

Execution Environments for Parallel Applications

Master CANS 2007/2008

**Departament d'Arquitectura de Computadors
Universitat Politècnica de Catalunya**

Outline

- ❑ **Supercomputers**
- ❑ **OS abstractions**
- ❑ **Tools for performance analysis**
- ❑ **Extended OS interfaces**
 - ❑ IRIX sysmp
 - ❑ IRIX memory placement
- ❑ **Example: the CPU Manager**
- ❑ **Linux interfaces**
- ❑ **Conclusions**



Tools for Performance Analysis

Blue Gene/L Performance Tools

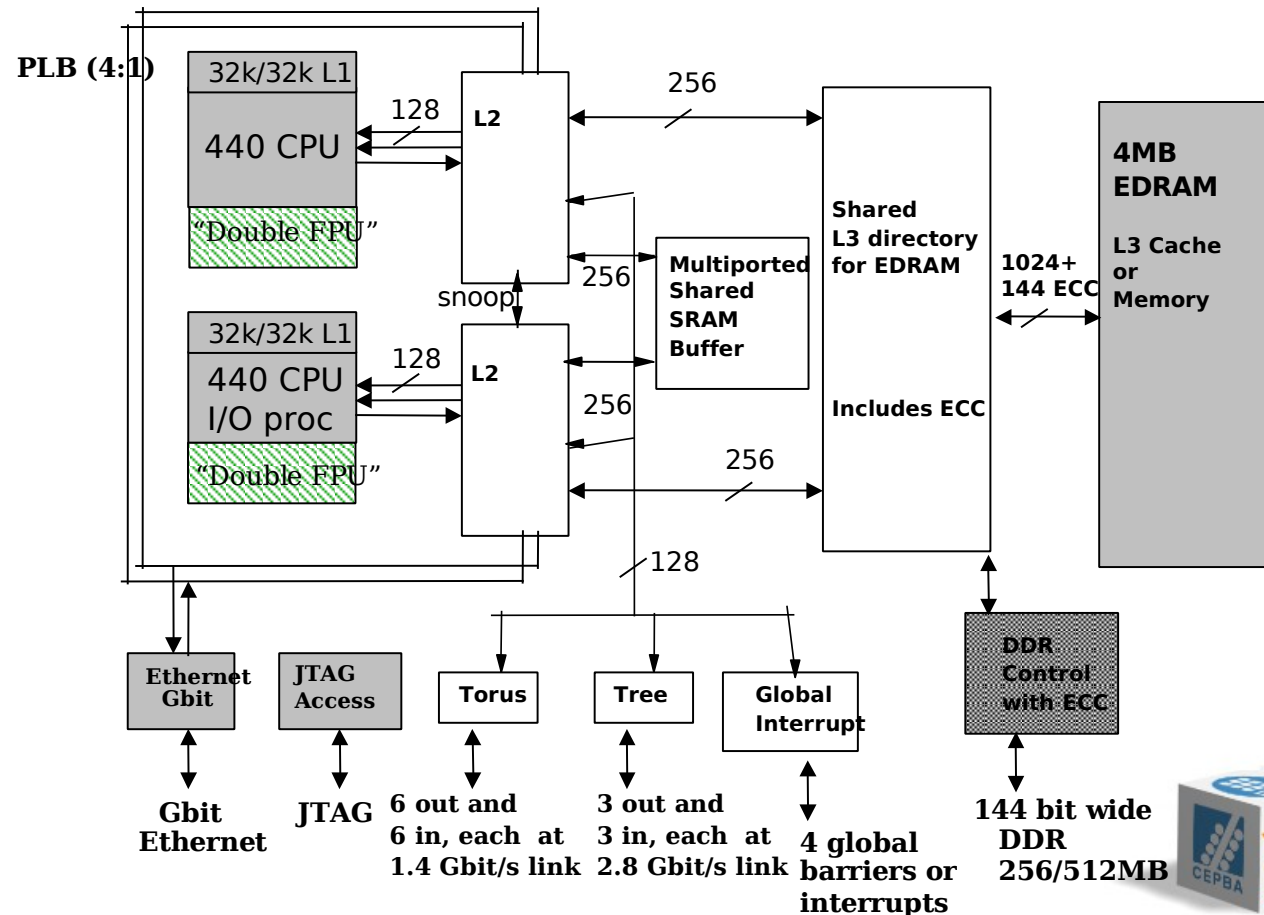
Hardware Performance Counters

- ❑ **Capable of tracking up to 50 counters at the same time**
- ❑ **Counting**
 - ❑ Events
 - ❑ Cycles an event is active
- ❑ **Classification of counters**
 - ❑ Double hummer
 - ❑ L3 cache
 - ❑ Prefetch unit
 - ❑ Tree network
 - ❑ Torus network



Hardware Performance Counters

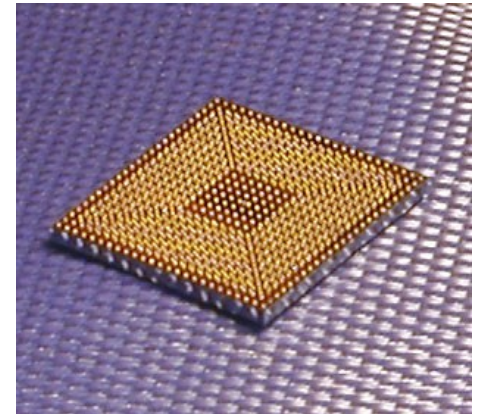
- For each component of the BG/L chip, except the ppc440 macros



Hardware Performance Counters

❑ Counters on the torus network

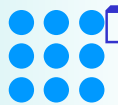
- ❑ 32-byte chunks sent (or passing-through)
- ❑ Packets sent (or passing-through)
- ❑ Acknowledgments sent
- ❑ Link available, no tokens
 - ❑ On the different channels
- ❑ Token with ack sent
- ❑ ... and more



❑ For each dimension and direction

- ❑ total of 6 in a 3D torus/mesh

❑ Similar counters for the tree network



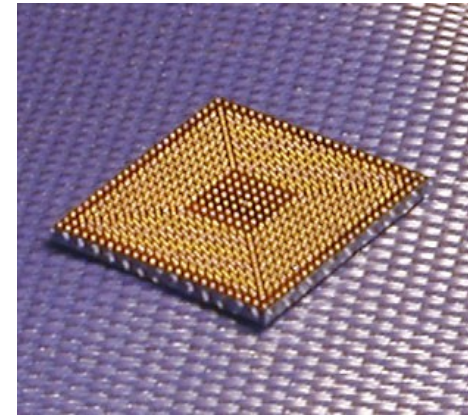
UPC



Hardware Performance Counters

❑ Double Hummer

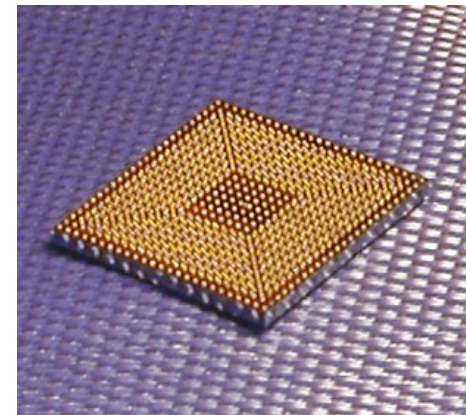
- ❑ ADD/SUBSTRACT
- ❑ MULT/DIV
- ❑ MULTIPLYADD
- ❑ DOUBLE_LOAD, DOUBLE_STORE
- ❑ QUAD_LOAD, QUAD_STORE
- ❑ ...



Hardware Performance Counters

❑ L3 cache & prefetch unit

- ❑ HIT
- ❑ MISS_DATA_COMING
- ❑ MISS_DATA_REQUEST
- ❑ EVICT_LINE
- ❑ PREFETCH_REQUEST
- ❑ RD_BURST_1024B
- ❑ RD_MODIFY_WR_CYCLE
- ❑ ...



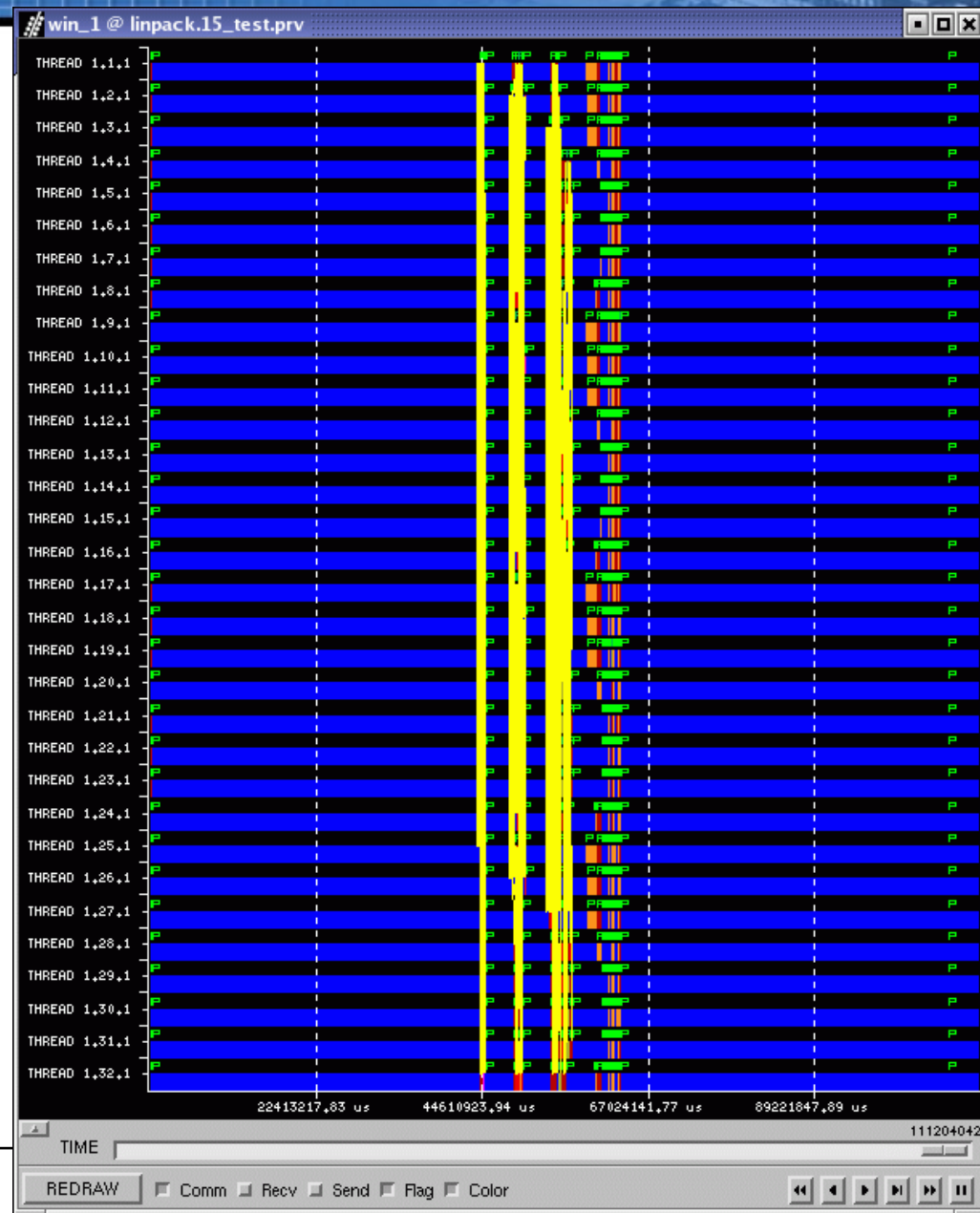
Collecting traces on BG/L

- ❑ **MPITrace tool is linked with the application**
 - ❑ Based on the PMPI interface
- ❑ **PAPI for BG/L is used to collect hardware performance counters**
- ❑ **A single file per node is generated during the execution**
- ❑ **All intermediate files are merged into a single Paraver trace**
 - ❑ example.prv contains the Paraver trace
 - ❑ example.pcf contains textual information
- ❑ **Visualization with Paraver**



Collecting traces on BG/L

- ❑ **Linpack on 32 nodes**
 - ❑ 8x4 (N=40960)
- ❑ **Only the first 3 iterations**
- ❑ **Long start and end phases**
- ❑ **Different communication patterns**

