Cray X1 Implementation

Tom Goozen
PE Libraries Manager

Cray X1 MPI Implementation

Agenda:

- Cray's commitment to MPI
- Taking Advantage of the hardware
- New algorithms implemented
- User coding/algorithmic suggestions
- Questions

Cray X1 MPI Implementation

- Cray's commitment to MPI
  - – Large customer following
  - – Customers require performance
  - – Mixing parallel programming models

Cray X1 MPI Implementation

- All MPI 1.2 functionality
- Most MPI 2 functionality
  - RMA (one-sided)
  - MPI IO
  - Not Extended Collectives
  - Not Dynamic Process Management
  - Not Generalized Requests

Cray X1 MPI Implementation

Taking Advantage of the Hardware

- Single System Image Context
- Distributed Memory Architecture
  - Symmetric memory allocations
  - Addressing is simplified
  - Intra-node access via RTT

Cray X1 MPI Implementation

Hardware Advantages (cont.)

- SSP vs. MSP mode
  - Compute intensive or Bandwidth intensive
  - Application characteristics differ so try both
  - One node or Multi-node applications
- Scaling
  - High speed interconnect
- 32 and 64 Bit libraries

Cray X1 MPI Implementation

- New Algorithms Implemented
  - Collectives
    - Gather, scatter, reduce, bcast, barrier
  - Point-to-point
    - Send/receive, type, test, (un)pack
  - Groups, context, communicators
    - Keyval, attributes, intercomms, groups

Cray X1 MPI Implementation

MPI_Bcast

- Root allocates a buffer and stores data
- Root sends buffer address, count and type to non-root processes
- Root spin-waits for non-root processes to pick up data

Cray X1 MPI Implementation

The Barrier Algorithm

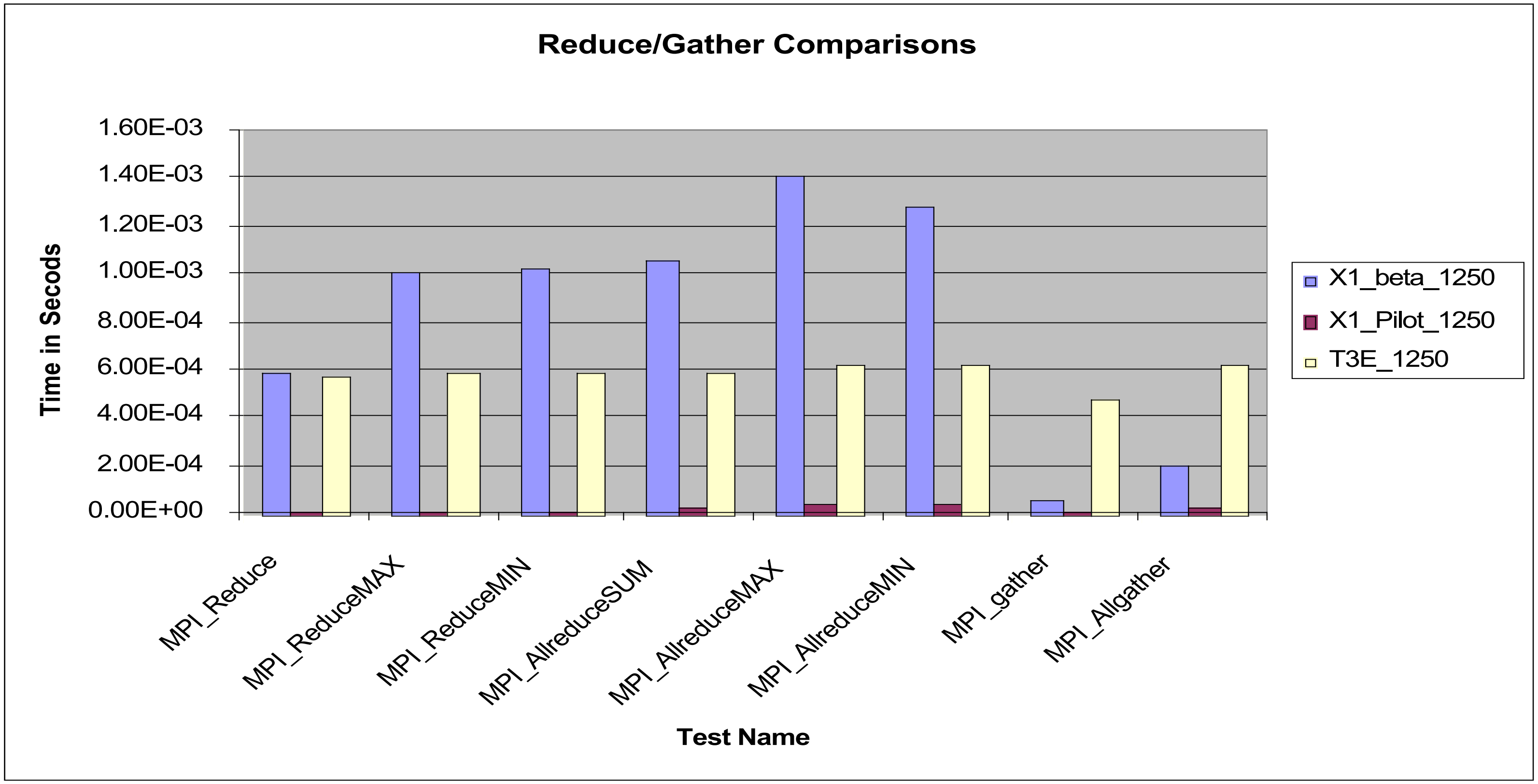
- Determine level, comm, rank within group
- Decrement group count
- Wait for group count to clear
- Last process through signals continue
- Done

