



The Portals 3.0 Data Movement Layer

Ron Brightwell

Sandia National Labs

Scalable Computing Systems Department

rbbrih@sandia.gov



Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

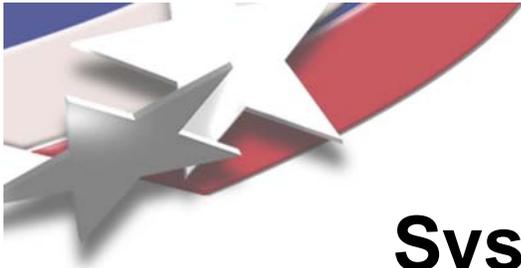




What is Portals?

- **A data movement layer**
 - Data movement is fundamental to more than just parallel applications
 - Runtime systems, I/O systems, parallel debuggers
- **A programming interface**
 - User-level or kernel-level
- **Not a programming model**
- **Not a protocol**





System Software R&D at Sandia

- Intel Paragon
 - 1890 compute nodes
 - 3680 i860 cpu's
 - 143/184 GFLOPS
 - 175 MB/sec network
- SUNMOS lightweight kernel
 - High performance compute node OS for distributed memory MPP's
 - Deliver as much performance as possible to apps
 - Small footprint
 - Started in January 1991 on the nCUBE-2 to explore new message passing schemes and high-performance I/O
 - Ported to Intel Paragon in Spring of 1993





System Software R&D (cont'd)

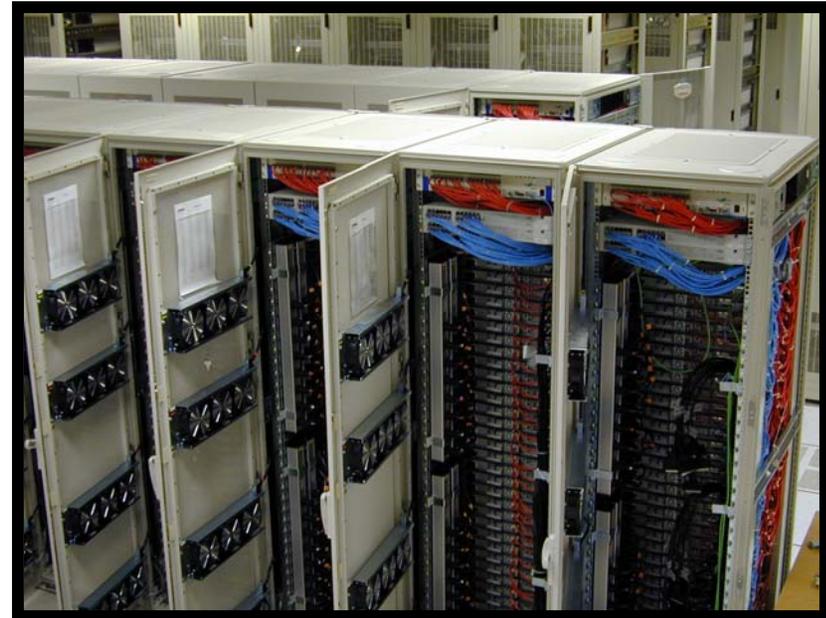
- Intel TeraFLOPS
 - 4576 compute nodes
 - 9472 Pentium II CPU's
 - 2.38/3.21 TFLOPS
 - 400 MB/sec network
- Puma lightweight kernel
 - Multiprocess support
 - Modularized (QK, PCT)
 - Developed on nCUBE-2 in 1993
 - Ported to Intel Paragon in 1995
 - Ported to Intel TFLOPS in 1996 (Cougar)
 - Portals 1.0
 - User/Kernel managed buffers
 - Portals 2.0
 - Avoid buffering and memory copies

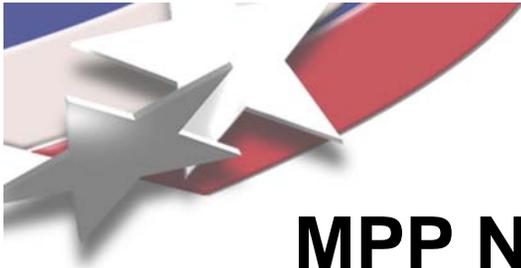




System Software R&D (cont'd)