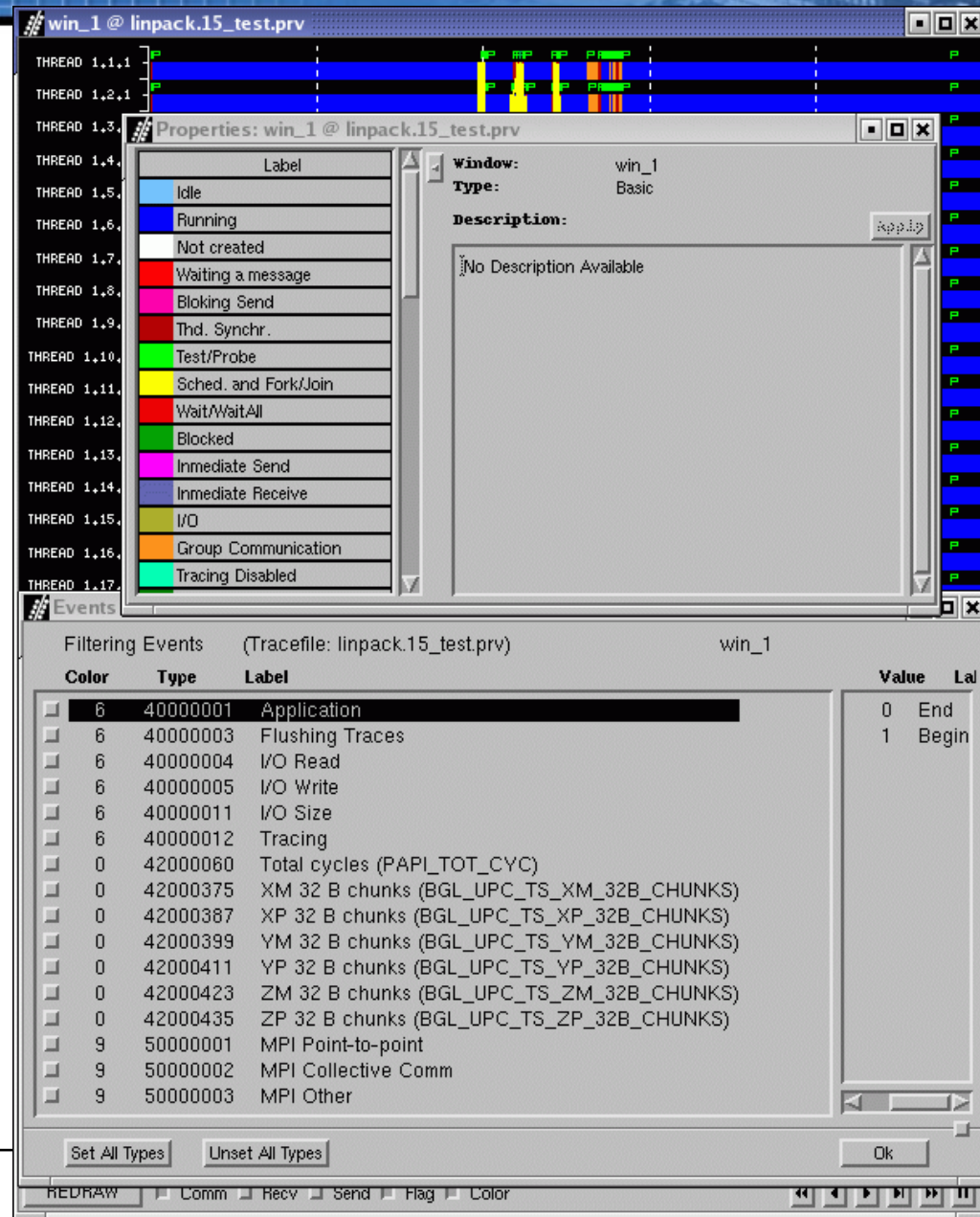


Collecting traces on BG/L

□ State information

□ Event information

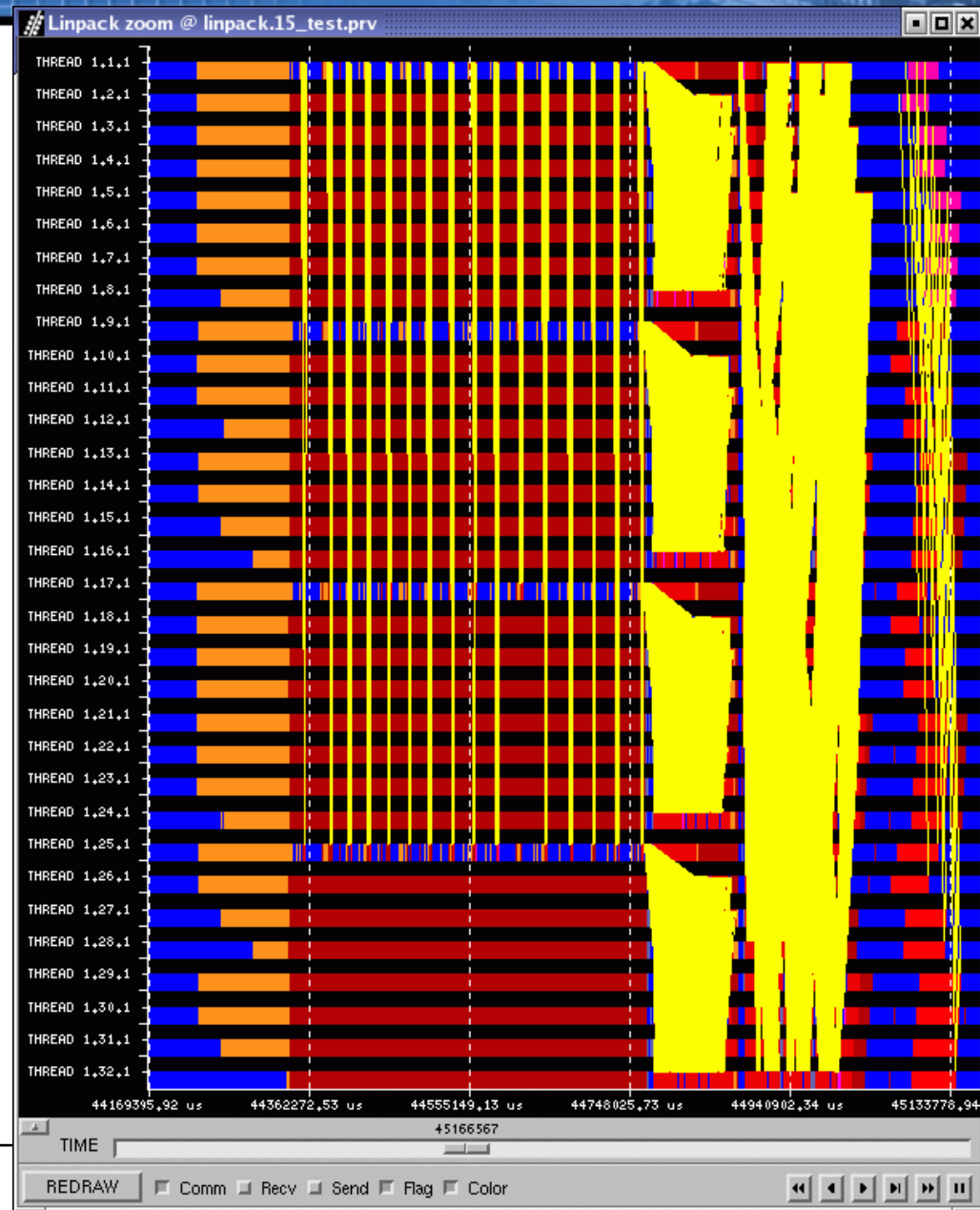
- Total cycles
- 32-byte chunks sent in each dimension
 - Including pass-through
- MPI
 - point-to-point comm
 - collective comm



