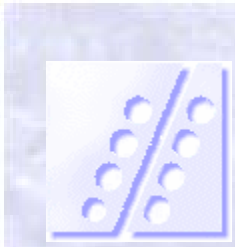


Analysis of AIX traces with Paraver

Judit Gimenez, Jesus Labarta (CEPBA-UPC)

Terry Jones (LLNL)

Technology Transfer Research Training Mobility of Researchers

User Support Education HPC Facilities Parallel Expertise

Index

- Motivation
- AIXtrace2paraver
- Some Examples
- Conclusions

Motivation

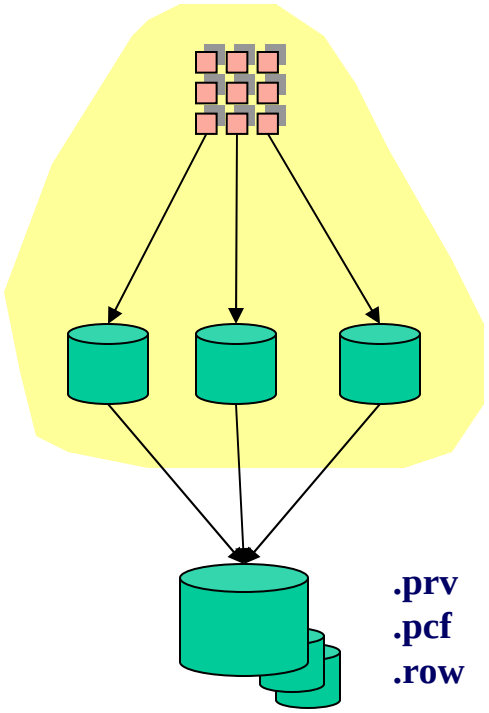
- **AIX Trace @ LLNL**
  - Very detailed information - good!
  - Generate tons of ASCII reports - not so good!
    - ✓ Scripts to extract some info
    - ✓ Lot of details “lost”
- **Paraver**
  - High potential of analysis
    - ✓ qualitative and quantitative
    - ✓ detailed analysis
  - no semantics neither on the tool, nor on the trace format
- **Objective**
  - Analyze with Paraver the information captured with AIX Trace

Index

- **Motivation**
- **AIXtrace2paraver**
  - Approach
  - Information emitted
  - Other features
- **Some Examples**
- **Conclusions**

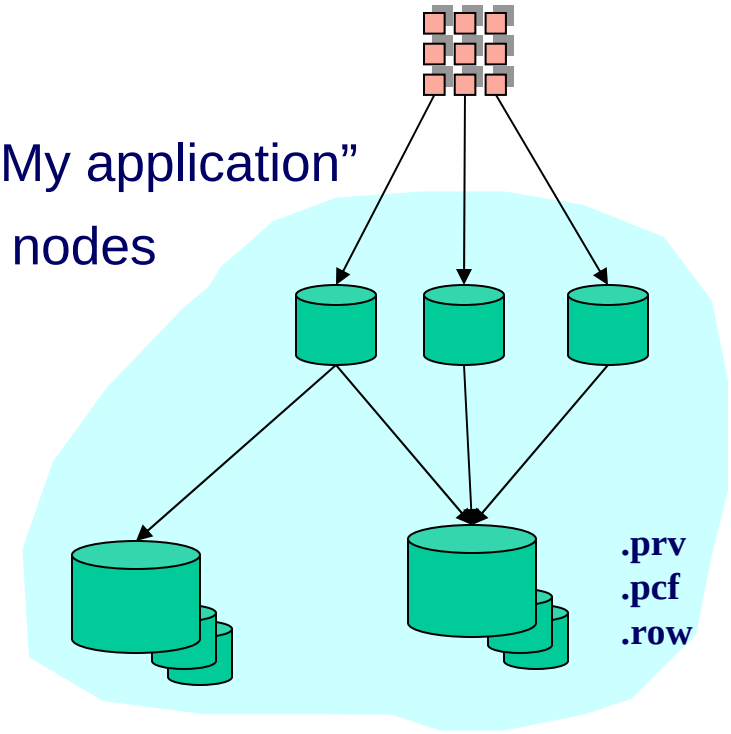
Approach: step 1 - AIXtracelauncher

- **This step is OPTIONAL**
  - Not required by the translator
- **Binary starting the AIX Trace Facility**
  - To simplify the launch of the tool
  - To read the AIX events that we translate
- **Three modes:**
  - Trace node during n seconds
  - Trace node during the execution of an application
  - Sample mode: trace intervals



