

Scalability and Performance of Salinas on the Computational Plant

Ron Brightwell

**Computation, Computers,
and Math Center**

Manoj Bhardwaj

Garth Reese
**Engineering Sciences
Center**

Sandia National Laboratories

Outline

- **ASCI/Red**
- **Computational Plant (Cplant™)**
- **Salinas**
- **Performance**
- **Scaling issue**

ASCI/Red Hardware

- 4640 compute nodes
 - Dual 333 MHz Pentium II Xeons
 - 256 MB RAM
- 400 MB/sec bi-directional network links
- 38x32x2 mesh topology
- Red/Black switchable
- First machine to demonstrate 1+ TFLOPS
- 2.38/3.21 TFLOPS
- Deployed in 1997



ASCI/Red Compute Node Software

- **Puma lightweight kernel**
 - **Follow-on to Sandia/UNM Operating System (SUNMOS)**
 - **Developed for 1024-node nCUBE-2 in 1993 by Sandia/UNM**
 - **Ported to 1800-node Intel Paragon in 1995 by Sandia/UNM**
 - **Ported to Intel ASCI/Red in 1996 by Intel/Sandia**
 - **Productized as “Cougar” by Intel**

ASCI/Red Software (cont'd)

- **Puma/Cougar**

- Space-shared model
- Exposes all resources to applications
- Consumes less than 1% of compute node memory
- Four different execution modes for managing dual processors
- Portals 2.0
 - High-performance message passing
 - Avoid buffering and memory copies
 - Supports multiple user-level libraries (MPI, Intel N/X, Vertex, etc.)

Computational Plant (Cplant™)

- **Cplant™ is a concept**
 - Provide computational capacity at low cost
 - Build MPPs from commodity components
 - Follow ASCI/Red model and architecture
- **Cplant™ is an overall effort:**
 - Multiple computing systems in NM & CA
 - Multiple projects
 - Portals 3.x message passing (with UNM and others)
 - Cluster infrastructure tools (with HPTi)
 - System integration & test
 - Operations & management
- **Cplant™ is a software package**
 - Available under GNU LGPL
 - Licensed to Unlimited Scale, Inc.

Cplant™ Approach

- **Hybrid approach combining commodity cluster technology with MPP technology**
- **Emulate the Intel ASCI/Red environment**
 - Partition model (functional decomposition)
 - Space sharing (reduce turnaround time)
 - Scalable services (allocator, loader, launcher)
 - Complete compute node resource dedication
- **Use Existing Software when possible**
 - Red Hat distribution, Linux
 - Software developed for Intel ASCI/Red

Cplant™ Systems (SNL/NM)