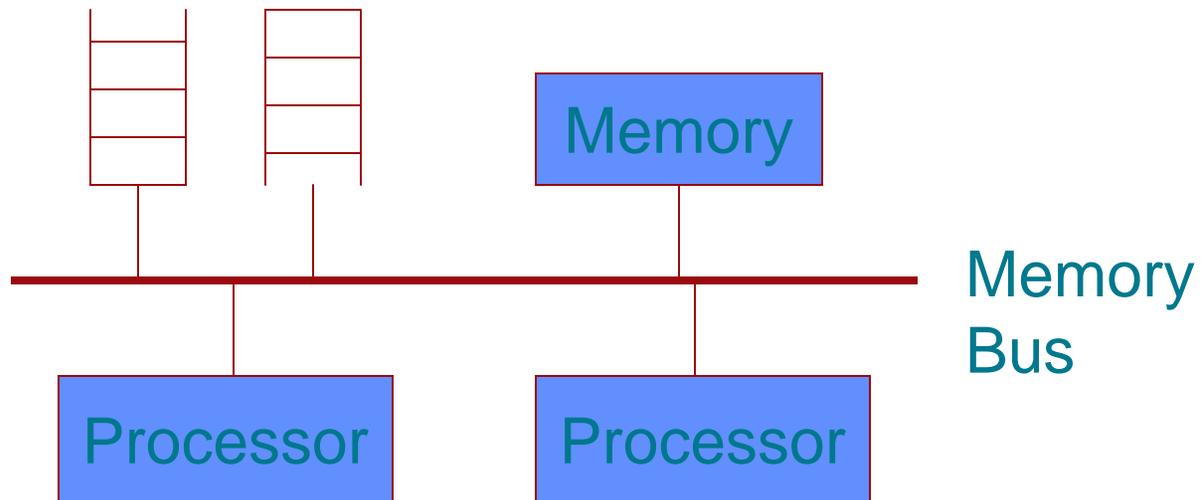
- **Computational Plant**
 - 1,792 compute nodes
 - ~2 TeraFLOPS peak
 - 706+ GFLOPS on 1369 nodes
 - 120 MB/sec network
- **Started in late 1997**
- **Linux operating system**
 - Leverage commodity OS
- **Scalable runtime system**
- **Portals 3.0**
 - New API for commodity hardware





MPP Network: Paragon and ASCI/Red

Network FIFOs





Background: Portals 0

- **SUNMOS (Sandia/UNM OS)**
 - Modeled on Vertex (the OS for the nCUBE)
 - Dynamic allocation for incoming messages
- **Experiments**
 - Multiple paths
 - Pre-posted receives
 - Use of message co-processor
- **nCUBE-2 and Intel Paragon**
 - Direct access to network FIFO's
 - Message co-processor (Paragon)





Background: Portals 1.0

- **Moved all message reception structures to user-space**
- **Types of portals**
 - **Kernel-managed**
 - **Single-block**
- **Never implemented**
- **Published ☺**