Cray X1 MPI Implementation

MPI_Gather

- Root process determines data segments
- Root process transmits address, count and datatype to non-root processes
- Root process spin-waits while non-root processes push their data to the root

Reduce/Gather Performance Graph



Cray X1 MPI Implementation

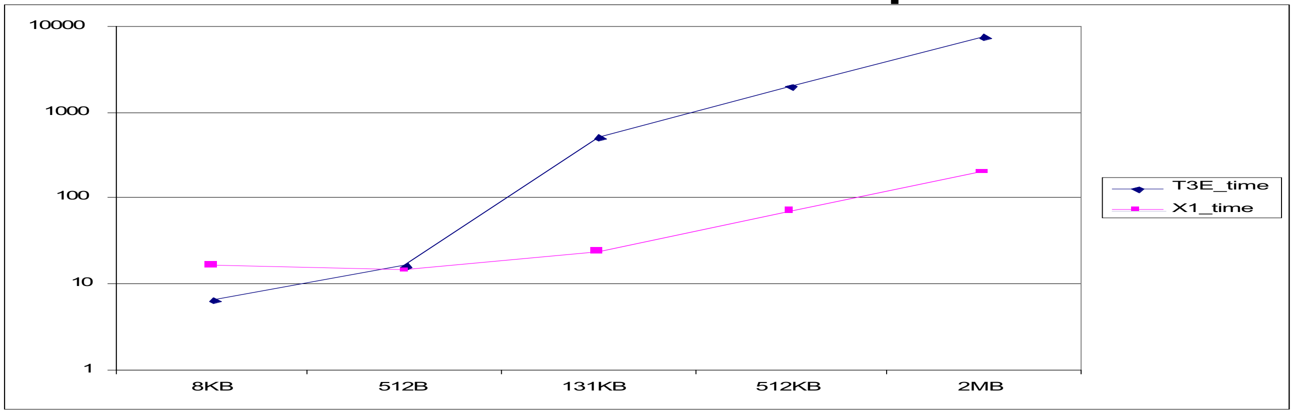
Send/Receive Algorithm

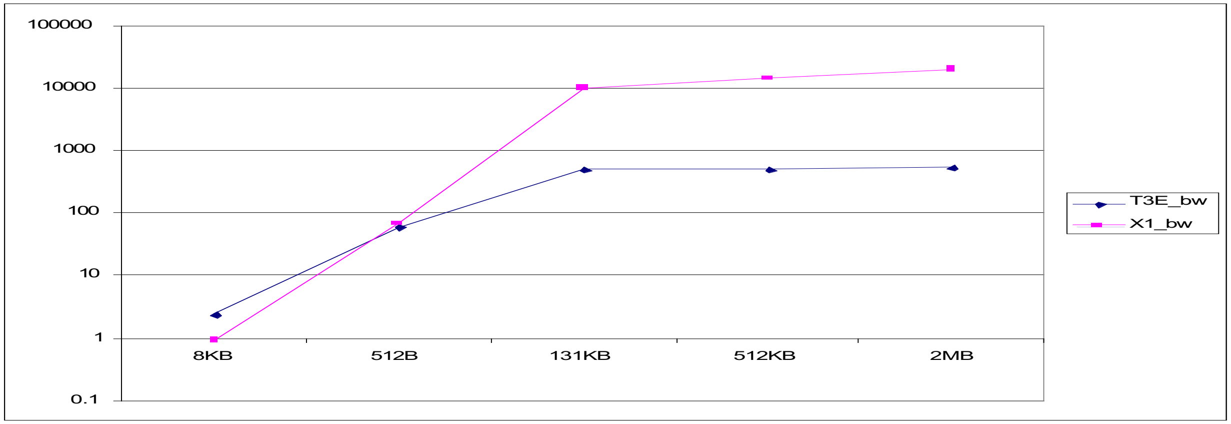
- MPI_Send
  - Get a packet from destination process
  - Put data in packet or address of data
  - Link packet on the receiver's incoming queue
- MPI_Receive
  - Scan incoming queue for matching tags
  - Pull data