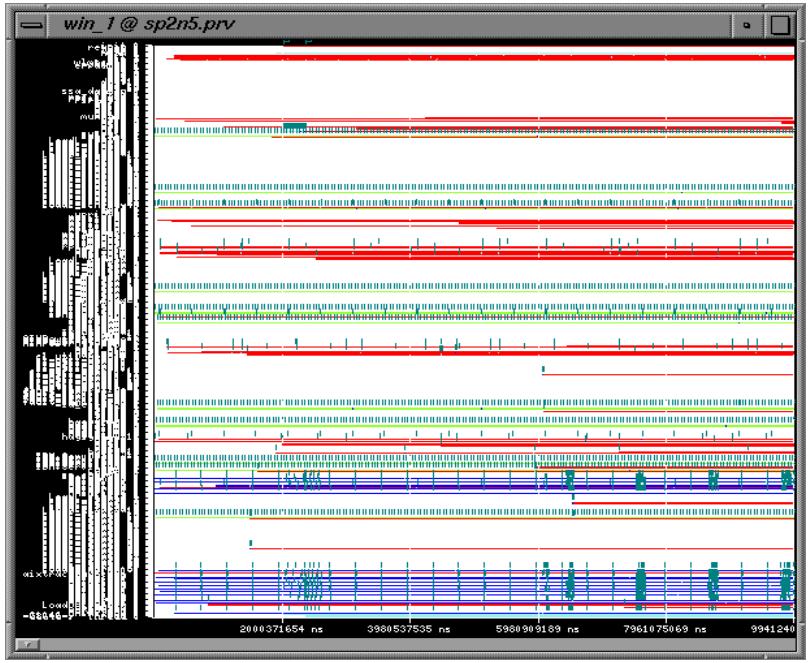
Approach: step 2 - AIXtrace2prv

- **Translator from AIXtrace binary format to Paraver format.**
- **Emit to the .prv trace:**
  - All processes in node
  - Only selected processes from node
  - All processes, mark selected ones as “My application”
  - Only selected processes from different nodes
- **Other options**
  - User events mapping
  - Software counters
  - Print list of processes



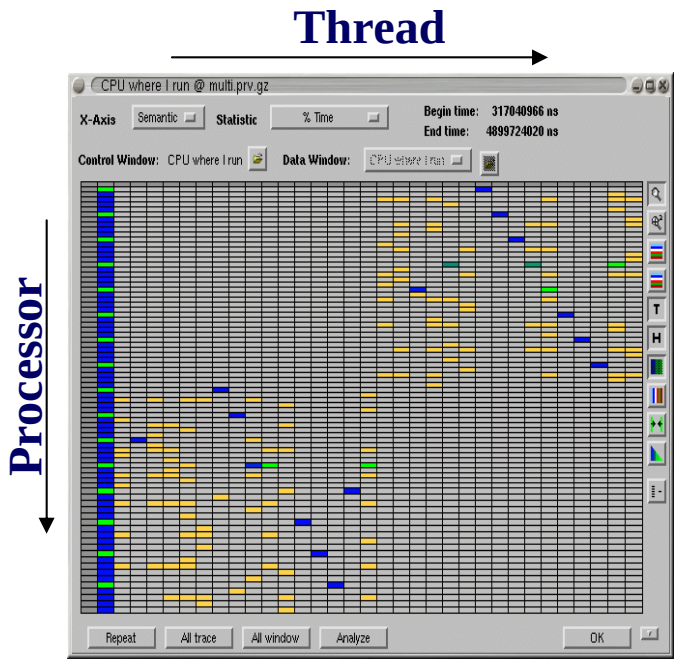
Information emitted to the Paraver trace

- **Per thread information:**
  - States and context switches:
    - ✓ Not created, no info, running, blocked, stopped, ready, yield
    - ✓ On which processor
  - Events:
    - ✓ System calls
    - ✓ Arguments to system call: fd, size
    - ✓ Return values of system calls
    - ✓ Sockets
    - ✓ SCSI driver calls
      - – strategy, bstart, iodone
    - ✓ **User events**



Other features

- **Traces from multiple nodes**
  - Synchronized reading the switch clock
  - Same configuration files
- **Software counters**
  - When high frequency of system calls
    - ✓ large traces
    - ✓ no need for the details of each call
  - Summarization:
    - ✓ at periodic intervals
    - ✓ how many calls of each type
- **Process classification**
  - My application, Other appl, System procs
  - Text file to define system procs names
- **Remove threads with no info**



Only for 32-bit kernels!