Zooming communication regions

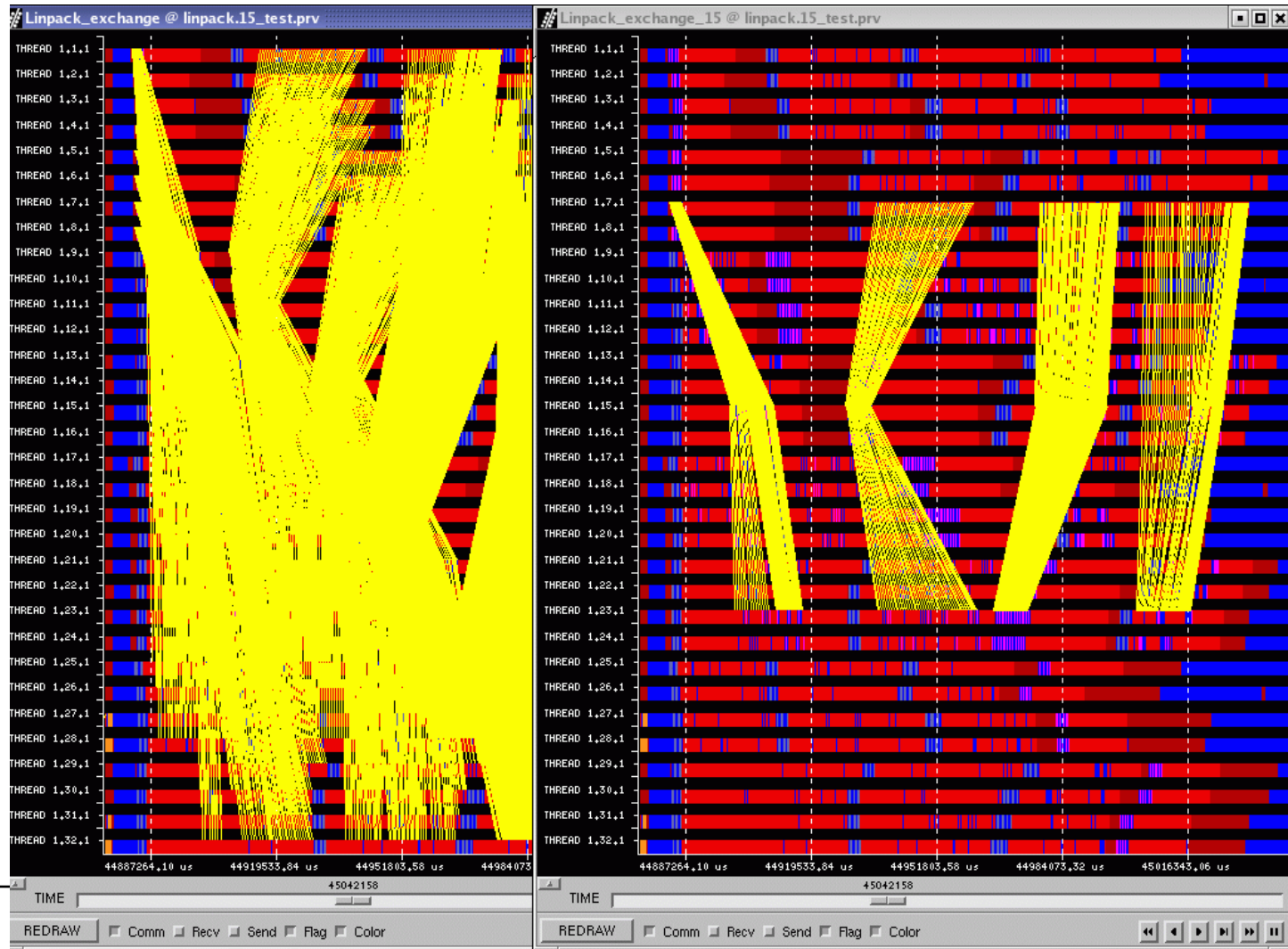
□ Single iteration

- Initial barrier
- Pivoting
- Group broadcast
- Column broadcast
- Final phase



Exchange phase

- Different nodes take different amount of time



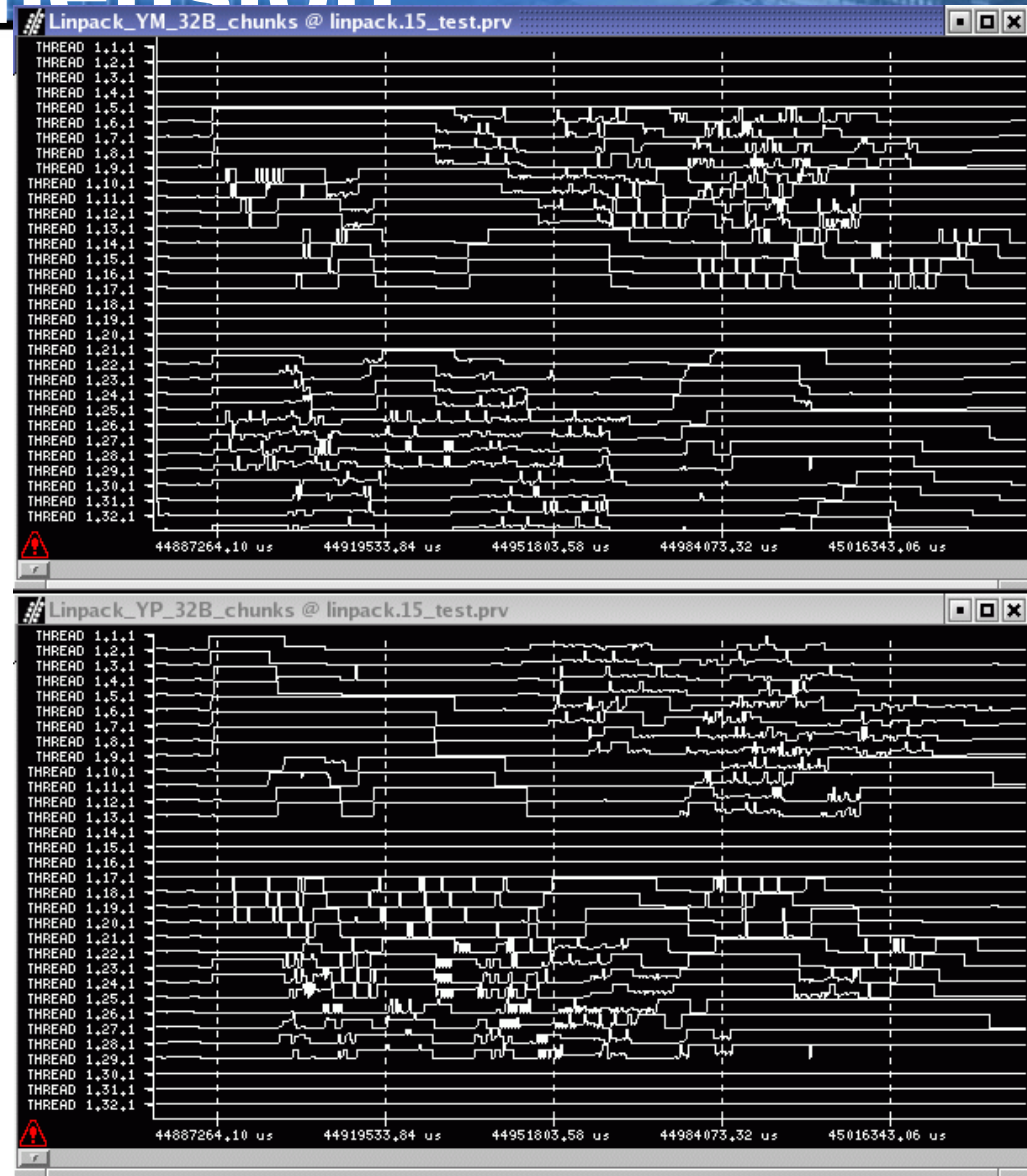
Communication on the X dimension

- ❑ Using the 32-byte chunks counter on XP and XM
- ❑ None!
- ❑ The XM & XP links are unused in this phase



Communication on the Y dimension

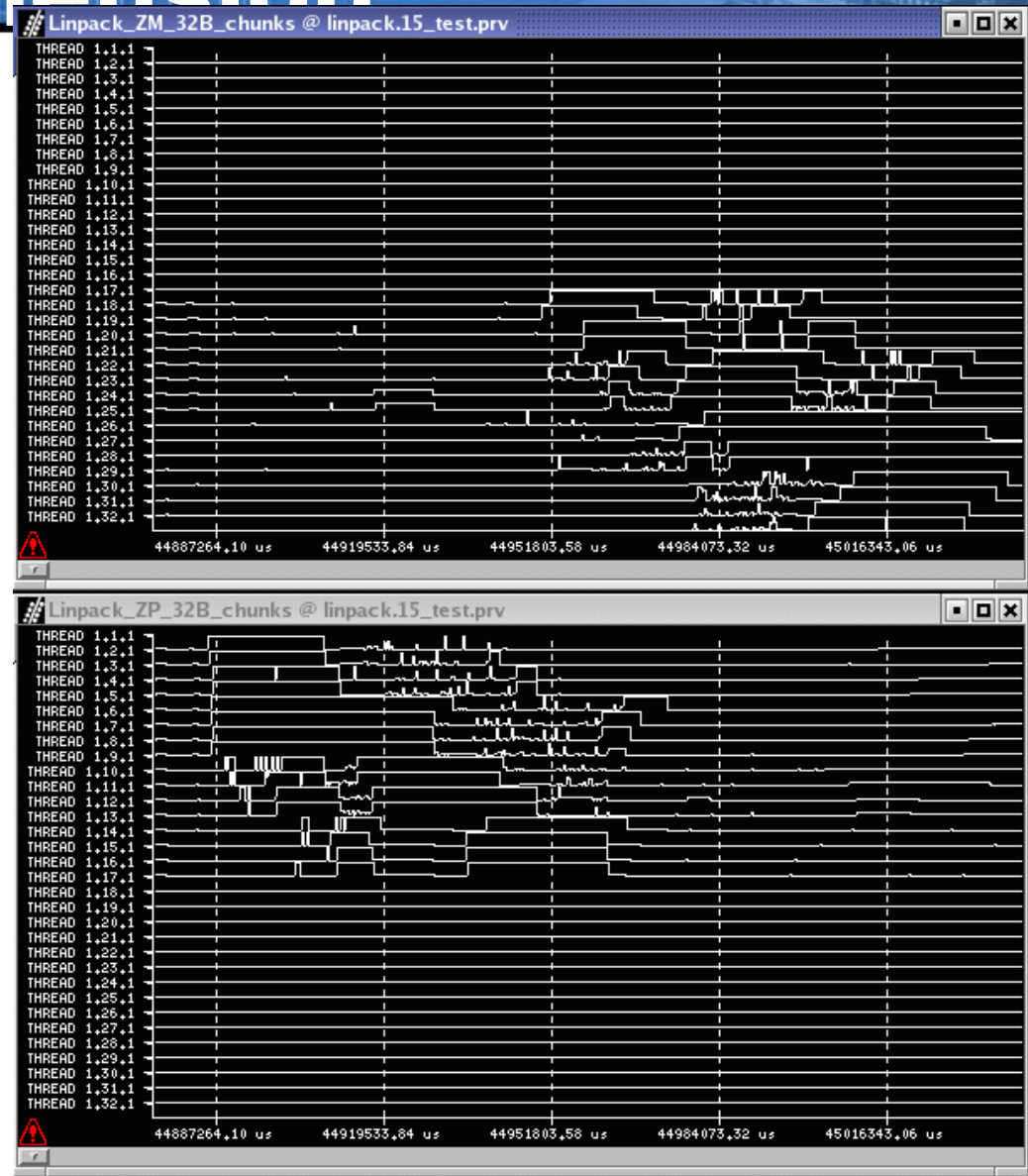
- ❑ Using the 32-byte chunks counter on YP and YM
- ❑ Heavy load



Communication on the Z dimension

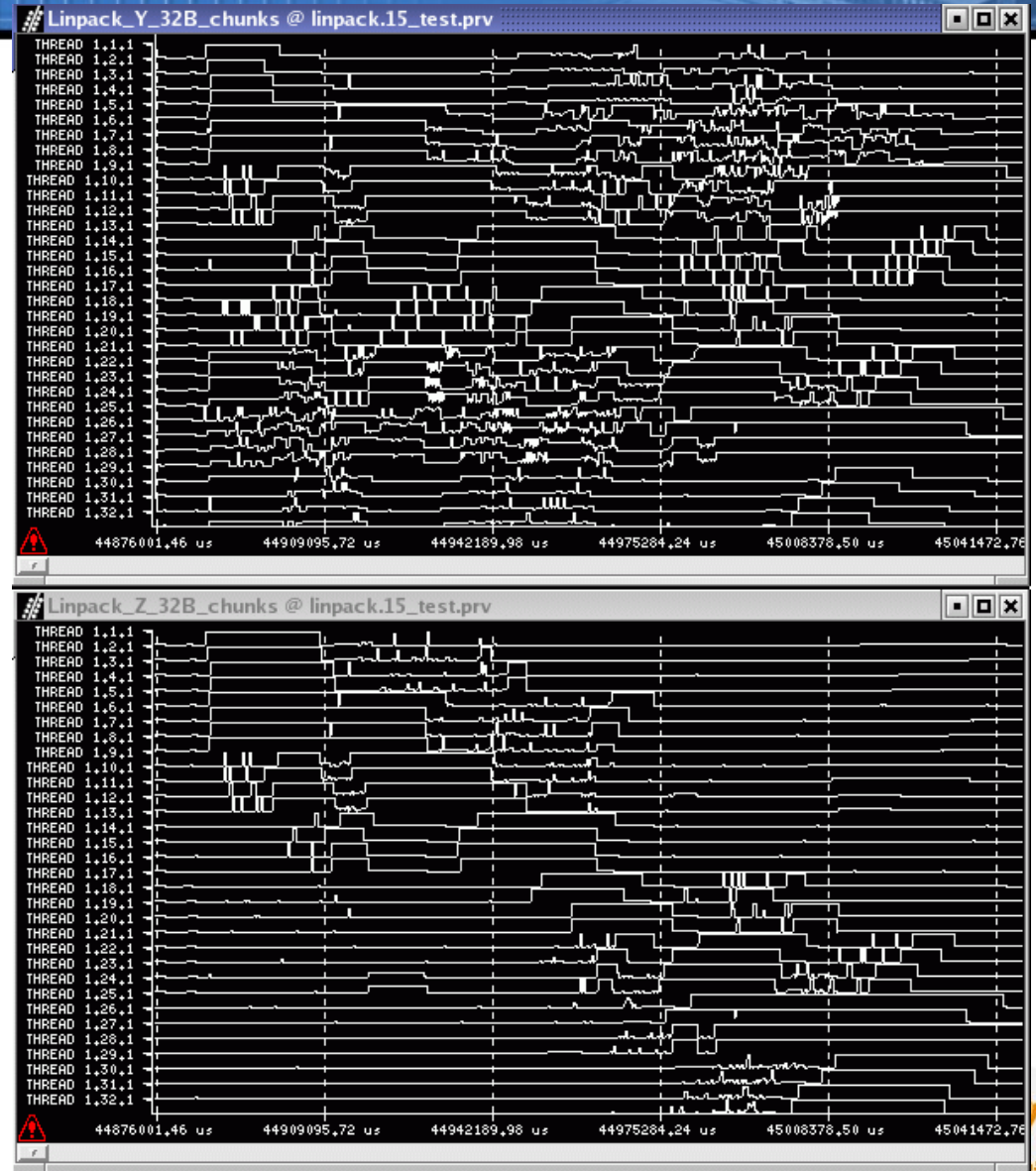
- Using the 32-byte chunks counter on ZP and ZM