Cray X1 MPI Implementation

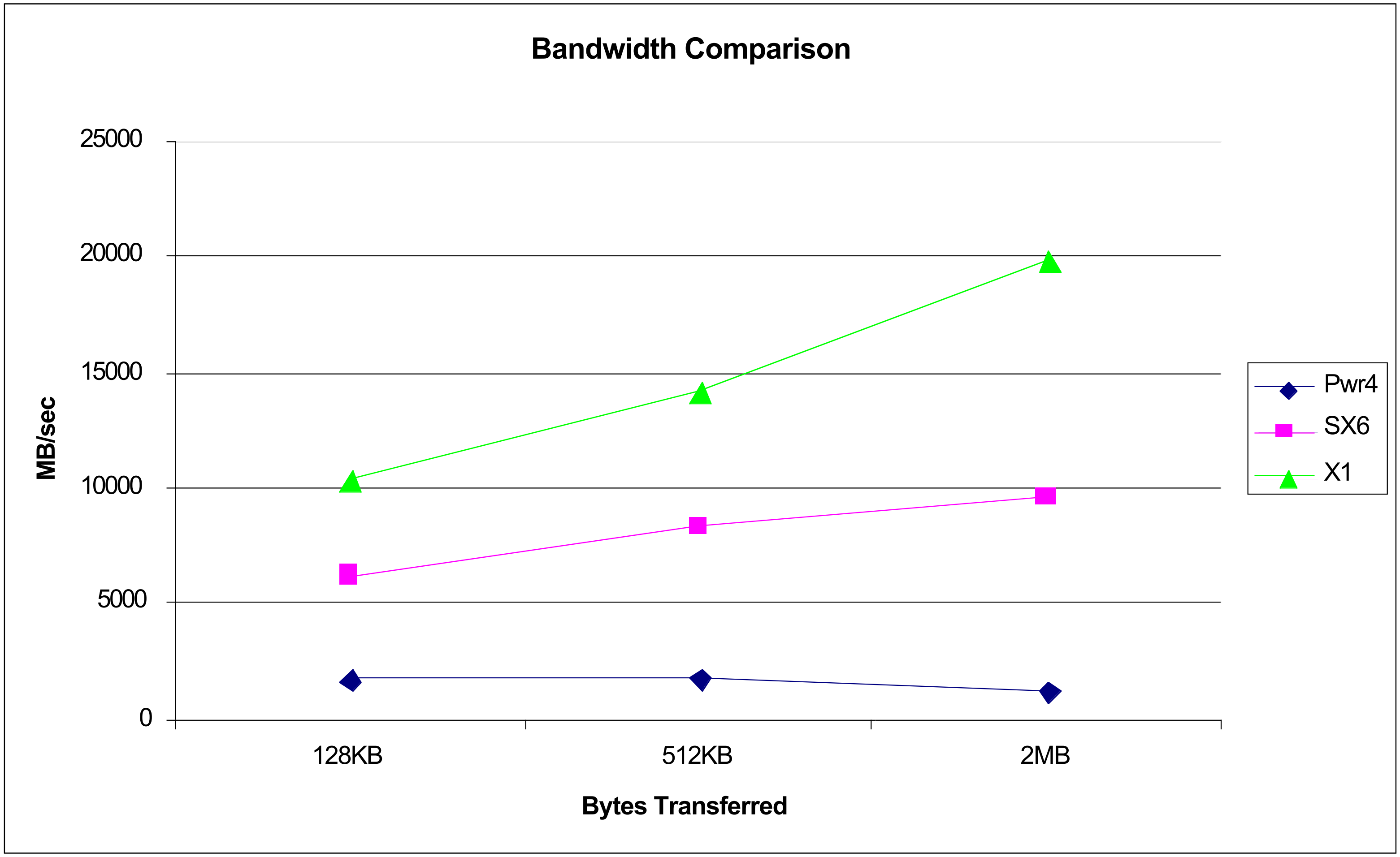
The 45th CUG Conference
CRAY USER GROUP
FLIGHT TO INSIGHT
OSC