Changes in Paraver

...This page has been intentionately left blank

Configuration files

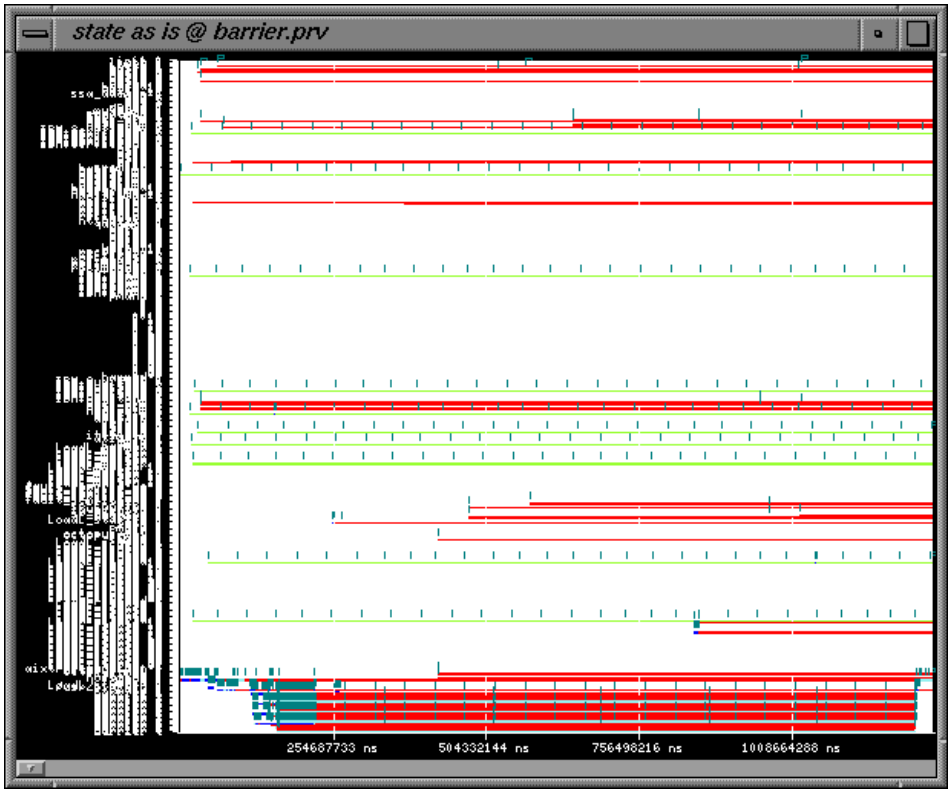
- **Some interesting views captured**
- **Provided in two major directories**
  - Node: analyses applicable to all the processes of the node
    - ✓ resources allocation, process mapping, system calls, disk activity, sockets primitives....
  - Application: applicable to the user application only
    - ✓ few generic views
    - ✓ most specific for each application analyzed:
      - – aggregate, barrier, NAS-BT

Index

- **Motivation**
- **AIXtrace2paraver**
- **Some Examples**
  - System interferences
  - Analyzing MPI behavior
  - IRS run @ LLNL
- **Conclusions**