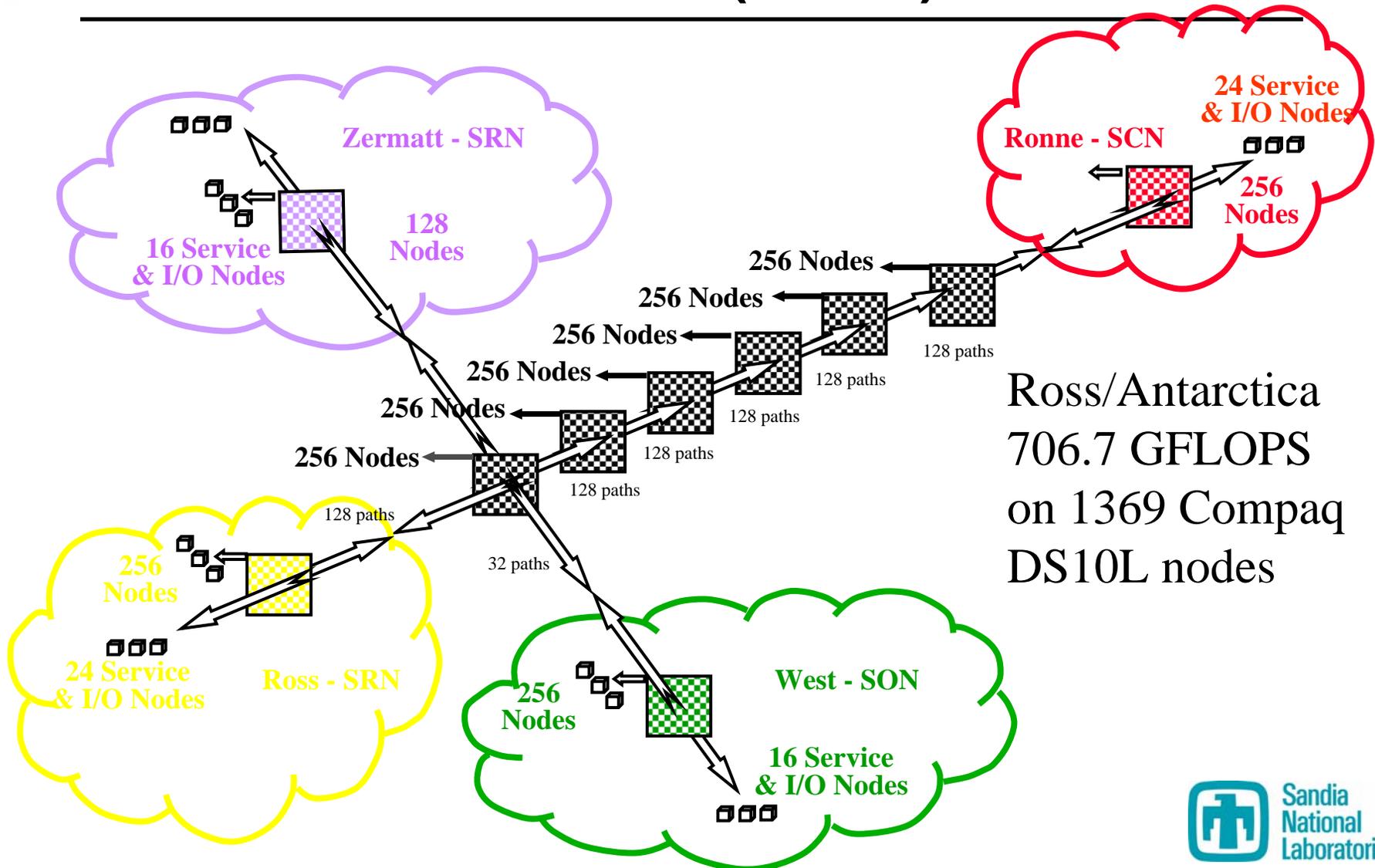
	Vendor	Compaq	Compaq	Compaq	Compaq	Compaq	Digital	Dell		
	Model	DS10L	DS10L	DS10L	XP1000	XP1000	500au	PowerEdge		
	CPU	Alpha EV67	Alpha EV6	Alpha EV6	Alpha EV6	Alpha EV6	Alpha EV56	Pentium III		
	CPU Freq.	617 MHz	466 MHz	466 MHz	500 MHz	500 MHz	500 MHz	1 GHz		
	Memory	1 GB	256 MB	1 GB	256 MB	1 GB	192 MB	1 GB		
Name	Network	Number of Nodes							Interconnect	Deployment
Ross	SRN	256							Myrinet LANai-7,9	Production
Ronne	SCN				256				Myrinet LANai-7,9	Production
West	SON			96		160			Myrinet LANai-7,9	Production
Center	Switchable		1536						Myrinet LANai-7,9	Production
Alaska	SRN						258		Myrinet LANai-4	Production
Zermatt	SRN				128				Myrinet LANai-7,9	Development
Iceberg2	SON				14				Myrinet LANai-7	Development
Iceberg	SRN						28		Myrinet LANai-4	Development
Quadrics	SRN							4	Quadrics ELAN-3	Development
Total Peak (GFLOPS)		315.90	1431.55	89.47	398.00	160.00	286.00	4.00		
								2684.93		

Flagship CplantTM Cluster - Antarctica

- 1792+ Compaq DS10L Slates
 - 466 MHz Alpha EV6, 1 GB
 - 617 MHz Alpha EV6, 1 GB
- 590 Compaq XP1000s
 - 500 MHz Alpha EV6, 1 GB
- Myrinet 33MHz 64bit LANai 7.x and 9.x
- Myrinet Mesh64 switches
- 3-D mesh topology
- Classified, unclassified, open, and development network heads