- Unbalanced load



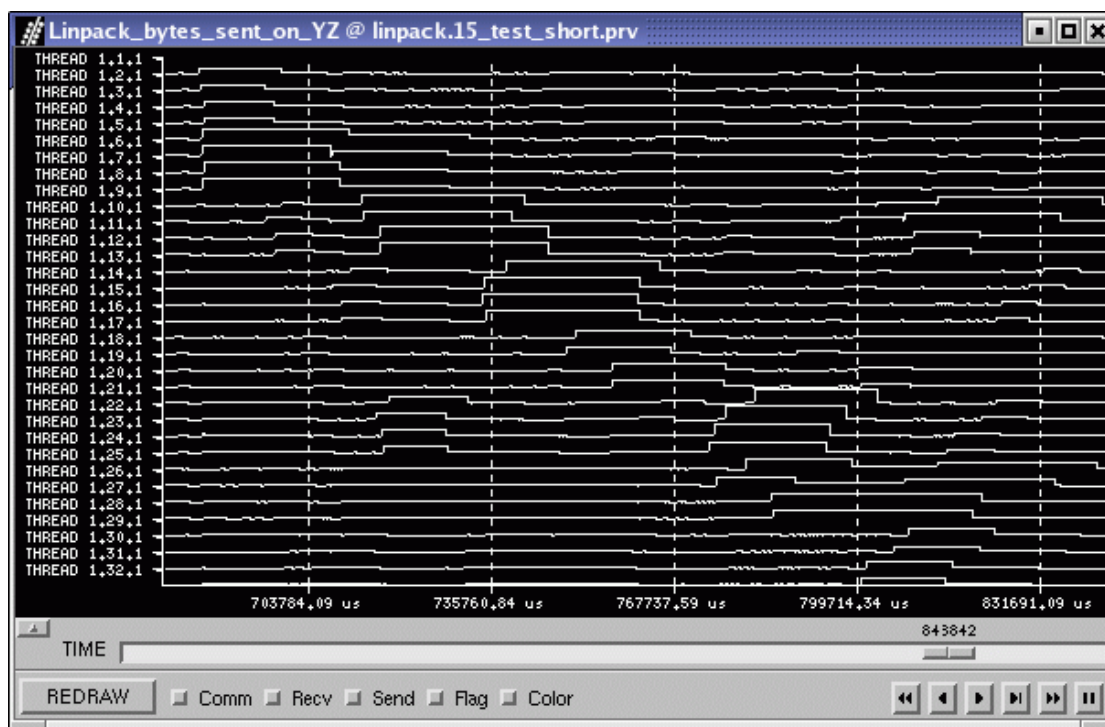
Adding

□ Y and Z dimensions



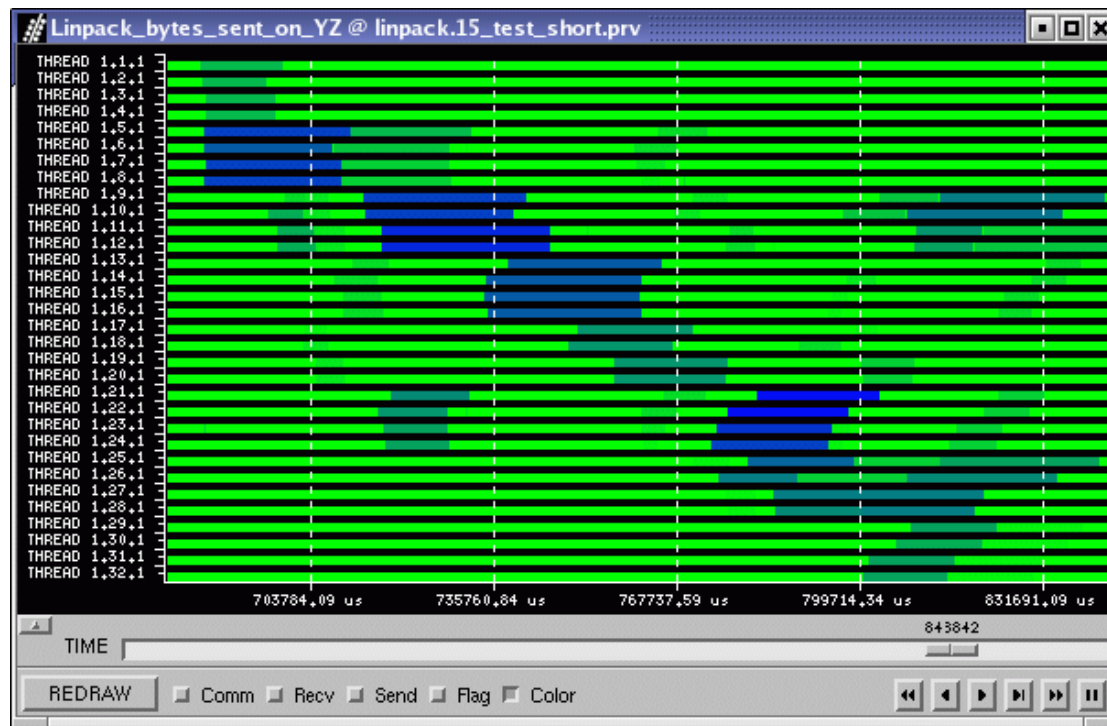
Adding (2)

□ Bytes send on YZ dimensions



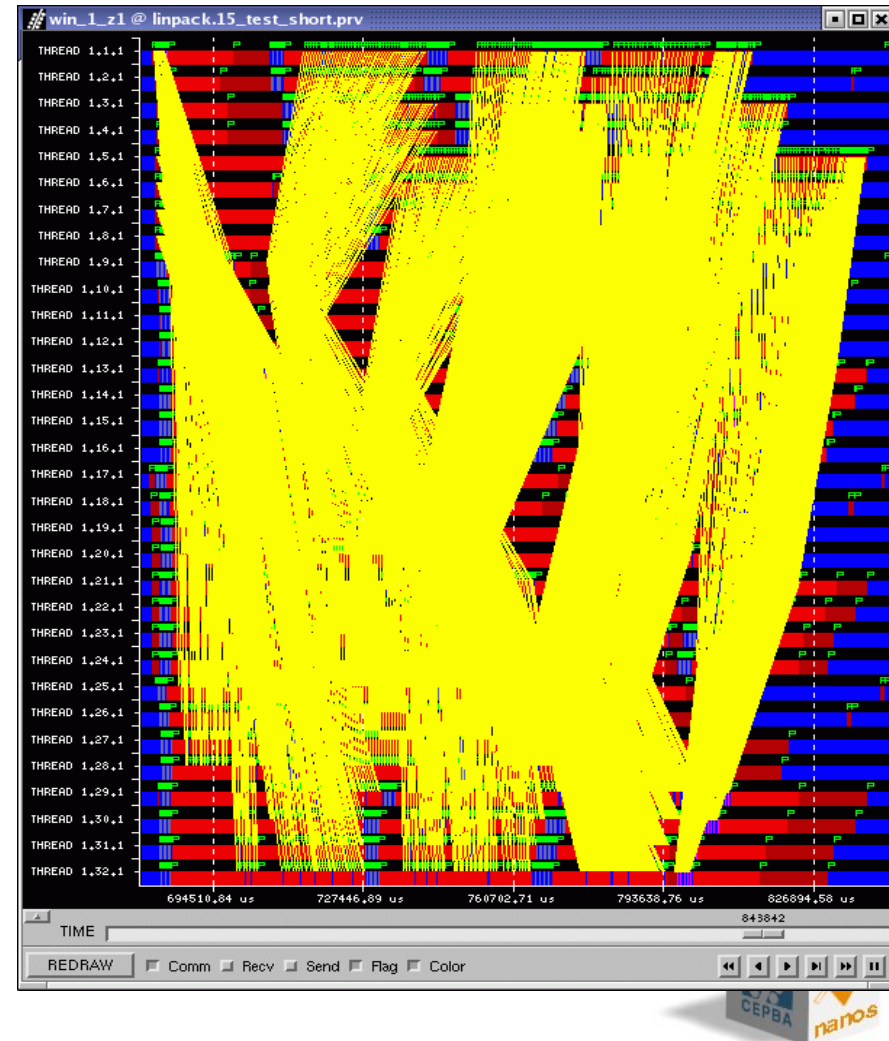
Gradient visualization

- Increases readability
- More intuitive representation



How to verify the results of the counters?

- ❑ **Hardware performance counters are not a priority for testing**
- ❑ **Can we find a different way of measuring the “bytes sent” in order to compare?**
 - ❑ Paraver can represent “bytes sent” per node from the information in the trace (without the need of hardware counters)

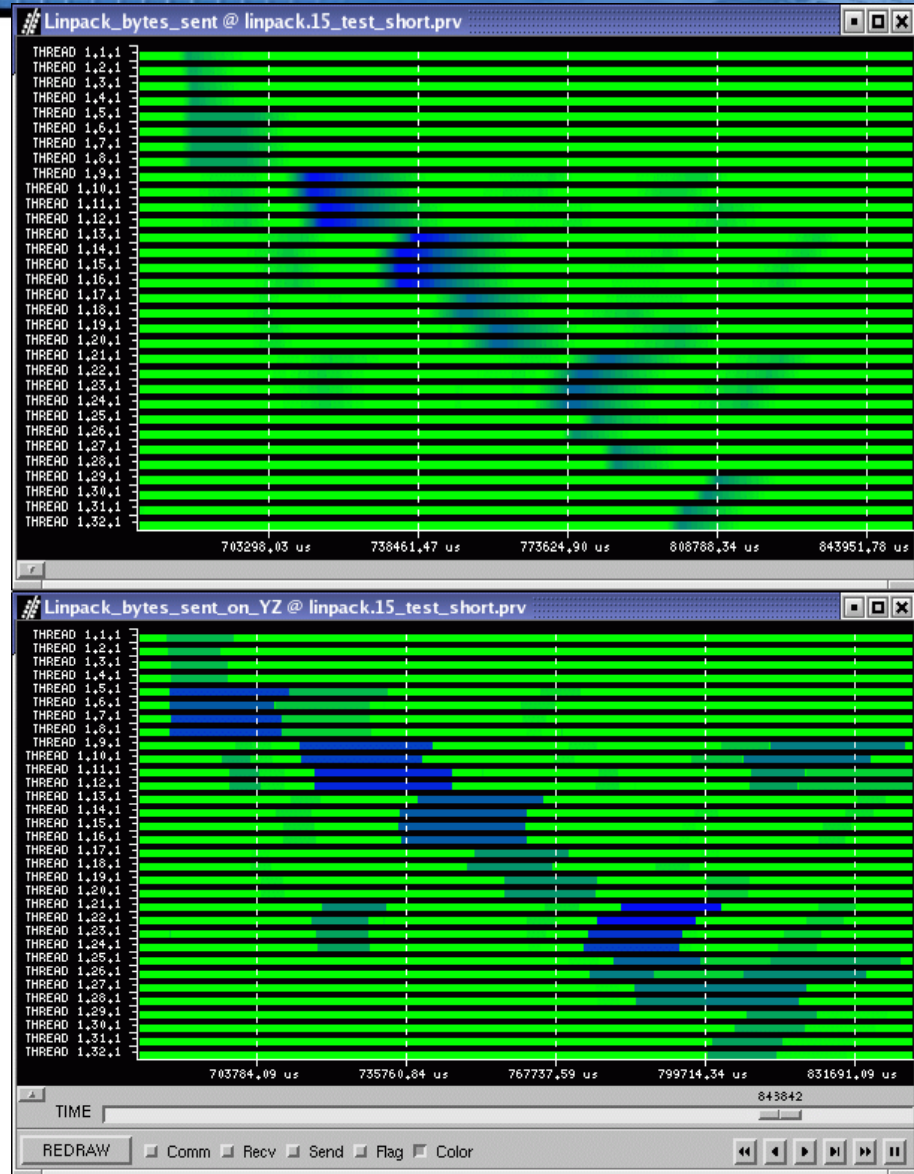


Verifying...

- ❑ Different?
- ❑ Different metrics?