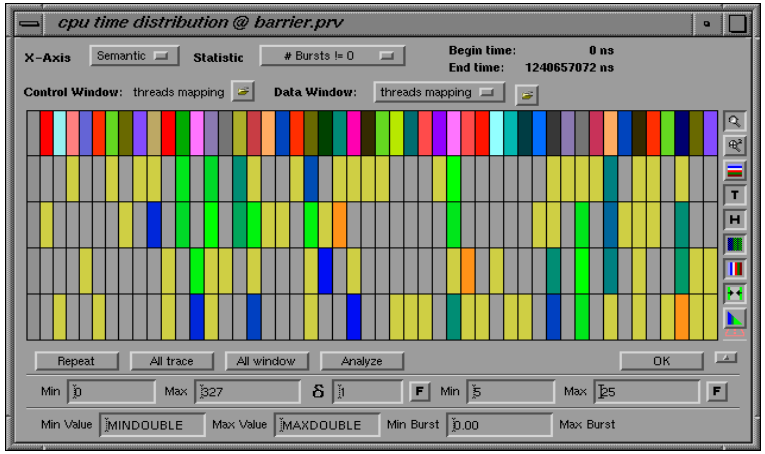
System interferences

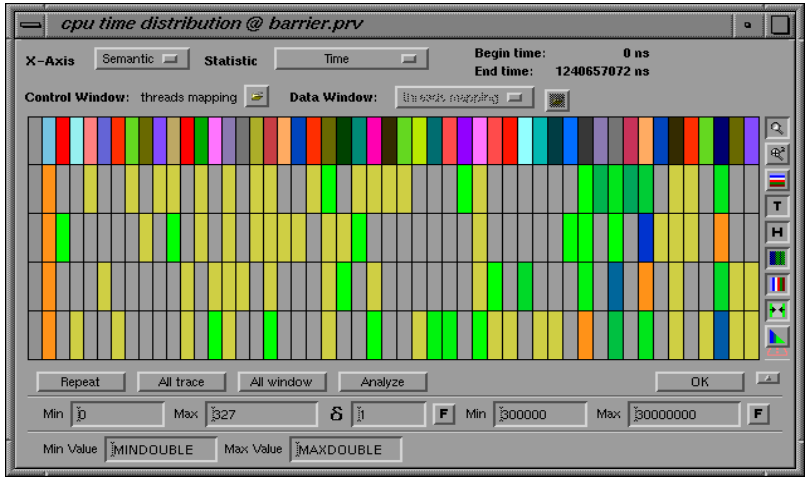
- **Environment**
  - Very fine grain application
    - ✓ Loop barrier - computation
  - 4 tasks in a 4-way node
  - No other users

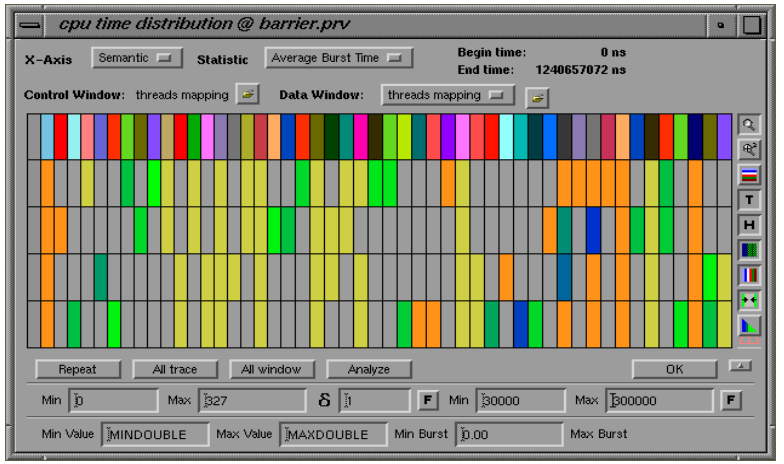


System interferences – CPU time distribution

- **Mapping – many processes run on most of the processors**
- **System processes -Typical runs of few tens of us**