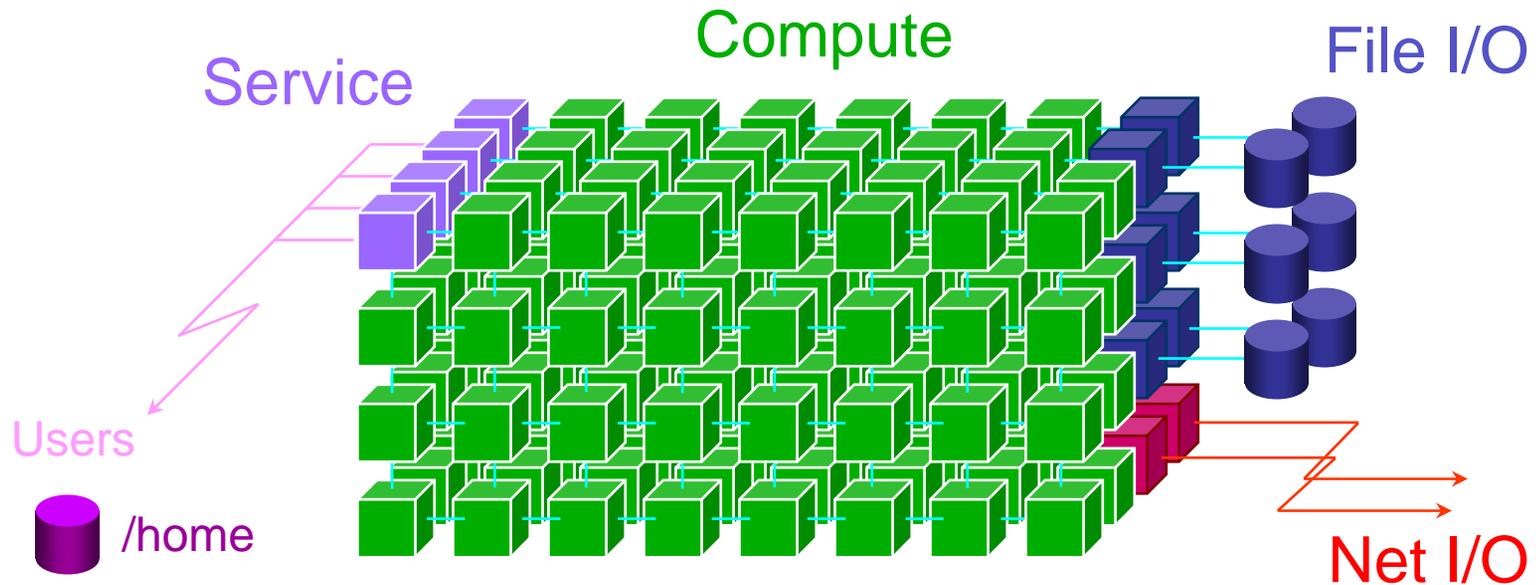


Antarctica (cont'd)