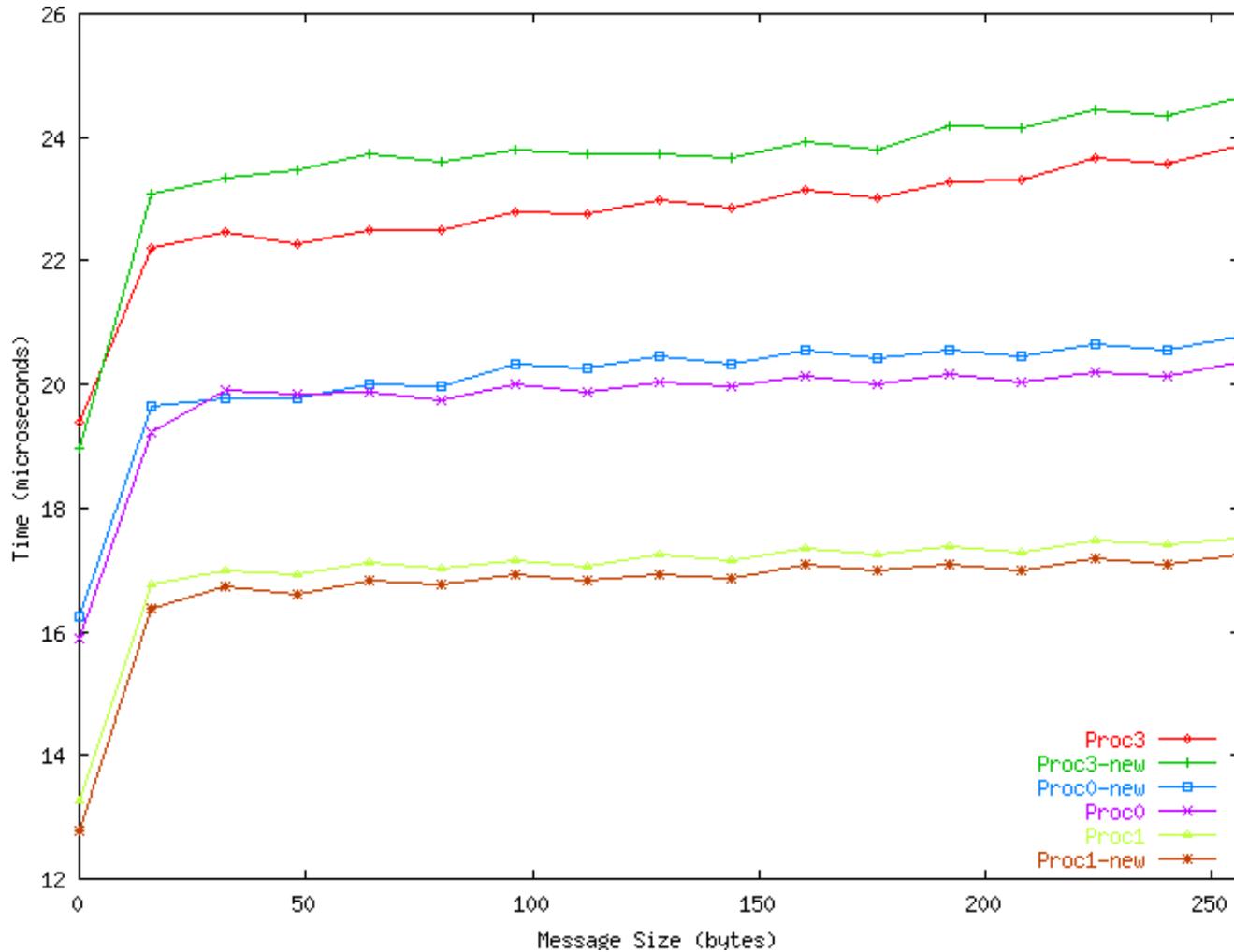
Background: Portals 2.0

- **Separate message selection from memory descriptors**
- **More types of memory descriptors**
 - Kernel-managed (dynamic)
 - Single block
 - Independent block
 - Combined block (never fully implemented)
- **Intel ASCI/Red**
 - Direct access to network FIFO's
 - Message co-processor