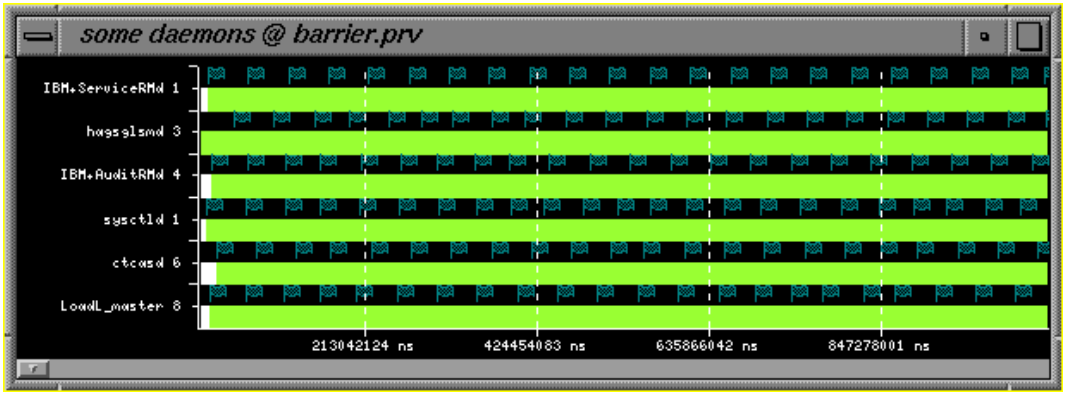


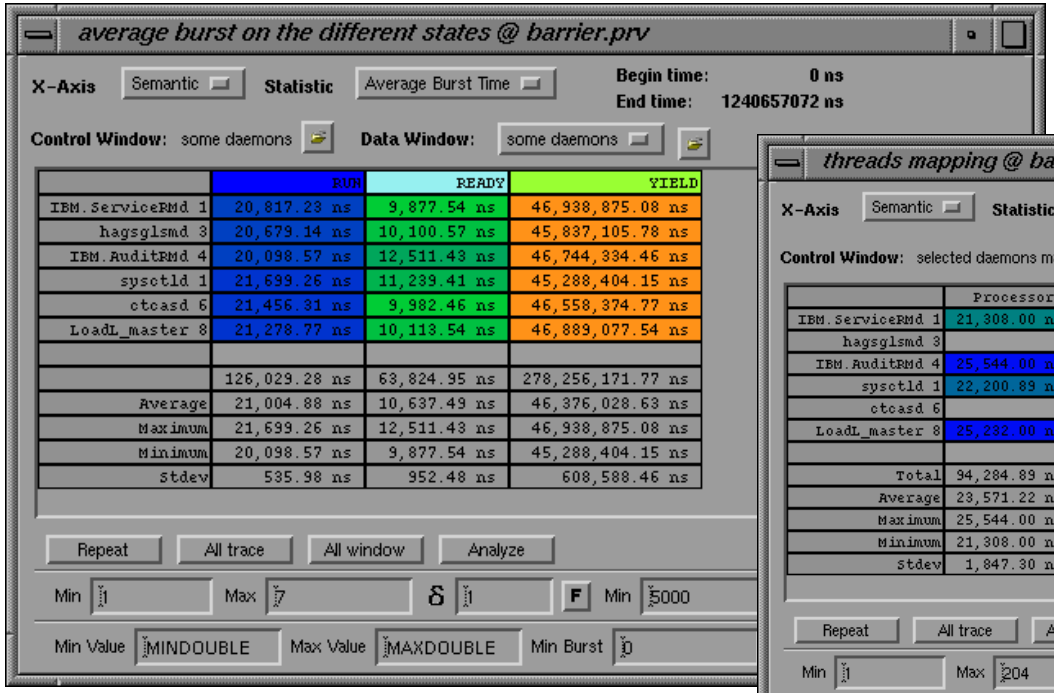


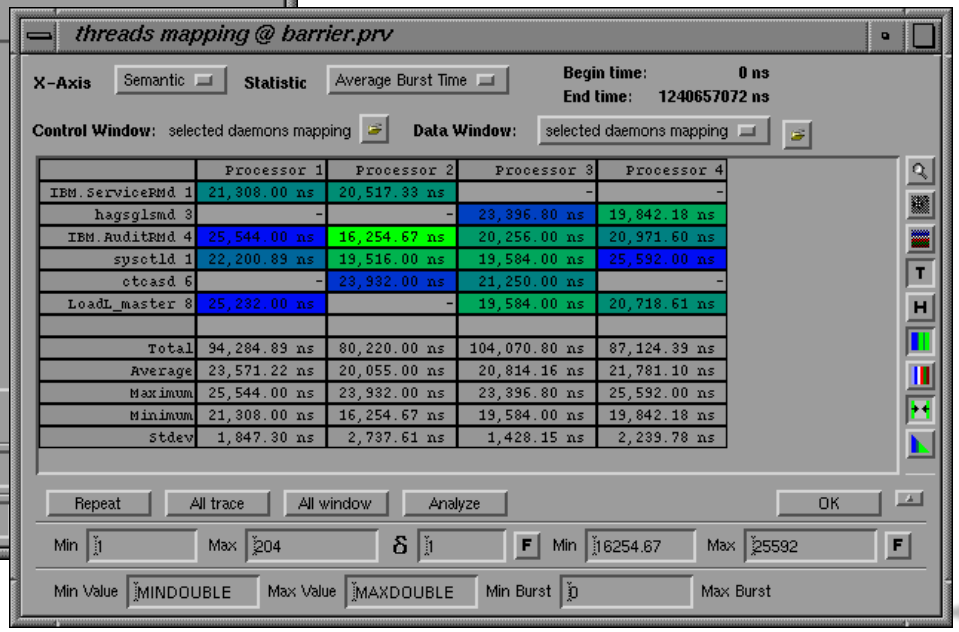


System interferences – system daemons

- **Similar behavior**
  - Yields for ≅ 46.4 ms
  - Run ≅ 21us
- **Most run on many CPUs**