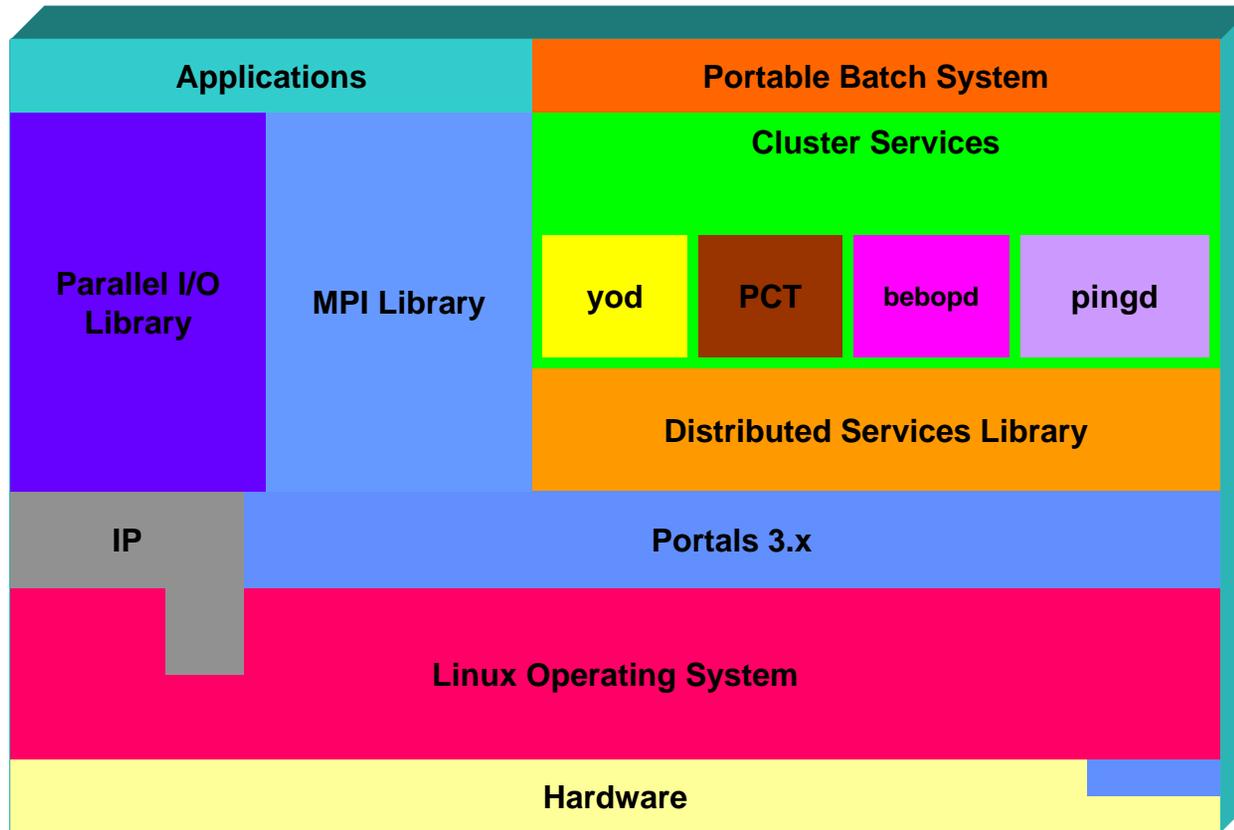


Ross/Antarctica
706.7 GFLOPS
on 1369 Compaq
DS10L nodes

Conceptual Partition Model



Cplant™ System Software



User-Level Software

- **Redirected standard C and I/O libraries**
 - Intercept system calls and let yod handle them
 - Uses a RPC library over Portals 3.x
- **Distributed services library**
 - Used by for communication between runtime system components (yod, pct, bebopd)
 - Implemented over Portals 3.x
- **Puma library**
 - Implements *dclock()* and others for compatibility with Puma
- **Startup code**
 - Initializes the parallel environment for a process

User-Level Software (cont'd)

- **MPI library**
 - Portals 3.x device layer for MPICH 1.2.0
 - Implements peer communication only
- **Dynamic allocation library**
 - New code to support MPI-2 dynamic process creation functionality
 - Not yet deployed in production
- **Job library**
 - Allows for user-implemented job launcher
- **Portals 3.x library**
 - Basic peer communication functions

Runtime System Components

- **Yod**
 - Parallel job launcher
- **Yod2**
 - Parallel job launcher for dynamic process creation
 - Not yet deployed in production
- **PCT**
 - Compute node resource manager
- **Bebopd**
 - Compute node allocator
- **Pingd/Showmesh**
 - Compute node status tool

Other Runtime System Tools

- **cgdb**
 - Application process debugger built on GDB
- **start-tvdsvr**
 - Utility for starting TotalView remote debug servers
- **OpenPBS**
 - Enhancements for non-blocking sockets for increased reliability

Kernel-Level Software

- **Minor patches to Linux 2.2.19 for memory locking and memory mapping**
- **cTask module**
 - Runtime system mappings for processes
 - Process cleanup
- **Portals 3.x module**
 - Implements Portals 3.x functionality
- **RMPP module**
 - Myrinet device driver
 - Reliability and flow control
- **MyrIP module**
 - Provides IP packets over Myrinet

ENFS

- **User-space daemon on I/O proxy nodes**
- **Compute nodes mount from the proxy nodes via Linux VFS**
- **I/O proxy nodes mount from a back-end cluster filesystem (currently XFS on SGI O2K)**

Device-Level Software

- **Myrinet Control Program**
 - Firmware running on LANai processor on NIC
 - Packet engine

Salinas

- **General-purpose, finite element structural dynamics code for massively parallel computers**
- **Currently offers**
 - **Static analysis**
 - **Direct implicit transient analysis**
 - **Eigenvalue analysis for computing modal response, modal superposition-based frequency response, and transient response**

Salinas (cont'd)

- **Includes extensive library of**
 - **Standard one-, two-, and three-dimensional elements**
 - **Nodal and element loading**
 - **Multi-point constraints**