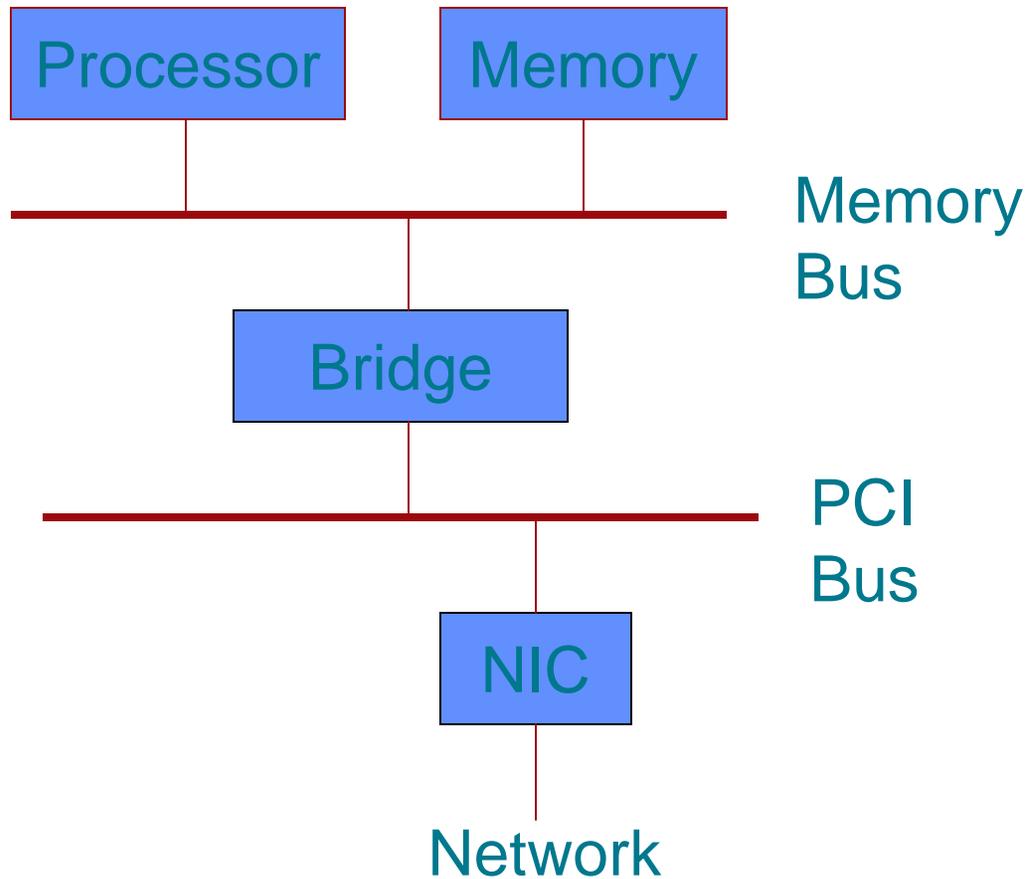


ASCI/Red Ping-Pong MPI Latency Performance





Commodity Network: Myrinet, Quadrics, ...





Problems with Portals 2.0

- **No API**
 - Data structures entirely in user-space
 - Protection boundaries have to be crossed to access data structures
 - Data structures must be copied, manipulated, and copied back
 - Requires interrupts
- **Address validation/translation on the fly**
 - Incoming messages trigger address validation
 - Doesn't fit Linux model of validating addresses on a system call for the currently running process





Portals 3.0

- **Operational API**
- **Unified memory descriptors**
- **Commodity processors and networks**
 - **Alphas, IA-32, IA-64, etc.**
 - **Linux OS with modules**
 - **Myrinet, Quadrics, etc.**
 - **DMA access to memory**
- **Fundamental change**
 - **NIC doesn't have logical address maps**
 - **NIC access to memory needs to be carefully managed**