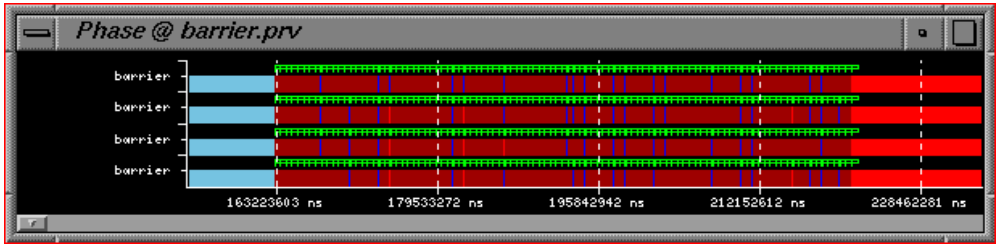


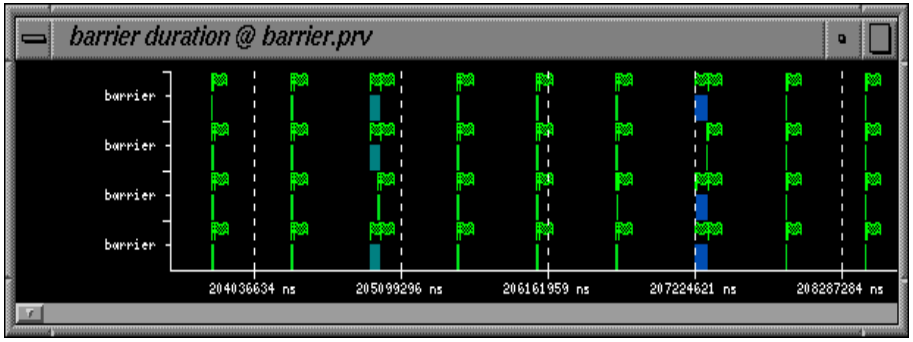


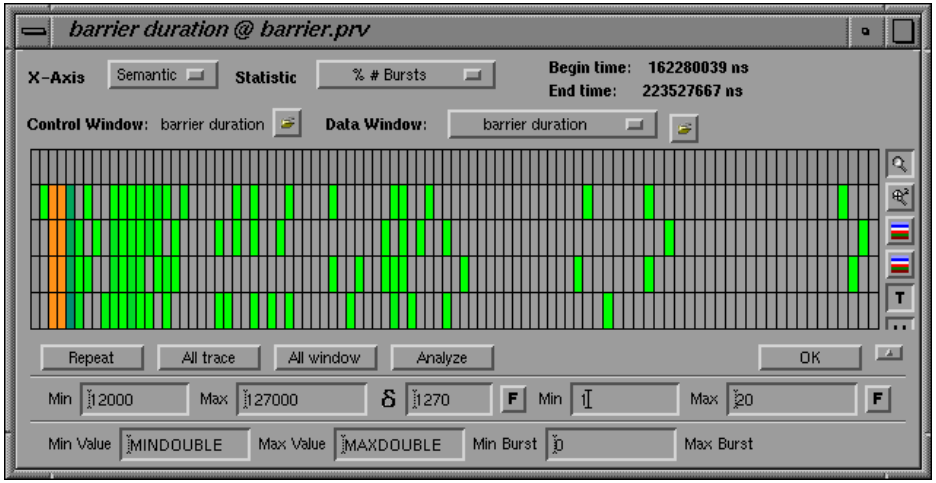


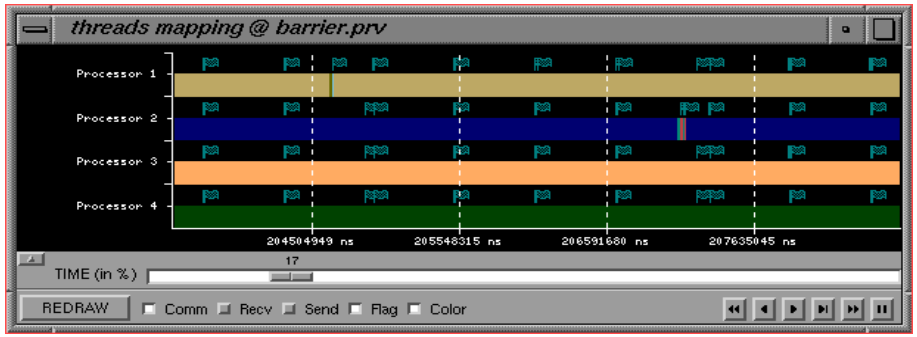
System interferences – impact on the appl.