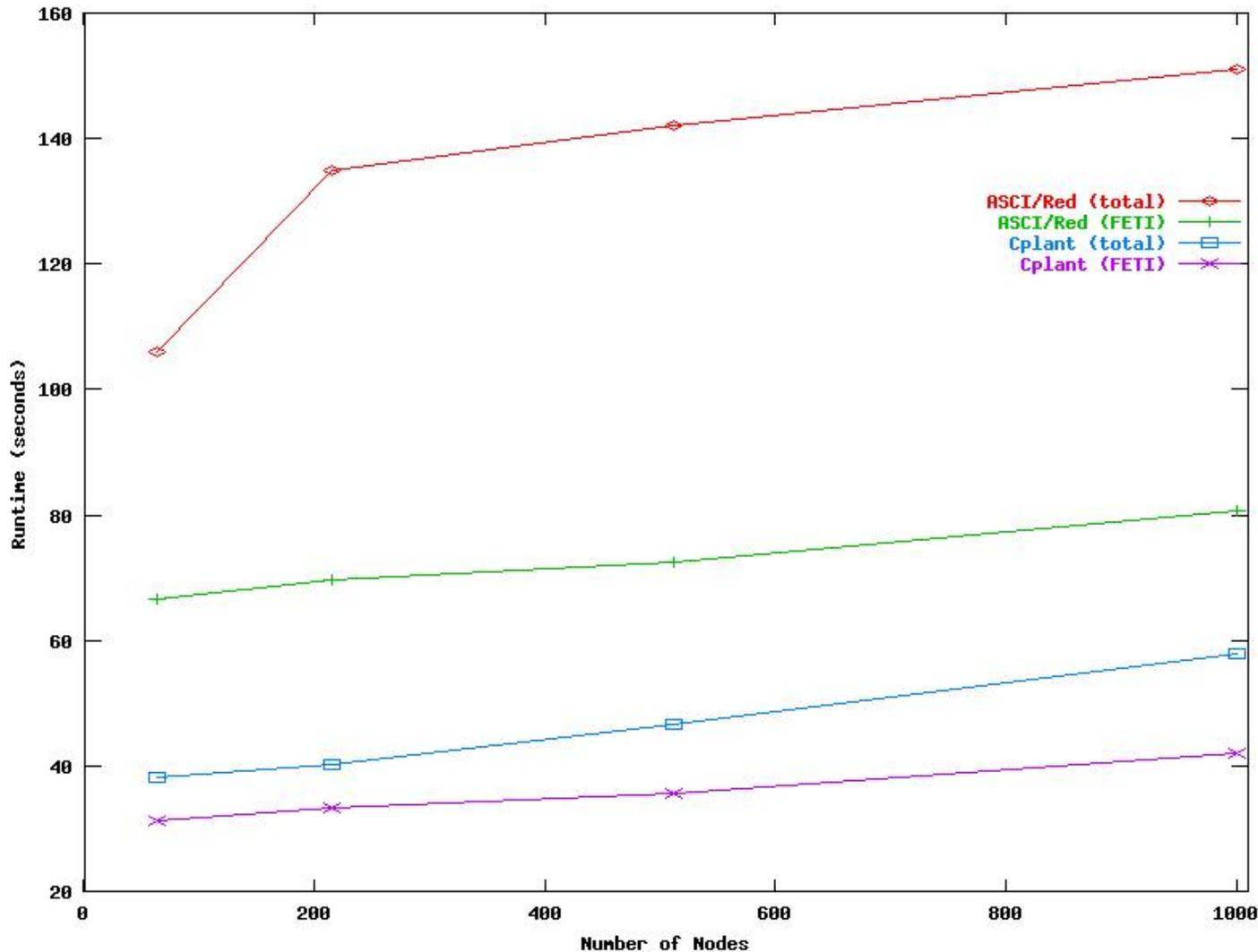
Salinas (cont'd)

- **Solves systems of equations using an iterative multilevel solver specifically designed to exploit massively parallel machines**
 - **Finite Element Tearing and Interconnect (FETI)**
 - **Mature**
 - Versions used in commercial finite element packages
 - **Scalable**
 - As the number of unknowns increases and the number of unknowns per processor stays constant, time to solution stays constant
 - **Accurate**
 - Convergence rate does not deteriorate as the iterates converge