Send/Receive Performance Graph





Cray X1 MPI Implementation



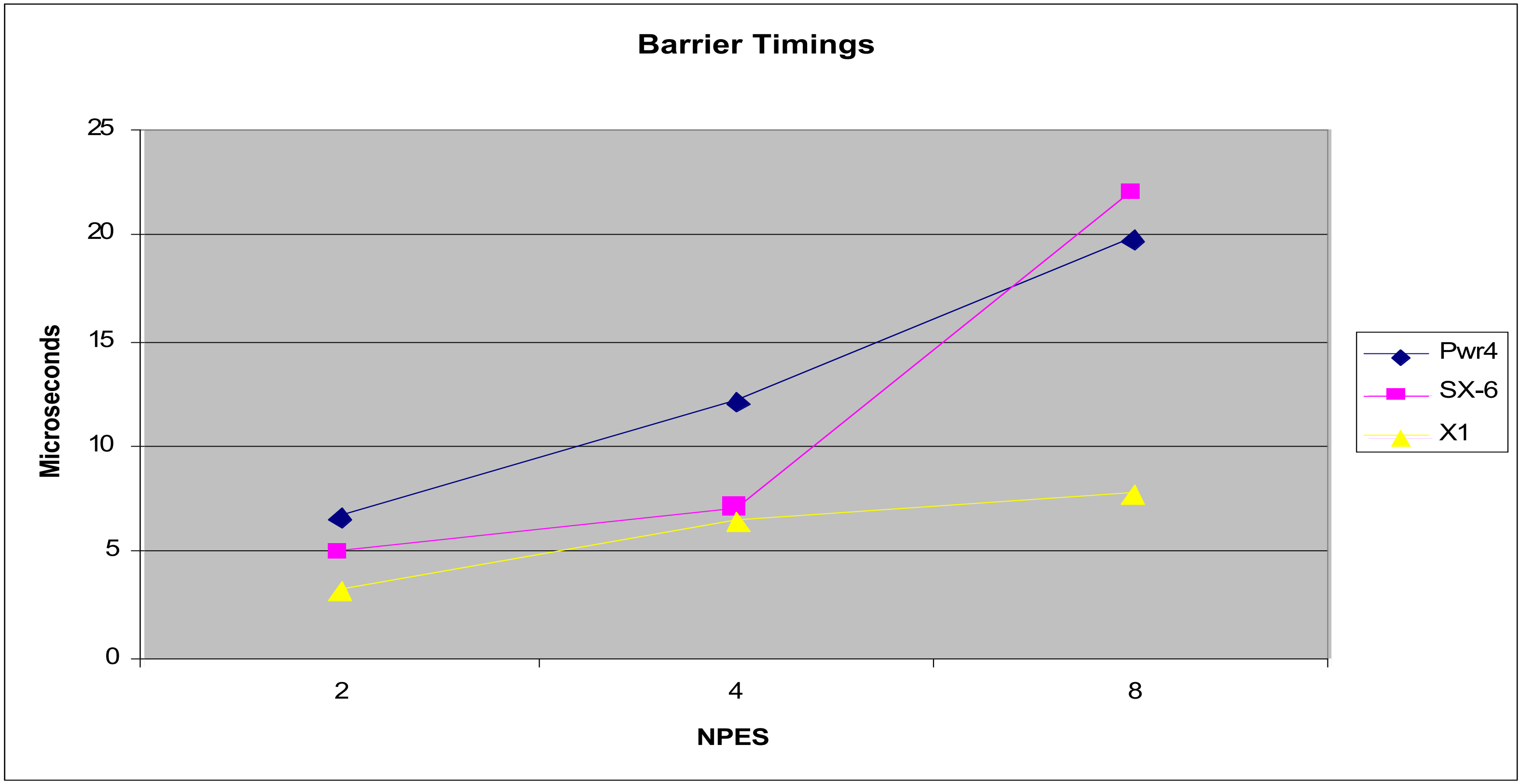
Cray X1 MPI Implementation

MPI_Barrier

- Uses a four-way tree
  - Each level barriers up to four processes
  - Each level barrier on one word
  - The depth of the tree is log(base4)
  - Uses atomic memory operation (fadd)

Cray X1 MPI Implementation

Barrier Performance Graph



Cray X1 MPI Implementation

Coding and Algorithmic Suggestions

- Take advantage of our high bandwidth
- Take advantage of our vector registers
- Review your coding techniques
- Review the problem you’re trying to solve

Questions?