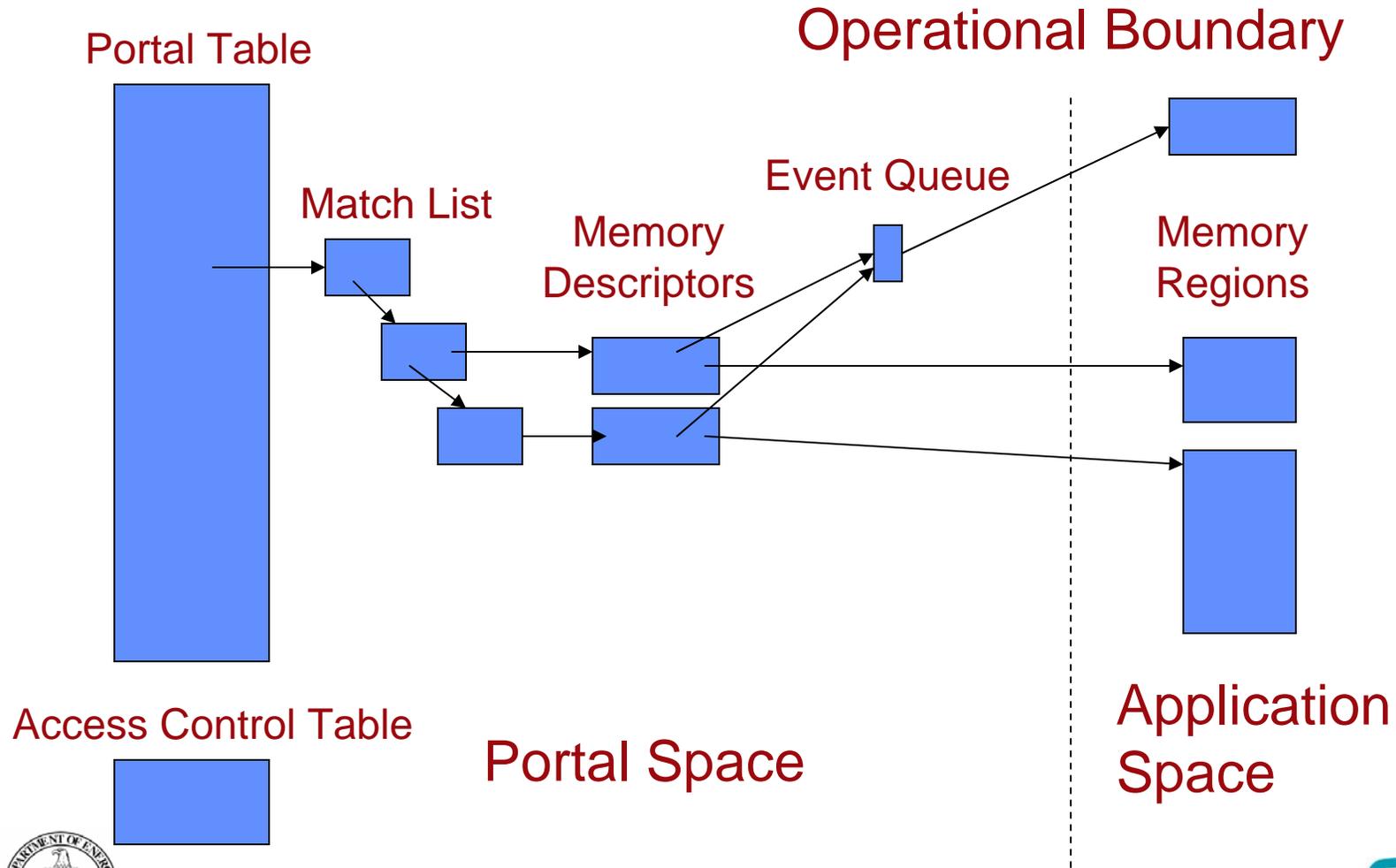
Features

- **Reliable, in-order delivery**
- **Well-defined transport failure semantics**
- **Expected messages**
- **One-sided operations**
 - **Put and Get**
- **Zero-copy message passing**
 - **Increased bandwidth**
- **OS-bypass implementation**
 - **Reduced latency**
- **Application-bypass semantic**
 - **No polling, no threads**
 - **No host CPU utilization**
 - **Reduced software complexity**