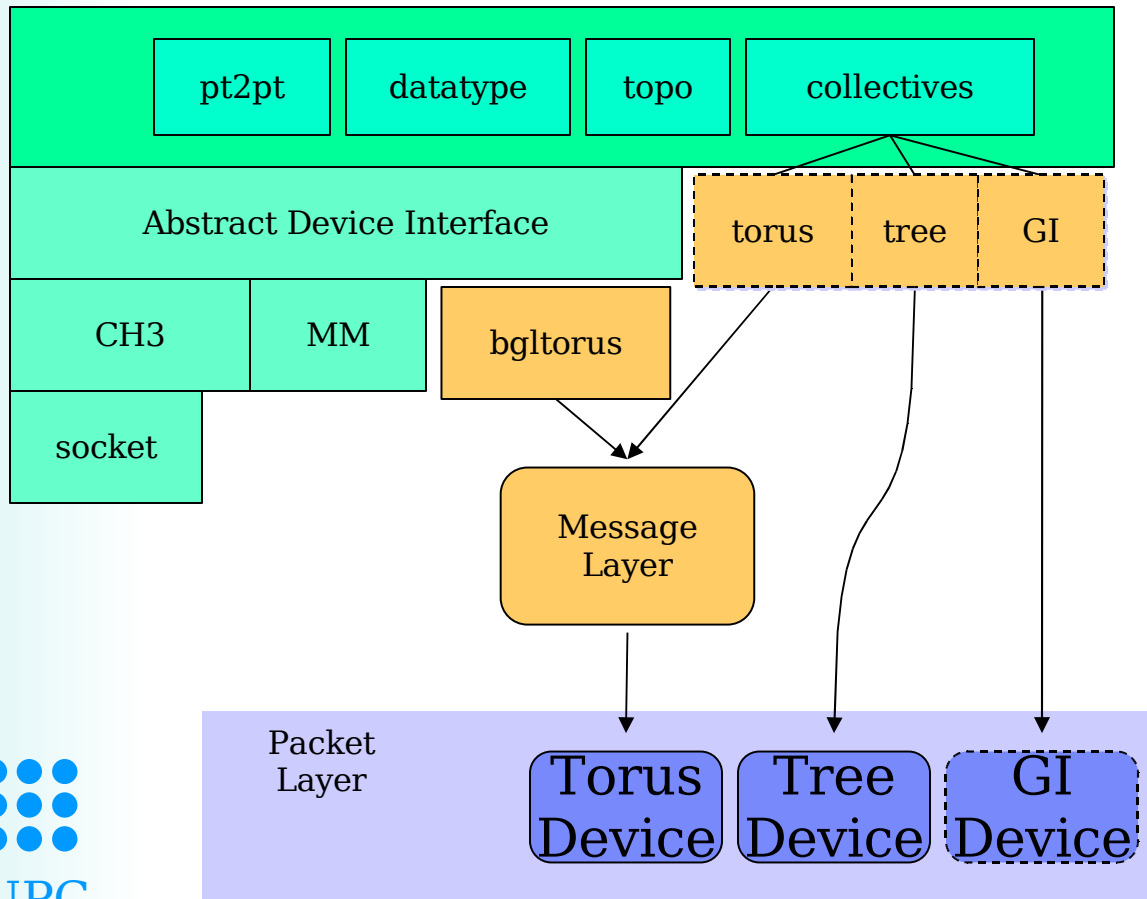
Based on the MPI communication

Based on the hardware counters

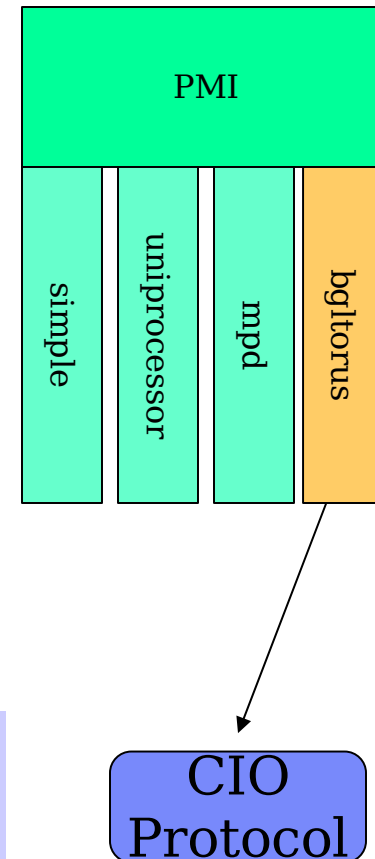


BG/L MPI Software Architecture

Message passing

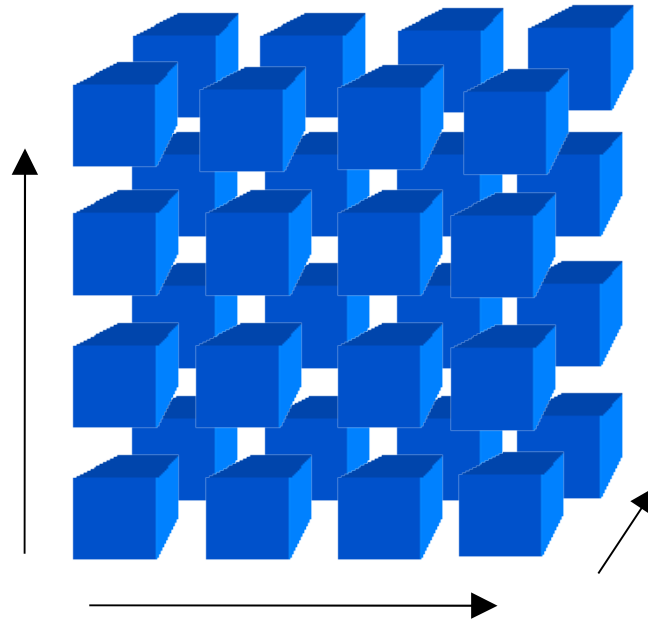


Process management



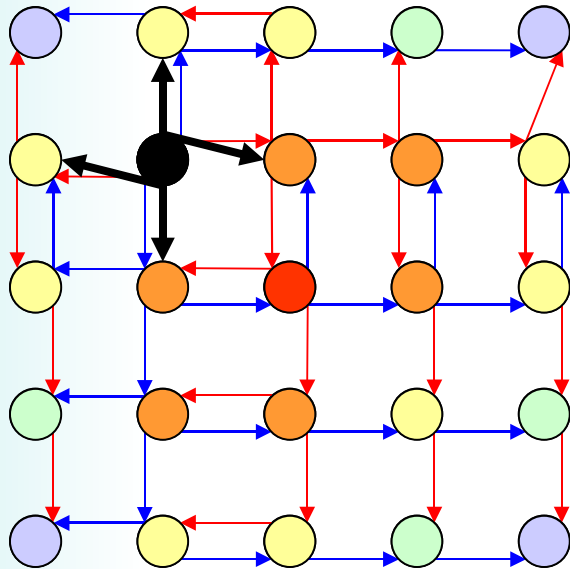
Mapping is important!

- **32 nodes are organized like 4x4x2**
 - 128, as 4x4x8



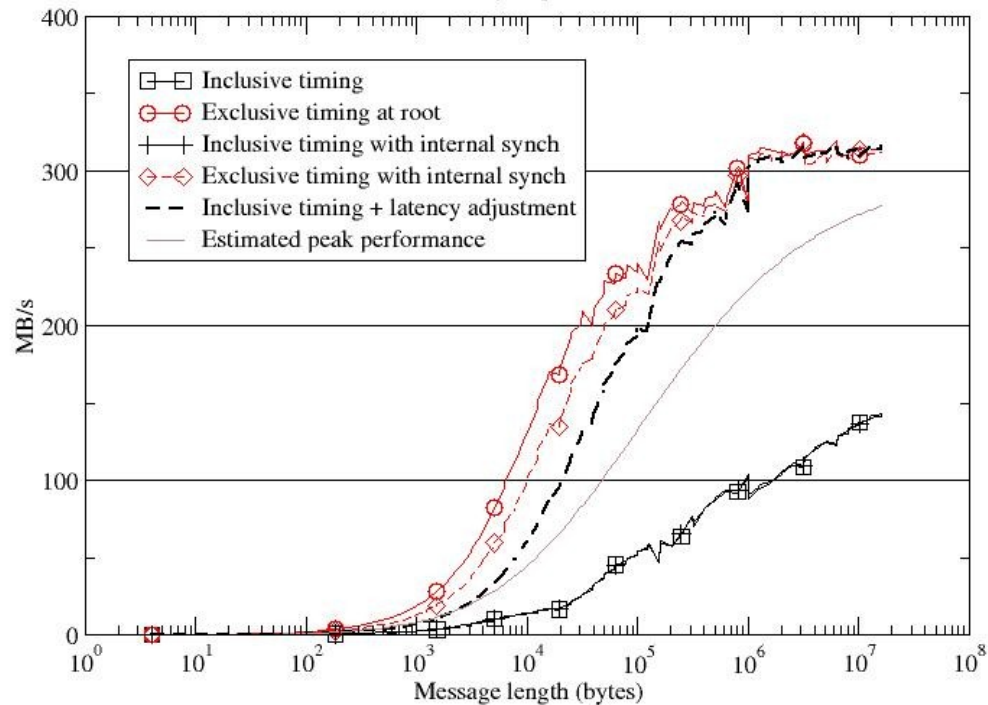
Broadcast on a mesh (torus)

Based on ideas from Vernon Austel, John Gunnels, Phil Heidelberger, Nils Smeds
Implemented & measured by Nils Smeds



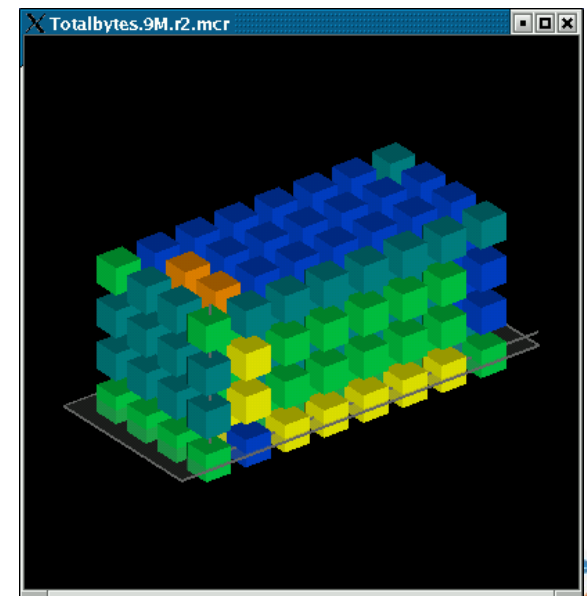
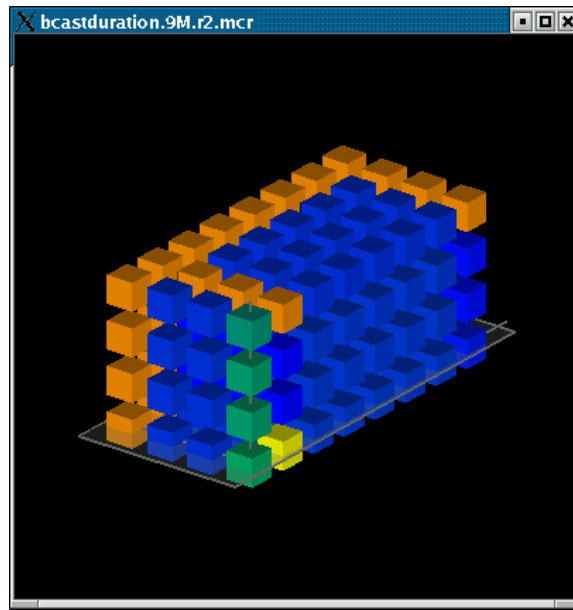
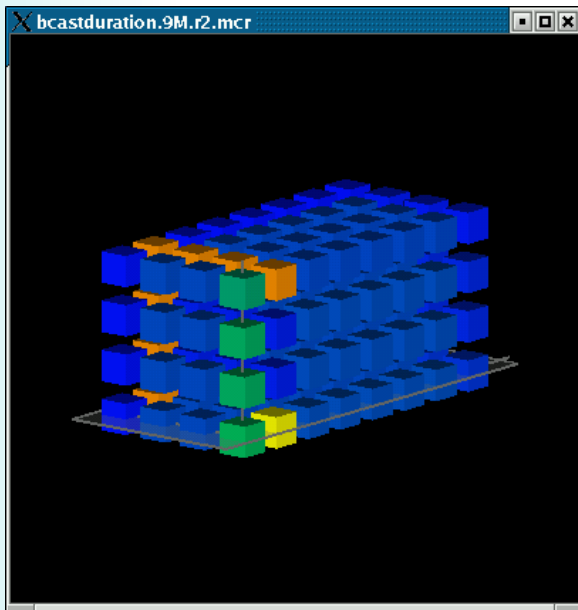
128-way broadcast timing

8x4x4 mesh topologi, root at [0,0,0]



