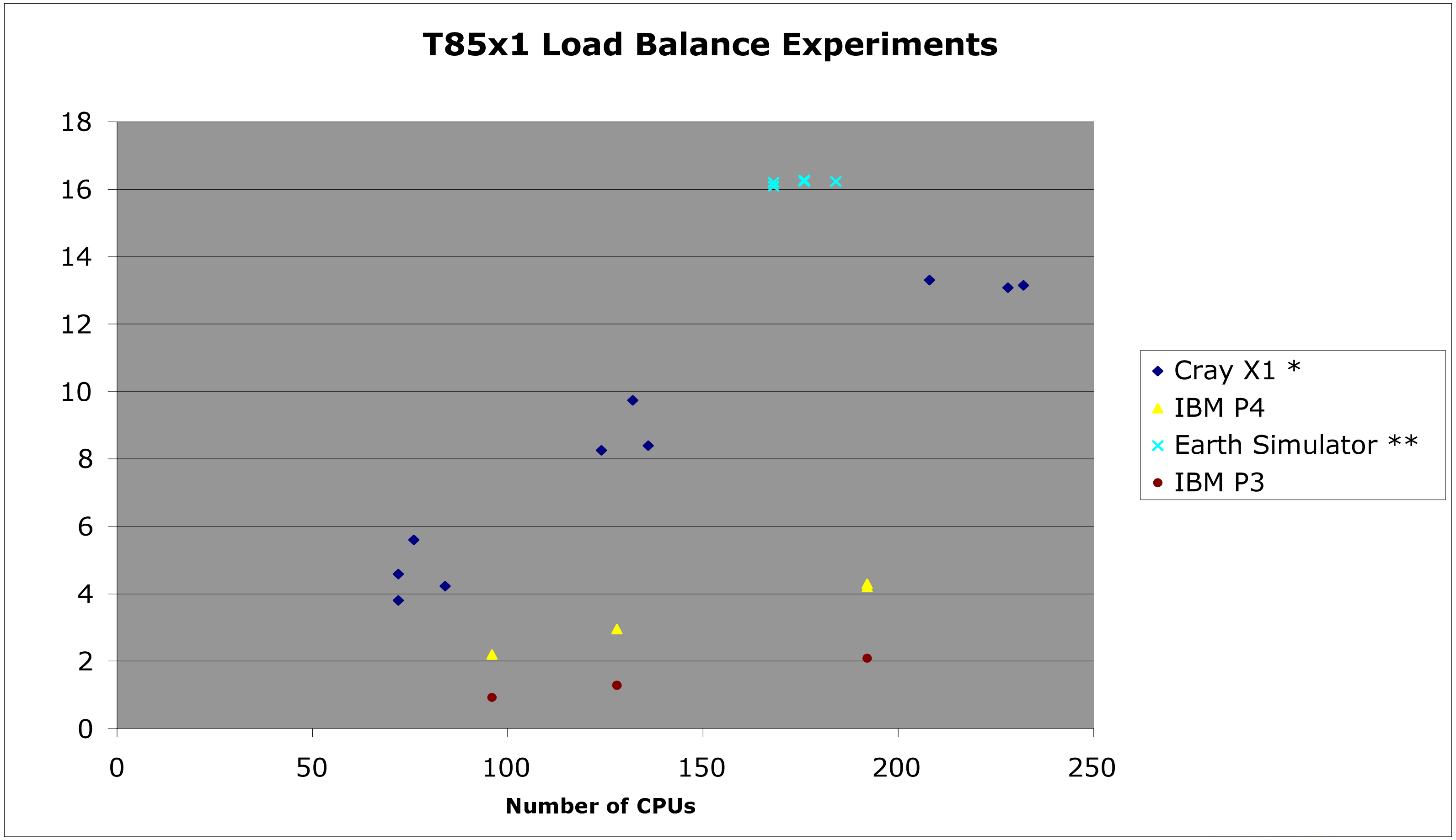


NCAR

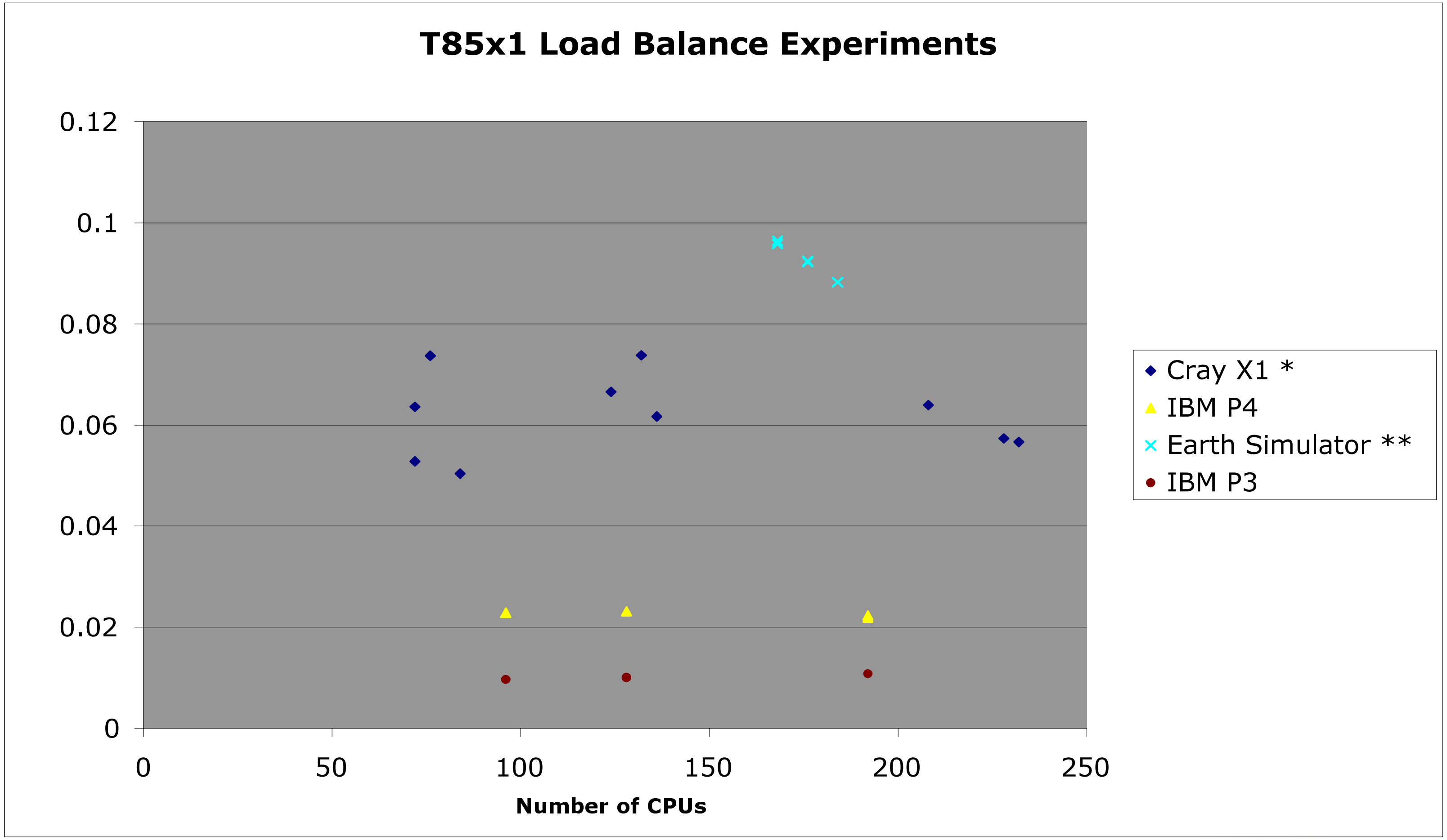
T31 Efficiency



T85 Performance



T85 Efficiency

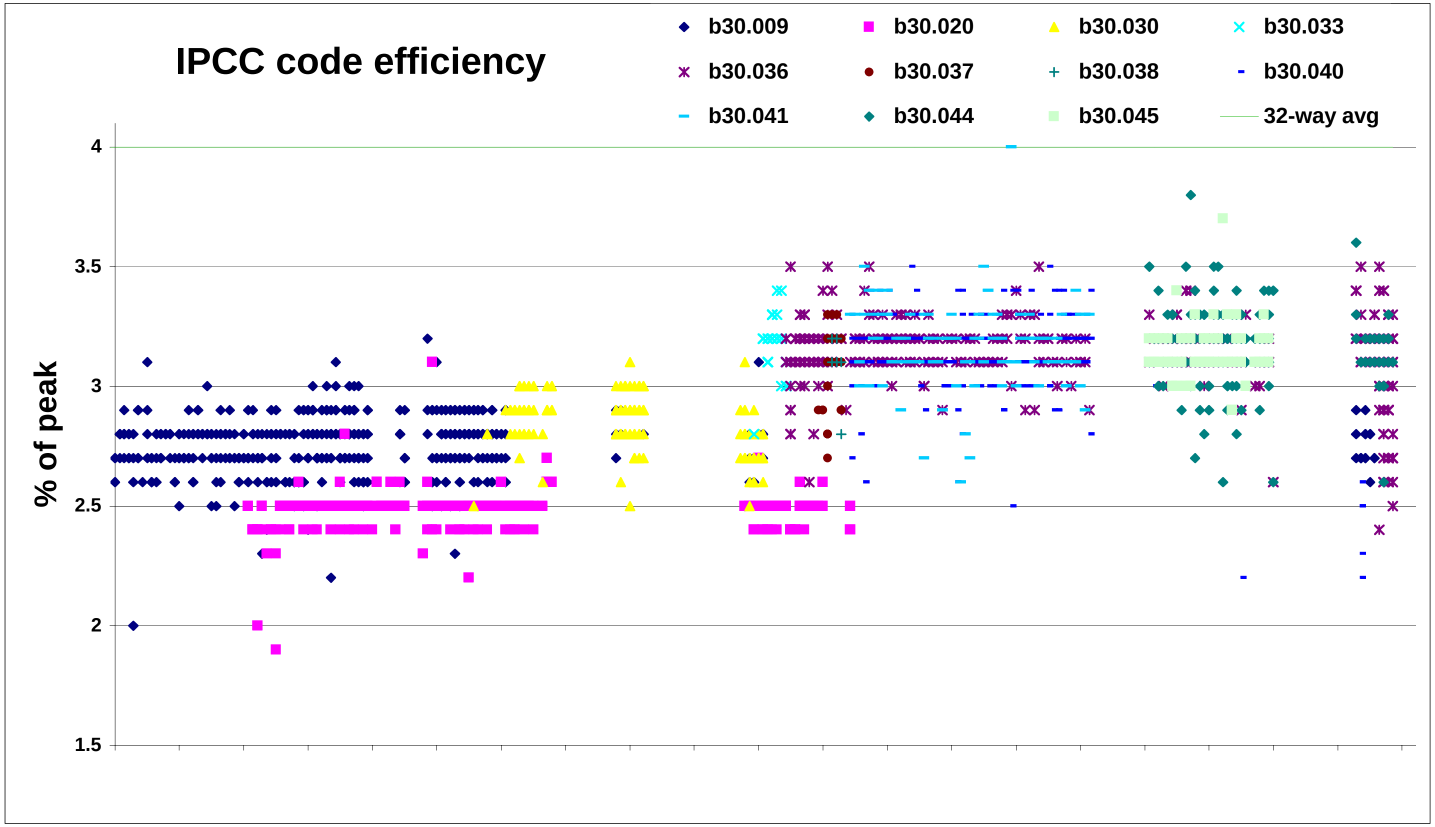


Future Work

- Look at production timing variations … might be more important than CPU speedups!
- Newer software version (includes newer MPI)
- Pat Worley's CAM dynamic load balancing (fast messaging makes this possible on X1)
- Performance Tuning
  - LND and CPL need attention.
  - Look at latest POP and CAM speedups (CAF issue)
- Some additional load balancing exercises
- T85 Validation
- Full Production

NCAR

IBM P4 Percent of Peak



Summary

- Significant work completed
- Things yet to do to bring CCSM into production on the X1
- Need to concentrate on production metric of system performance
- Thanks ORNL and Cray for great support
- Thanks ECMWF