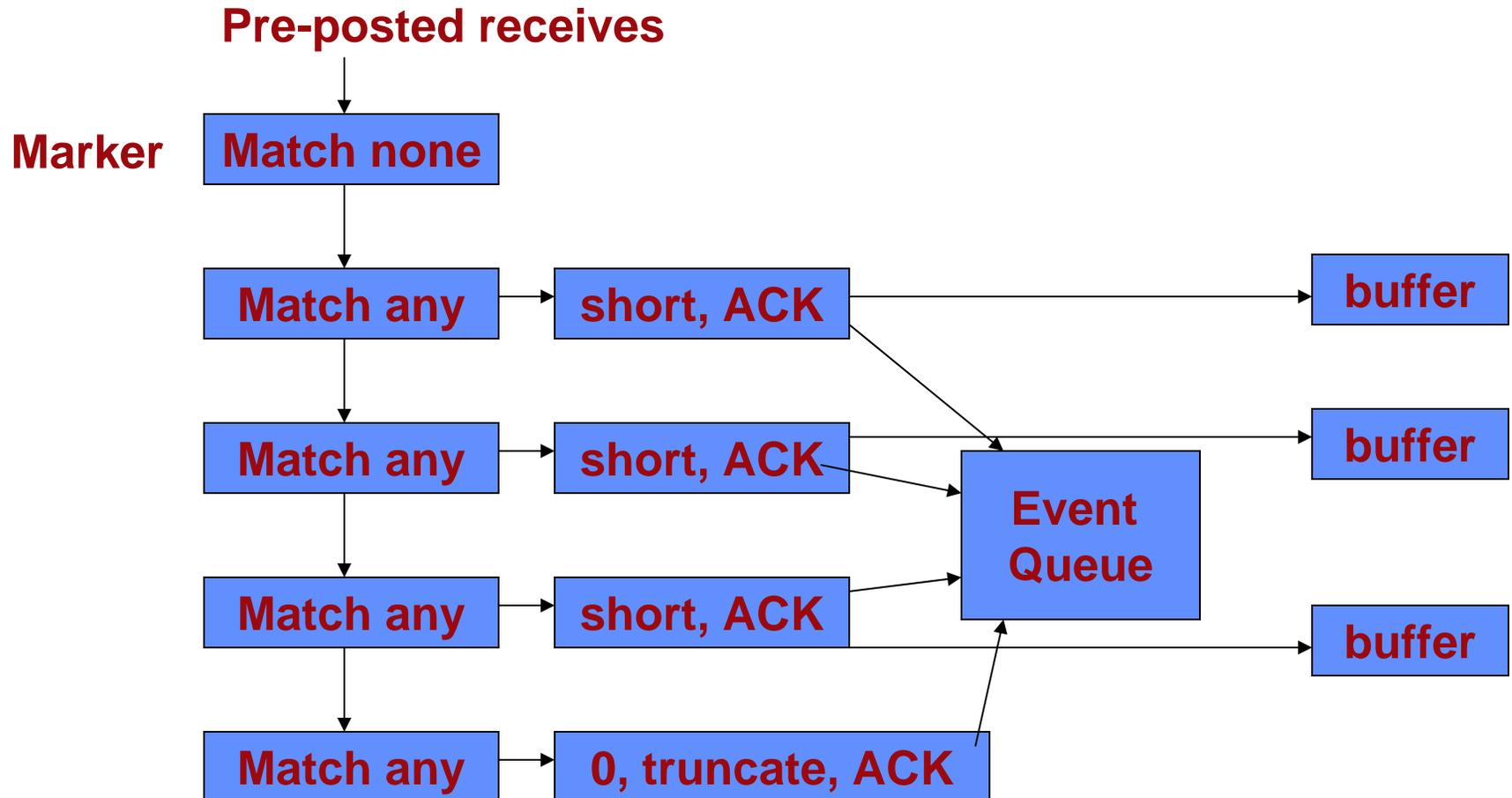


Portal Addressing





Example: Implementing MPI





What Makes Portals Different?

- Provides elementary building blocks for supporting higher-level protocols well
- Allows structures to be placed in user-space, kernel-space, or NIC-space
- Allows for OS-bypass implementations
- Receiver-managed offset allows for efficient and scalable buffering of unexpected messages
- Supports multiple protocols within a process