3D Visualization

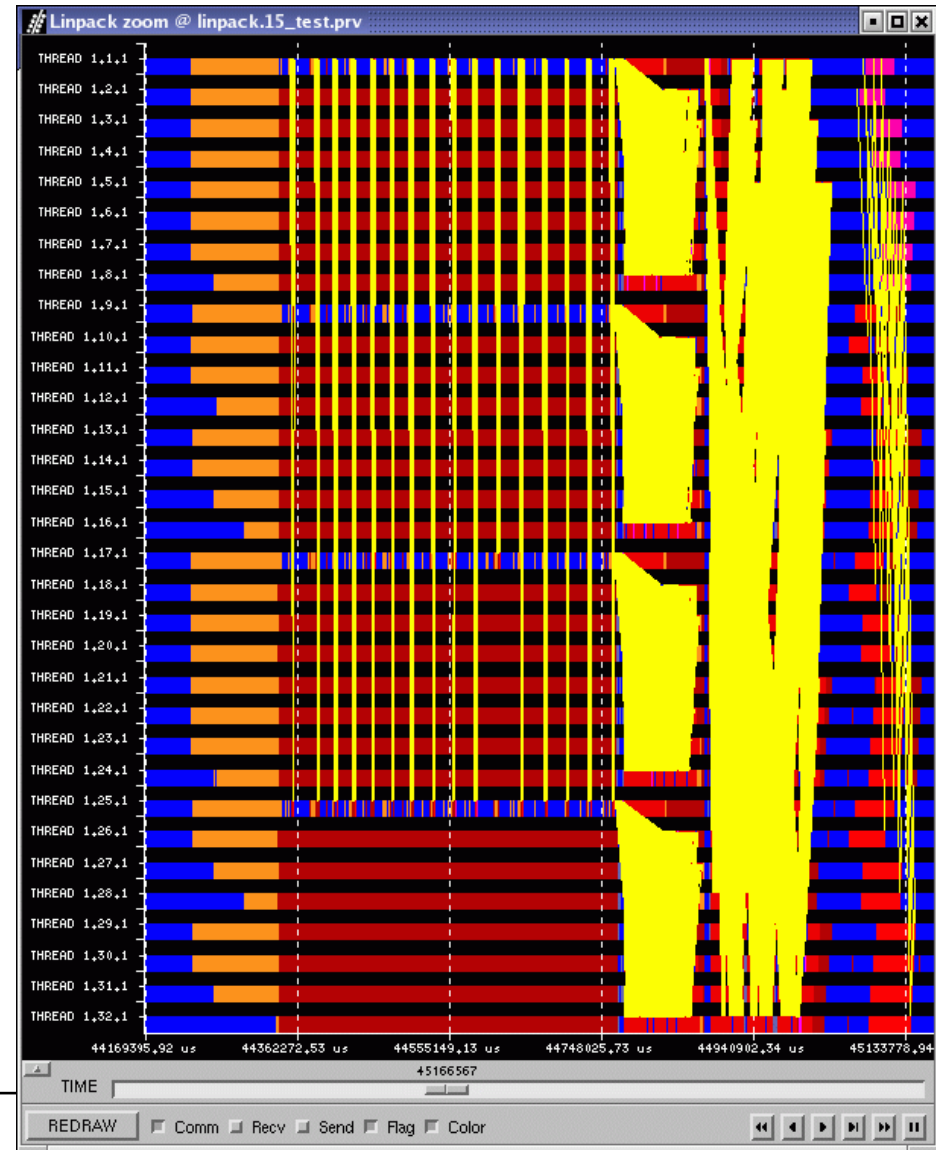
- **Physical 3D structure (128 nodes)**
 - Spatial localization of bottlenecks



Impact of MsgLayer Support

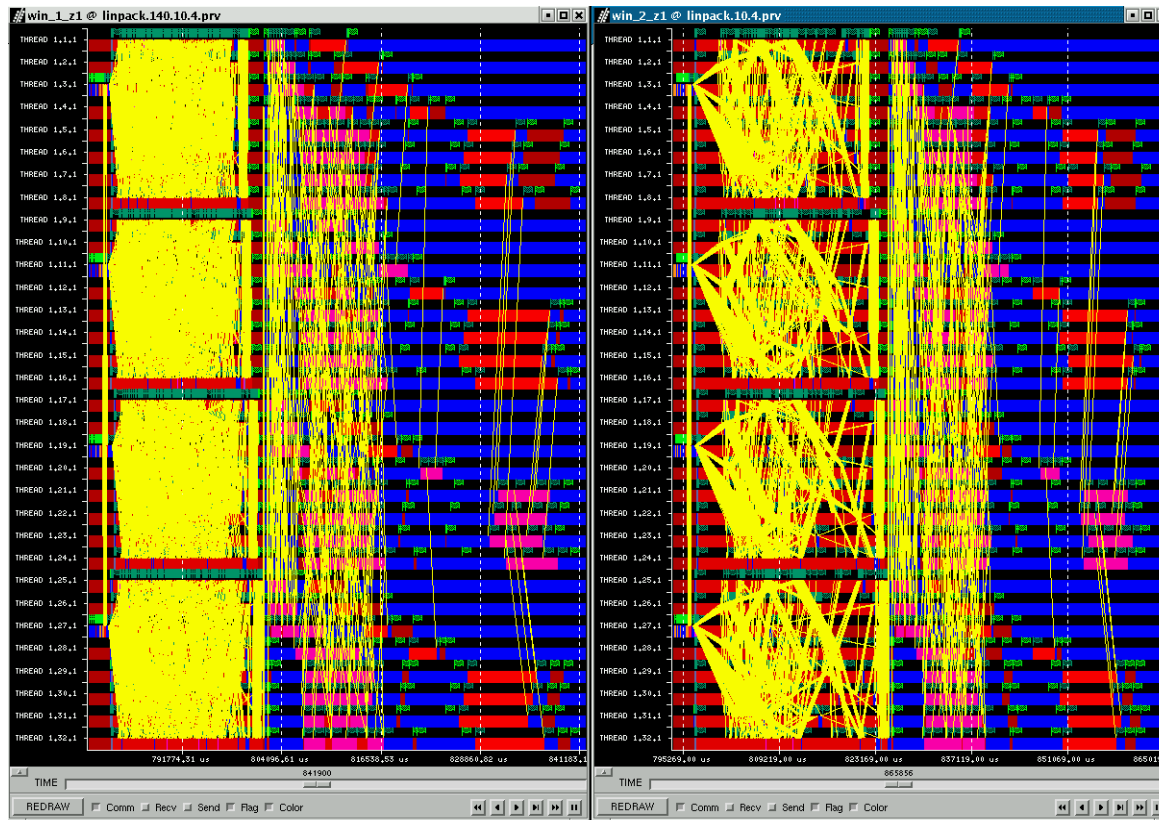
□ Linpack

- Initial barrier
- Pivoting
- Group broadcast
- Column broadcast



Impact of MsgLayer Support

- **Different implementations of broadcast**
 - Why are they showing so many differences?
 - Which one is better and when?



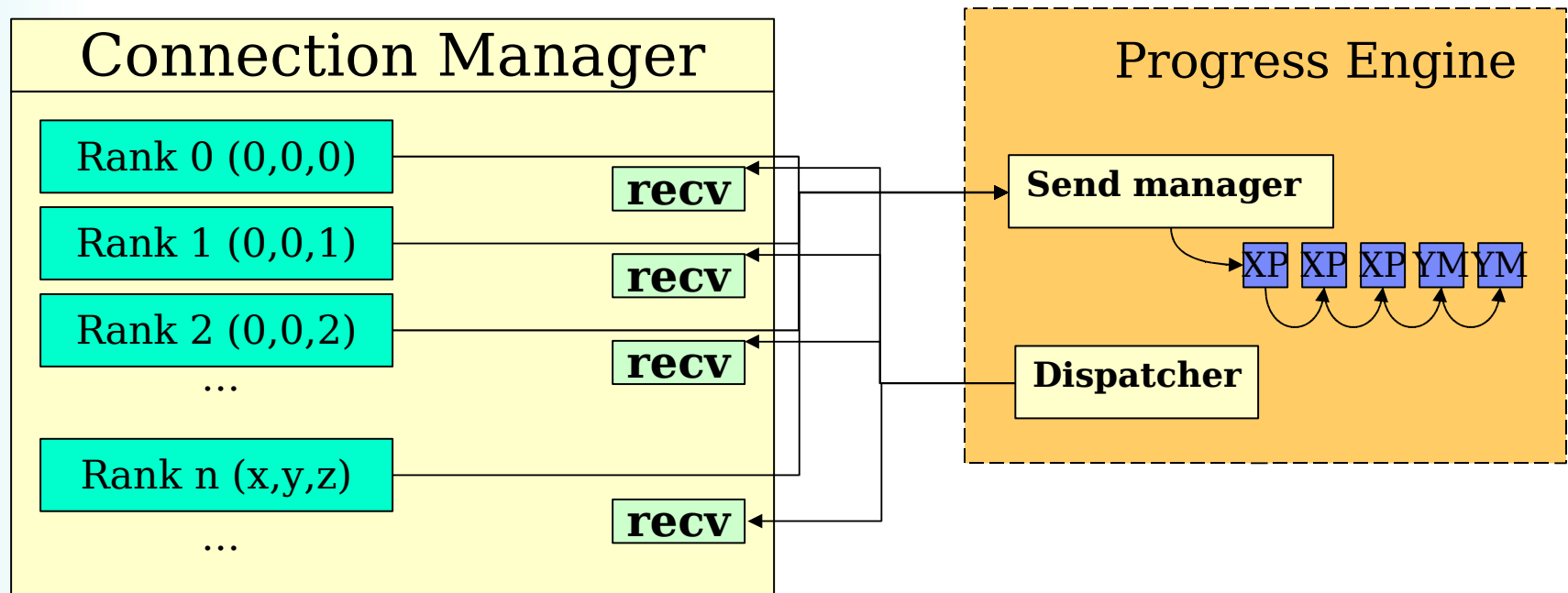
Impact of MsgLayer Support

□ Sending messages

- Overlapping messages
 - Several messages at the same time
 - to the same destination
 - using the same output links
 - Packets picked up from the available messages in a round-robin fashion
- One message at a time, FIFO mode, per direction
 - Up to 6 messages in parallel, using different links

Impact of MsgLayer Support

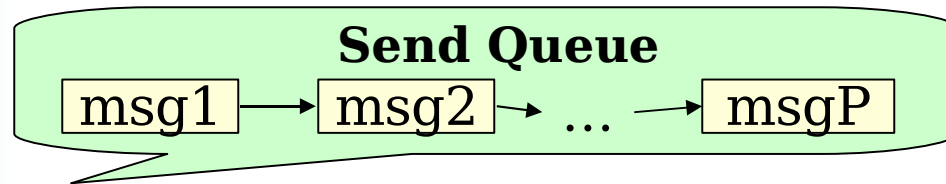
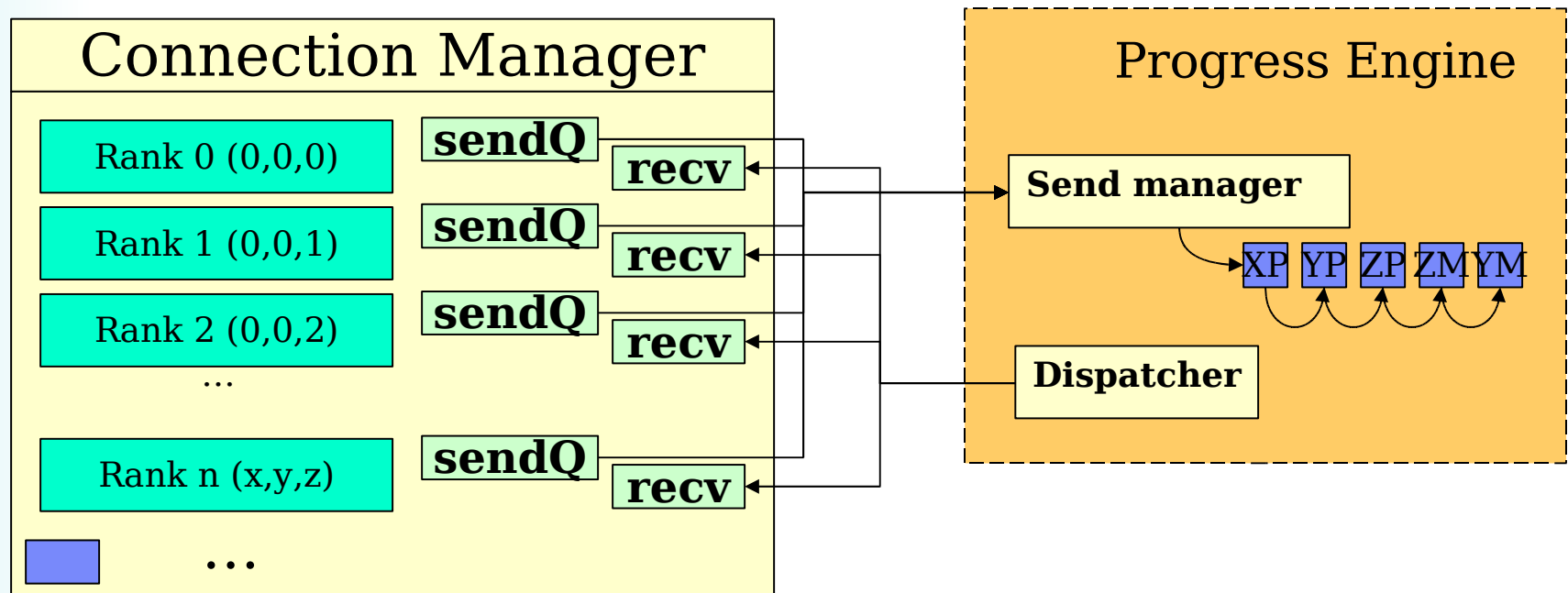
- **Sending messages**
 - Overlapping messages