- **User events**
  - Some “Very large” barriers
    - ✓ Typical – 14us
    - ✓ Large – range 66-93us
  - The cost is paid by all the tasks
    - ✓ 1 task delayed by the system
    - ✓ 3 tasks wait in the barrier







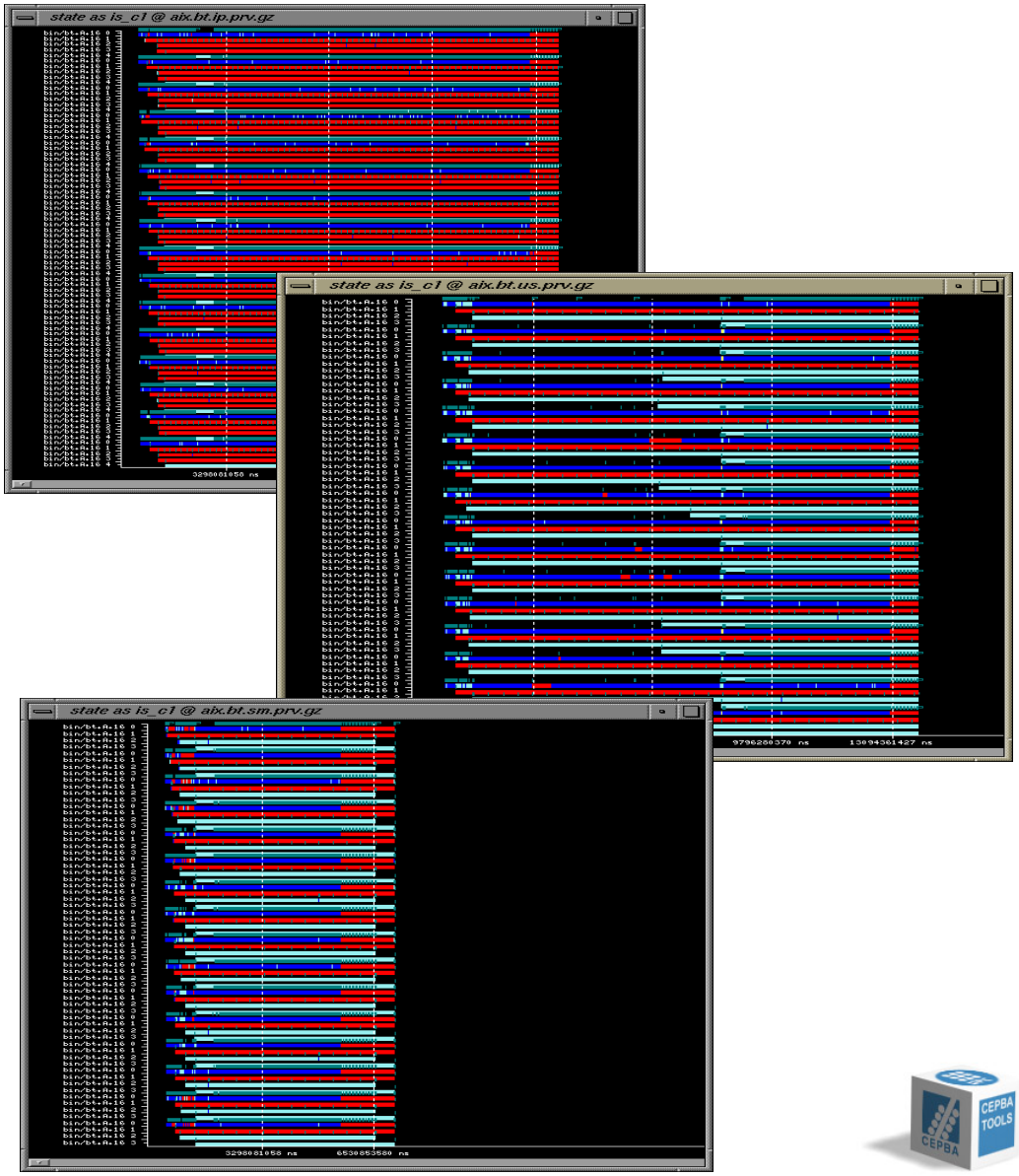