- Runtime system independent
- Well-defined failure semantics
- Application-bypass semantic is a good thing





MPI Double-Buffer Benchmark

Rank 0

```
isend A;  
isend B;  
for ( ) {  
    fill A; wait CTS A;  
    isend A;  
  
    fill B; wait CTS B;  
    isend B;  
}
```

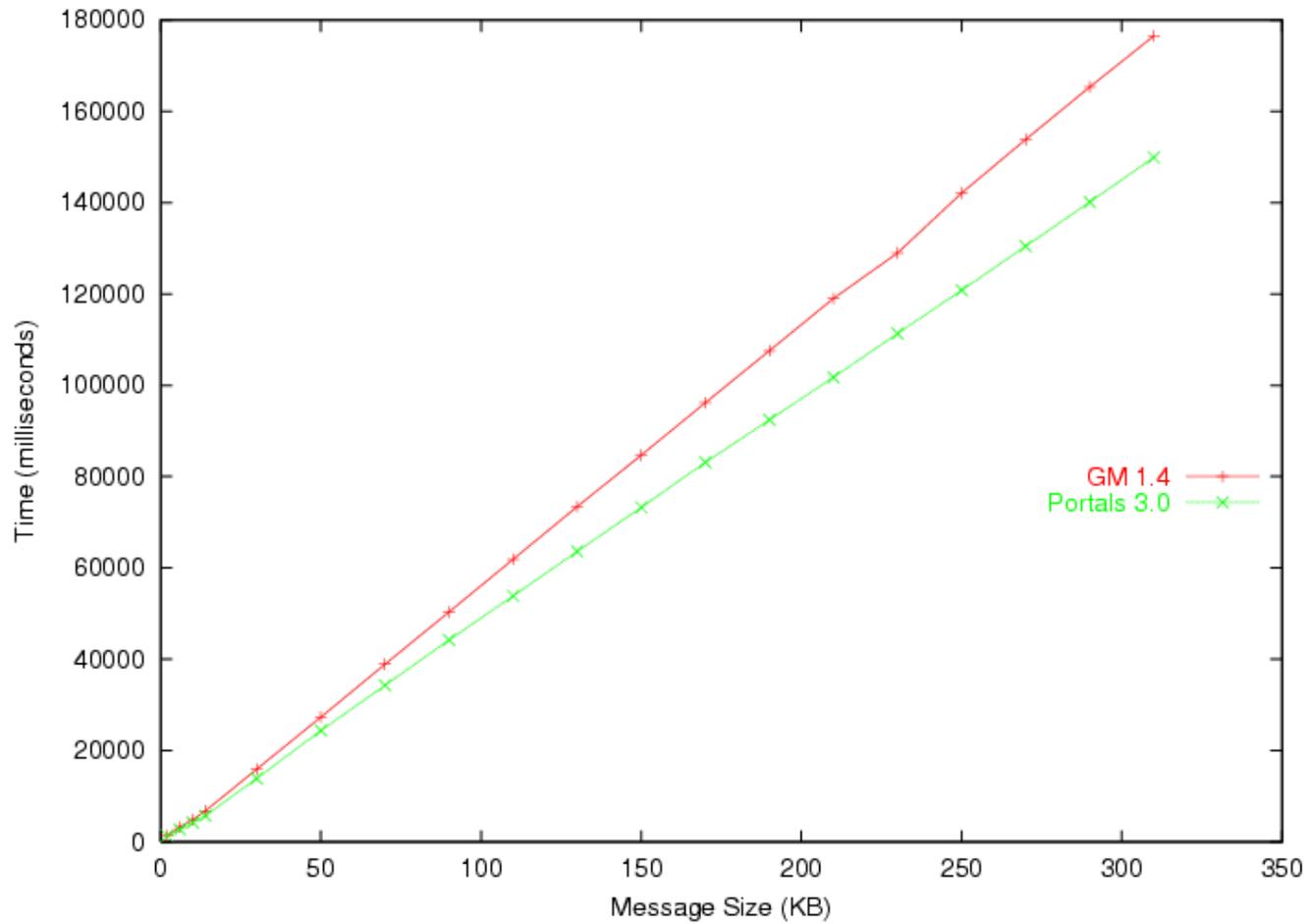
Rank 1

```
start = get_time();  
for ( ) {  
    wait A; sum A;  
    isend CTS A;  
  
    wait B; sum B;  
    isend CTS B;  
}  
end = get_time();
```