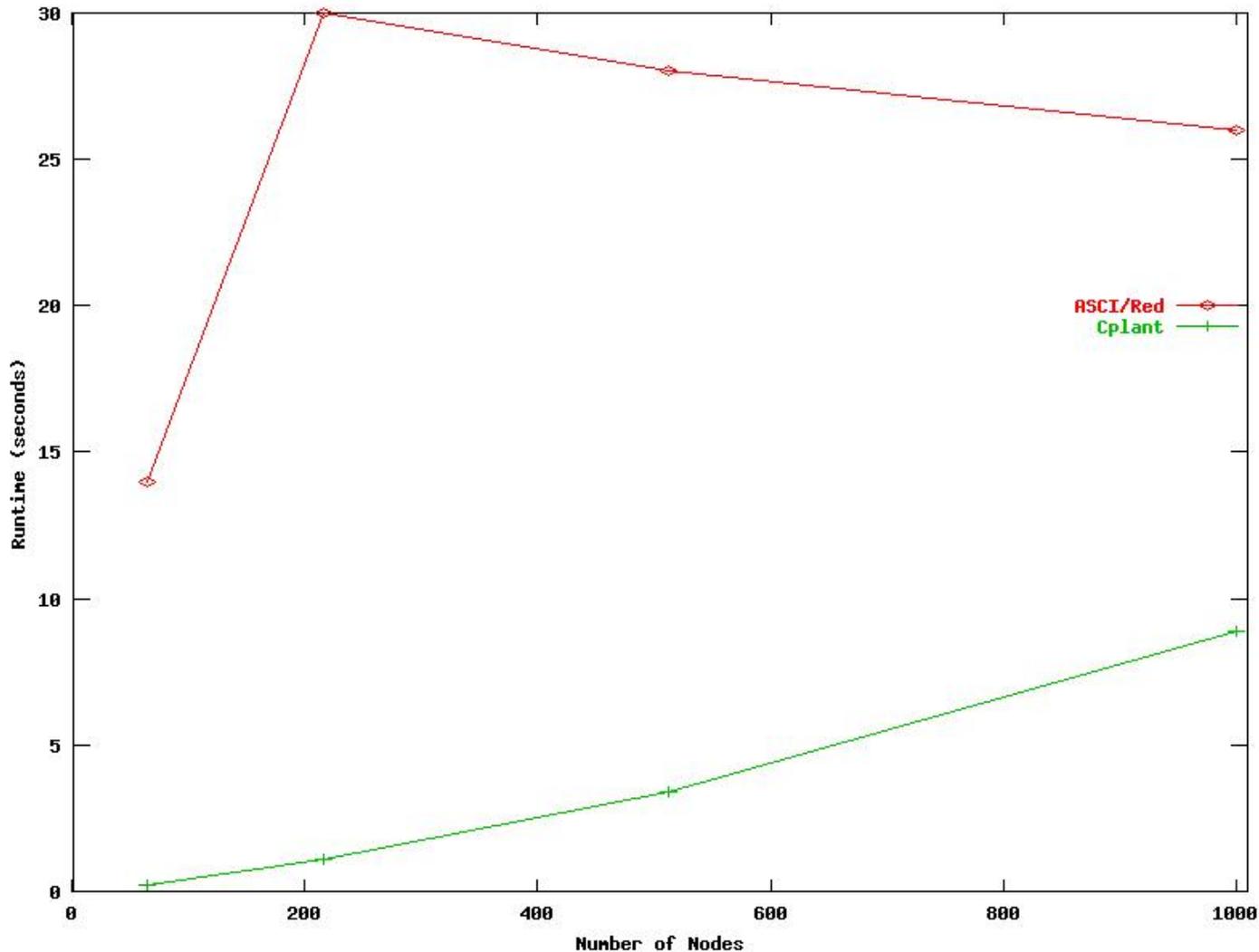
Salinas Sample Problem

- **Small problem size**
 - Only about 3 MB per node
- **Stresses the system more than larger problems**
 - Ratio of computation to communication is larger
 - Higher frequency of message passing
- **Good indicator of scaling efficiency for larger problems**
- **Dedicated time on CplantTM**
- **Non-dedicated time on ASCI/Red using a single processor per node**
- **Average of five runs**