NCAR

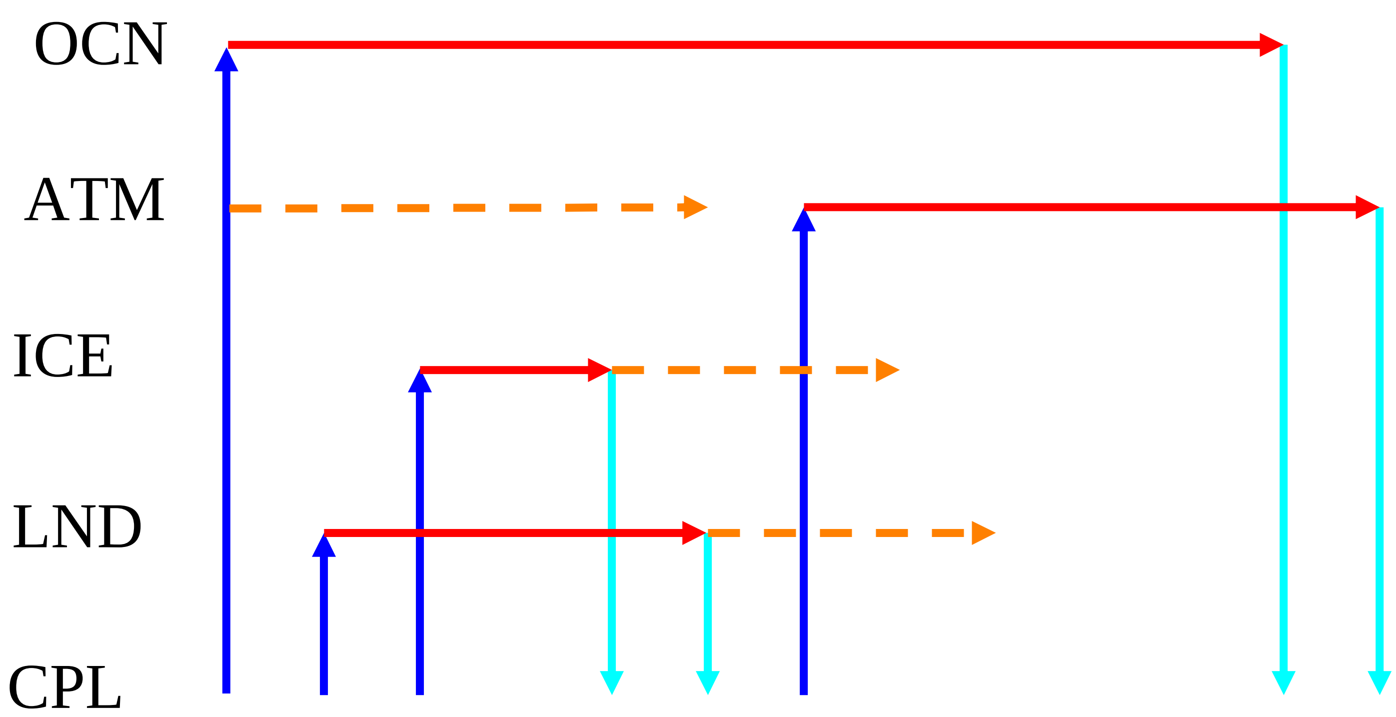
Questions

- CCSM web pages
  - http://www.ccsm.ucar.edu/ccsm3
  - http://www.ccsm.ucar.edu/support_model
    - See CCSM User's Guide
    - See Scripts Tutorial
    - Performance and Platform information will be added
  - http://www.ccsm.ucar.edu/support_model/mach_support.html
- CCSM Bulletin Board
  - http://bb.cgd.ucar.edu
- ORNL web
  - http://www.csm.ornl.gov/evaluation/PHOENIX
- gcarr@ucar.edu

NCAR

Supplemental Charts

CCSM3 Process Flow



CPL sending data to component (state 1)

CPL receiving data from component (state 3)

Component processing data (state 2)

Component processing (state 4)

NCAR

The Balancing Act

- Each component has different scaling attributes in part based on different grid sizes
- System architecture/configuration constraints
- No power of 2 performance charts

NCAR

Load Balancing Example - X1

T31x3	OCN	ATM	ICE	LND	CPL	Tot	Yrs/Day
Case 1	4	16	8	8	4	40	20.76
Case 2	2	16	2	8	8	36	22.12

Case 2 used fewer processors and got better performance

NCAR

Vectorization Process

- For each component model
  - Port to new (vector) system
  - Optimize performance (including vectorization)
  - Merge subset of modifications back into development trunk
  - Validate/Evaluate updated model on all "category 1" platforms
- For CCSM
  - Import updated component models (lags behind individual)
  - Port and optimize scripts and other CCSM infrastructure to new system
  - Verify that CCSM runs correctly in all required configurations and tests
  - Validate climate produced by CCSM
  - Tune configuration to optimize performance on new system

Merge Guidelines and Process

- **Cannot degrade performance significantly on other target systems**
  - **Allowable degradation depends on perceived importance (availability) of given platform for science.**
- **Cannot alter solution (bit-for-bit) on other platforms**
  - **Can be relaxed when climate validation needs to be repeated on other platforms anyway.**
- **For CAM and CLM, solution must be independent of number of processors (i.e., reproducibility).**
- **Limited amounts of architecture-dependent code allowed (i.e., no large scale #ifdef NEC/CRAY/IBM sections)**
  - **This is for code maintainability. What is or is not permitted varies among the CCSM working groups.**
- **Actual merge process consists of making a proposal to the relevant component Change Review Board, followed by some period of negotiation.**