MPI Double-Buffer Performance





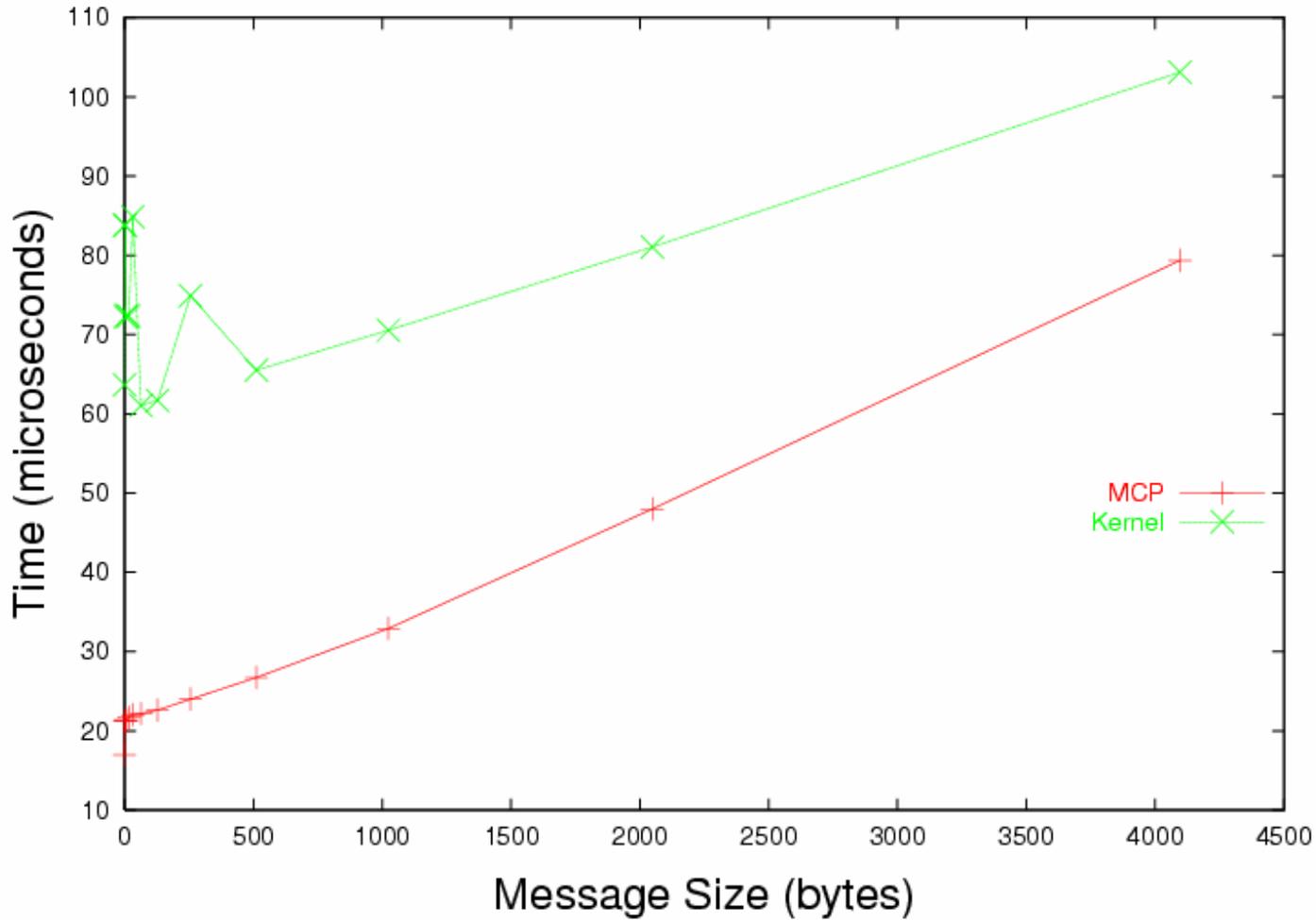
Current NAL Implementations

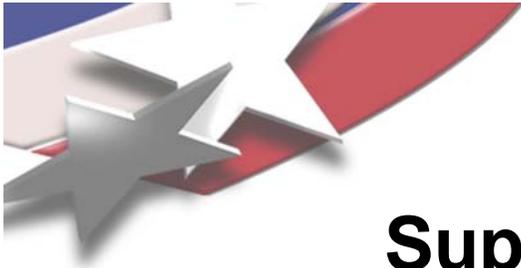
- **RTS (Sandia)**
 - Linux kernel module that does reliability and flow control
 - Can use any Linux network device (skbufs)
- **IP (Sandia)**
 - Reference implementation
 - Really UNIX Pipes
- **Quadrics ELAN3 (Sandia)**
 - Uses ELAN thread and DMA queues
- **Myrinet MCP (Sandia)**
 - Designed to work with Cougar
- **Alteon GigE (University of New Mexico)**
- **In-kernel TCP/IP (Peter Braam, Cluster File Systems, Inc.)**
- **Quadrics ELAN Kernel Comms (Marcus Miller, LLNL)**
- **Quadrics Tports (Unlimited Scale, Inc.)**





Portals 3.0 Myrinet MCP Implementation

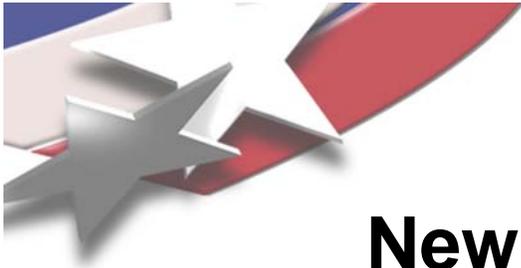




Supported Higher-Level Systems

- High-level message passing libraries
 - MPICH
 - MPI/Pro, ChaMPIon/Pro
 - RPC
 - InterComm
 - Intel NX
 - nCUBE Vertex
 - Initial MPI-2 one-sided specification from March 1996
 - Cray SHMEM variant
- Cplant™ Runtime system
 - Distributed server library
 - Dynamic process creation library
- File systems
 - ASCI/Red fyod
 - Cplant™ ENFS
 - Lustre
- In progress
 - MPI-2 one-sided (MSTI)
 - TotalView channel implementation (Sandia)





New OS Initiative - Filling the Gap

- **Two most scalable systems did not use full UNIX-based operating systems (ASCI/Red,Cray T3)**
- **Current and future initiatives for tera-scale and peta-scale systems are focusing more on hardware architecture and programming models, less on operating systems and runtime system support**
- **Still many basic research questions regarding operating systems and runtime systems for 100 teraOPS and petaOPs platforms (extensibility, fault tolerance, etc.)**
- **Need to start gathering support for new initiative now**
- **First workshop in March associated with WIMPS**
- **Next workshop coming soon**





Question #3

- **How to develop middleware and run-time support so that the abstractions of the programming models can be implemented in a portable and high-performance manner while remain compatible with future networking and computing technologies?**
 - **Develop abstractions that map well to future hardware**
 - **Develop abstractions that future hardware can map well to**
 - **Well-defined components and abstractions**
 - **Well-defined interfaces between components**
 - **Discourage vendors from providing the entire system**