Salinas is 2.5x Faster on Cplant™ at 1000 nodes



I/O Time Is Not Scaling As Well on CplantTM



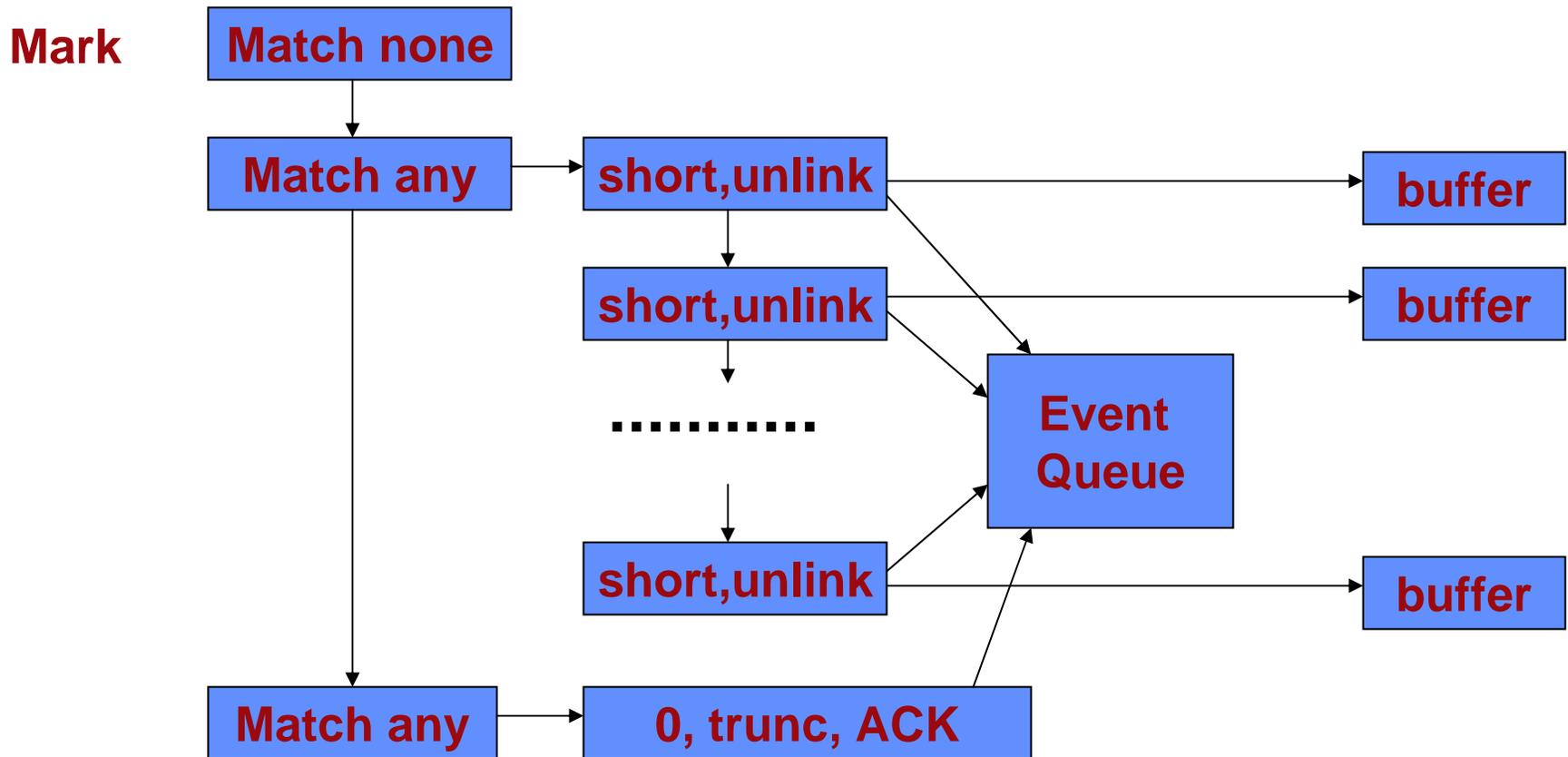
Scaling Issue on Cplant™

- MPI resource exhaustion at several hundred nodes
- “Too many MPI unexpected messages”
 - AKA “Not enough posted receives”
- Short message protocol for MPI is eager
- Unexpected messages are buffered at the receiver
- Initial MPI implementation set aside 1024 8 KB buffers
- A single message of any size consumes a buffer

- MPI_Gather() in MPICH 1.2.0 is implemented via N-to-1 algorithm
- Quick workaround was to add an MPI_Barrier() to make MPI_Gather() synchronous