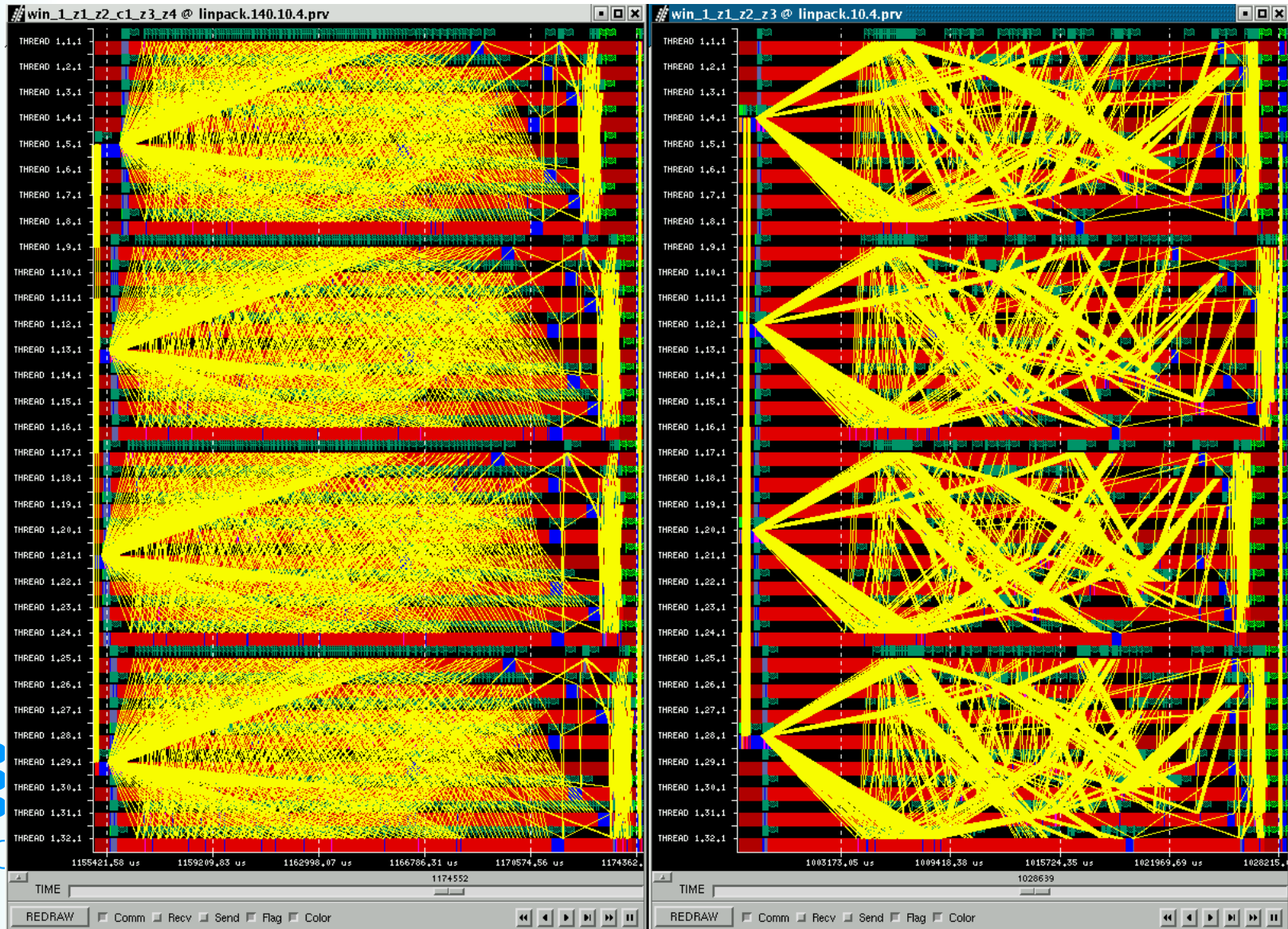
Impact of MsgLayer Support

□ Sending messages

- One message at a time in each direction

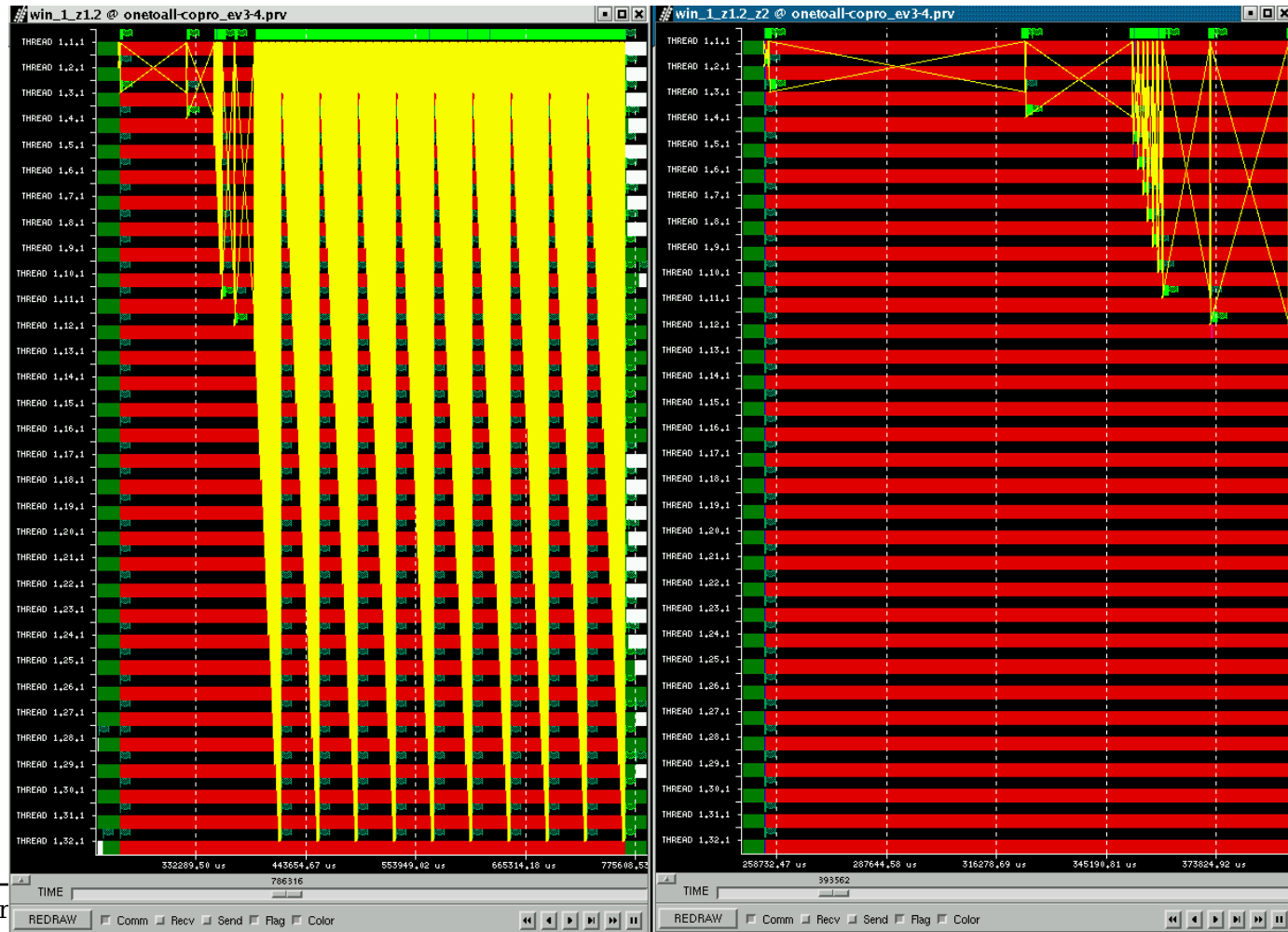


Impact of MsgLayer Support



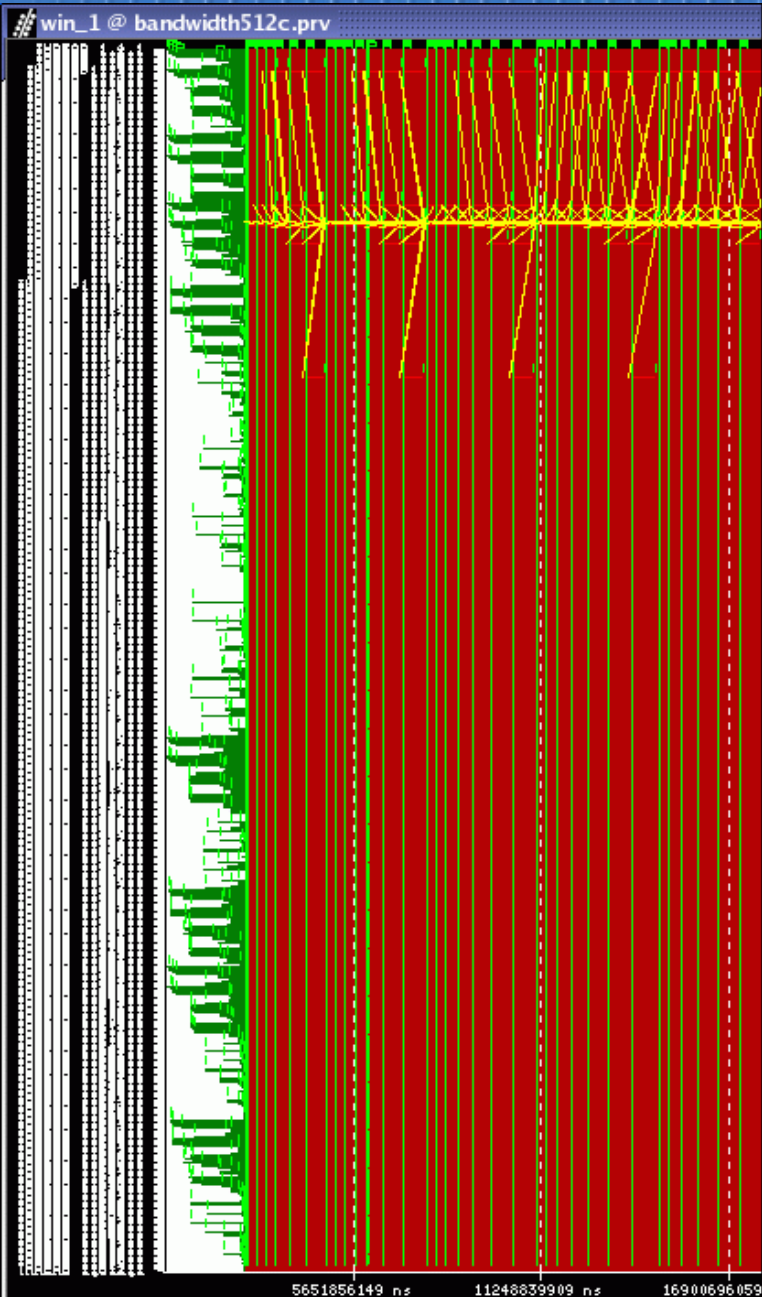
Analyzing System Performance

□ Solving implementation “bugs”