Previous Strategy for Unexpected Messages

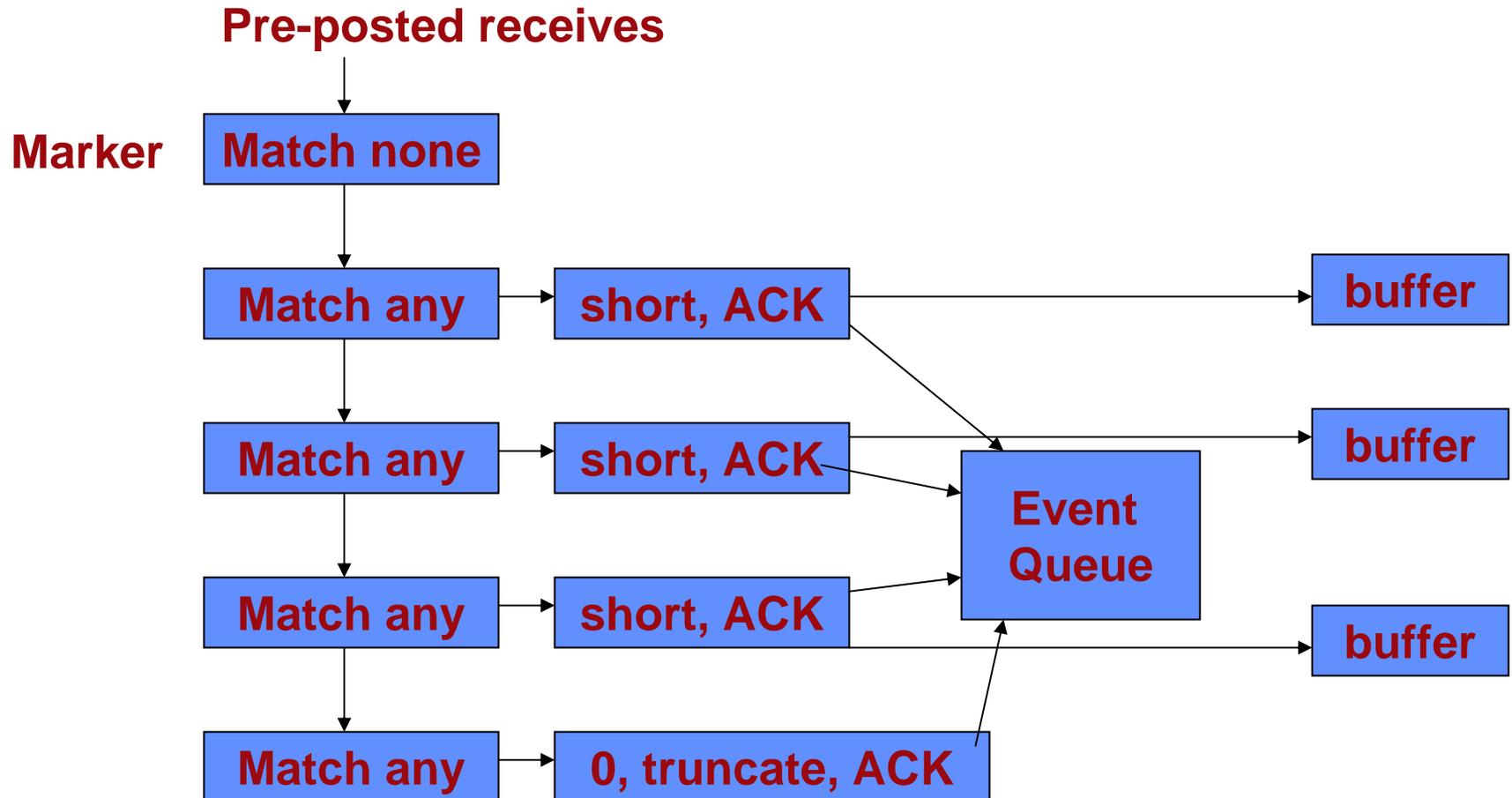
Pre-posted



Limitations

- **Limited number of unexpected messages allowed due to kernel (or NIC) memory resources**
- **Any size unexpected message consumes an unexpected message slot, even zero-length**
- **Unexpected message limit based on count rather than size**
- **Consumes a significant amount of Portals resources**
 - 1025 memory descriptors

Current Strategy



Advantages

- **More efficient use of unexpected message memory**
 - A zero-length message doesn't consume any memory
 - Limitation becomes space rather than count
- **Uses only a few Portals resources**
 - Four memory descriptors versus 1025
- **More efficient for NIC-based implementations**

As for Salinas...

- **Change to MPI library had minimal effect on performance**
- **Overhead of extra MPI_Barrier() operation to synchronize MPI_Gather() operation is negligible**

Summary

- **A commodity Linux cluster is able to sustain competitive performance for a real-world code out to 1000 nodes**
- **Cplant™ is a viable large-scale platform**
- **Issues with network resources become important as applications scale**

Acknowledgments

- **Salinas**
 - Organization 9142
- **Cplant™**
 - Organizations 9223, 9224, 9338
- **Portals**
 - Organization 9223
 - University of New Mexico
 - Cluster File Systems, Inc.

<http://www.cs.sandia.gov/cplant>