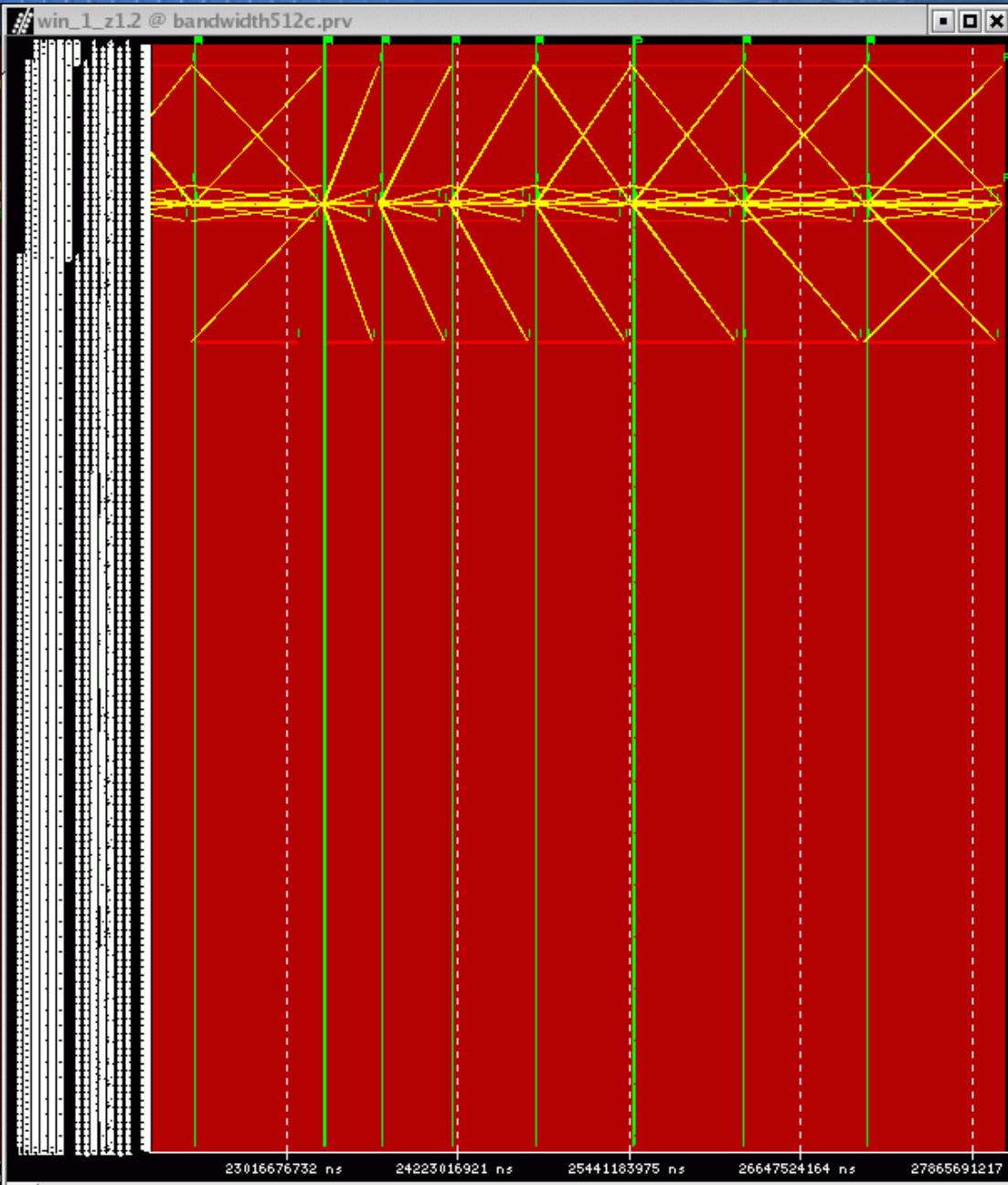
Scalability

- ❑ **65536 nodes are a challenge**
 - ❑ Number of intermediate files
 - ❑ Maximum file size
- ❑ **Restrict both space and time of tracing**
 - ❑ Select representative nodes while trying smaller configurations



TIME (in %)

REDRAW ☐ Comm ☐ Recv ☐ Send ☐ Flag ☐ Color



TIME (in %)

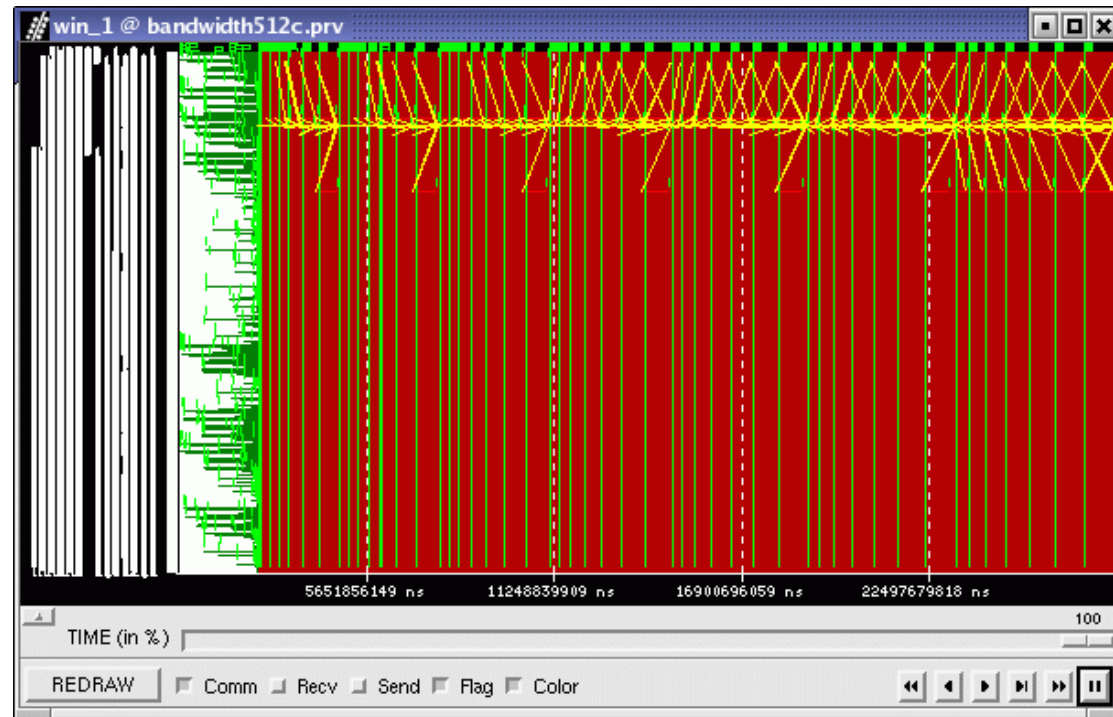
REDRAW ☐ Comm ☐ Recv ☐ Send ☐ Flag ☐ Color

100

Navigation icons: back, forward, search, etc.

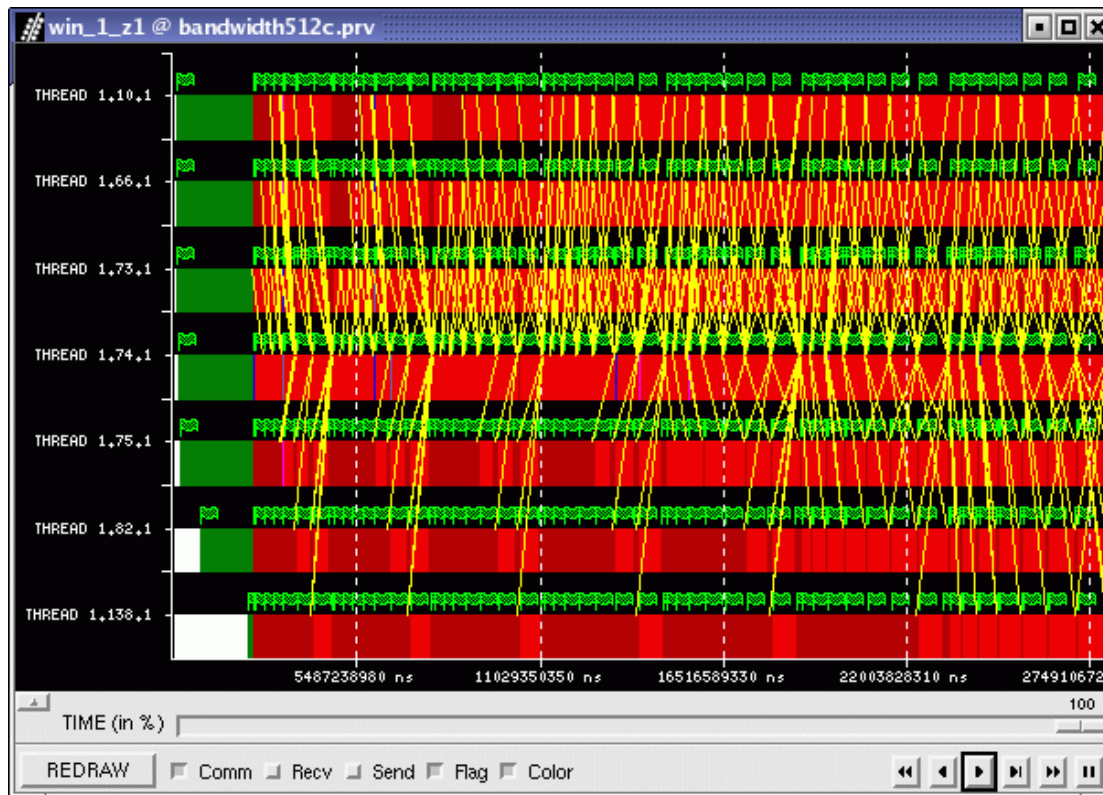
512 nodes

- This trace represents a microbenchmark to measure the bandwidth obtained from the torus network
- Only 7 nodes are really used in a 512-node system

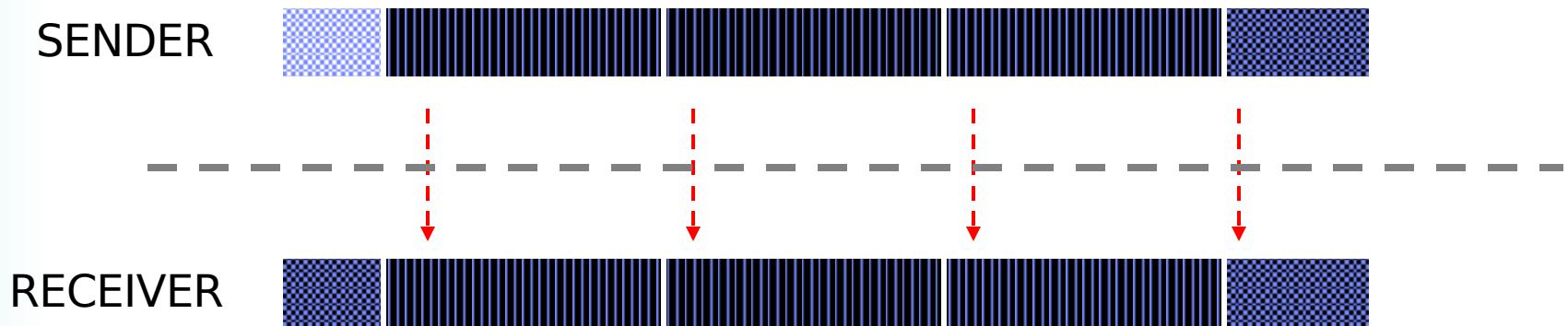


512 to 7

□ Displaying the “interesting nodes”

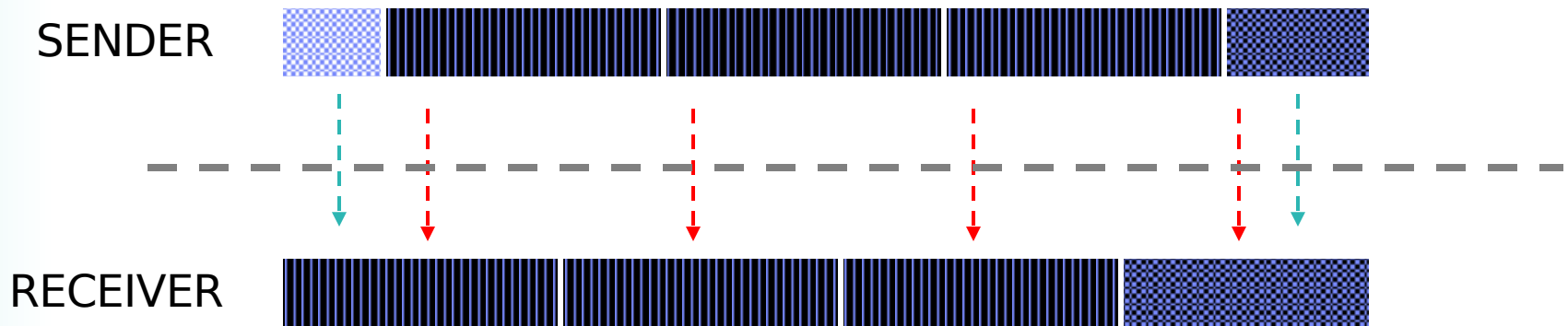


Packetization and alignment



- **Problem:** Torus hardware only handles 16 byte aligned data
- When sender/receiver alignments are same:
 - head and tail transmitted in a single “unaligned” packet
 - aligned packets go directly to/from torus FIFOs
- When alignments differ, extra memory copy is needed

Packetization and alignment



- **Problem:** Torus hardware only handles 16 byte aligned data
- When sender/receiver alignments are same:
 - head and tail transmitted in a single “unaligned” packet
 - aligned packets go directly to/from torus FIFOs
- When alignments differ, extra memory copy is needed

The cost of packet realignment

- The cost (cycles) of reading a packet from the torus into unaligned memory