Index

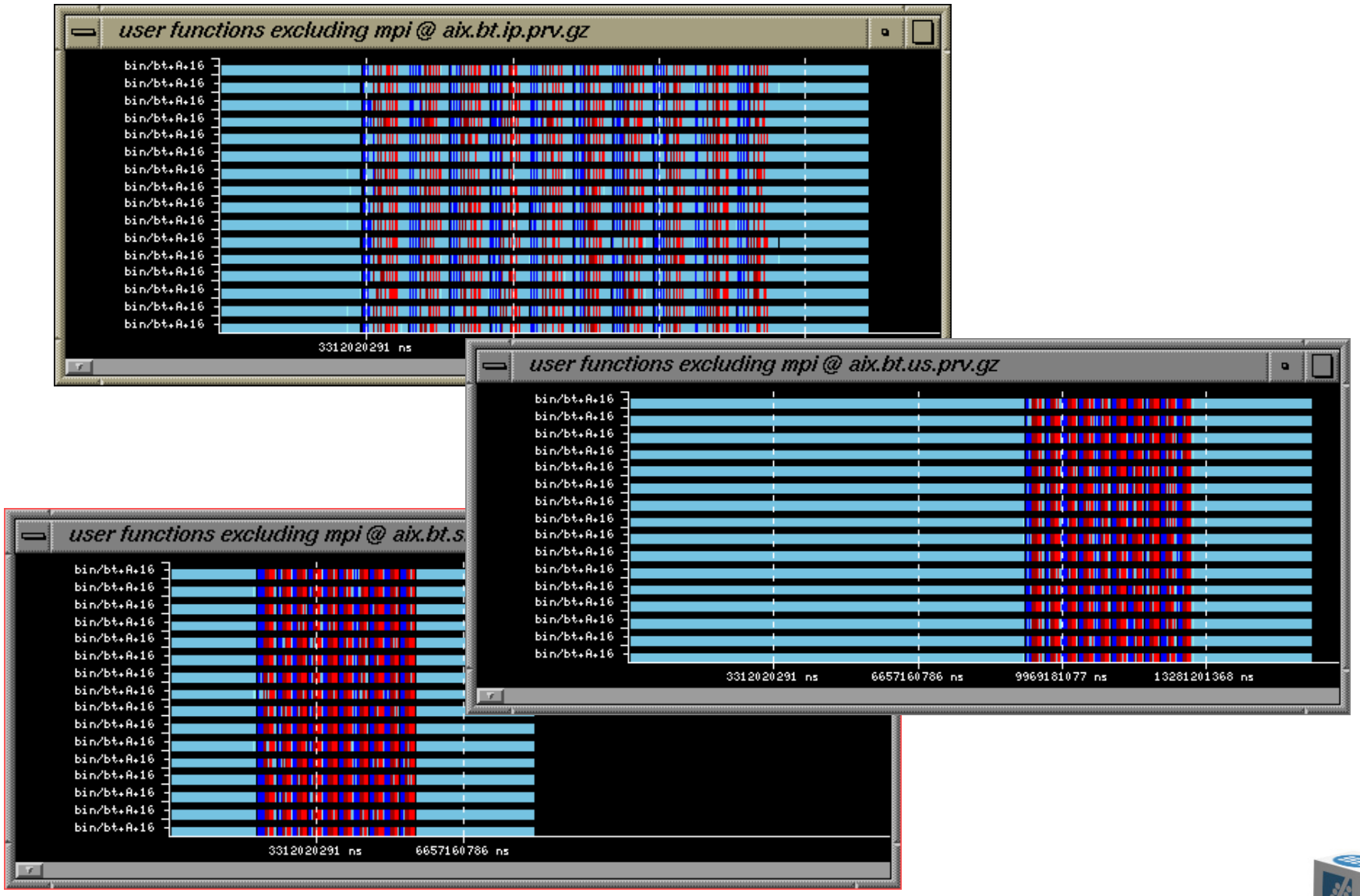
- **Motivation**
- **AIXtrace2paraver**
- **Some Examples**
  - System interferences
  - Analyzing MPI behavior
  - IRS run @ LLNL
- **Conclusions**

Analyzing MPI behavior

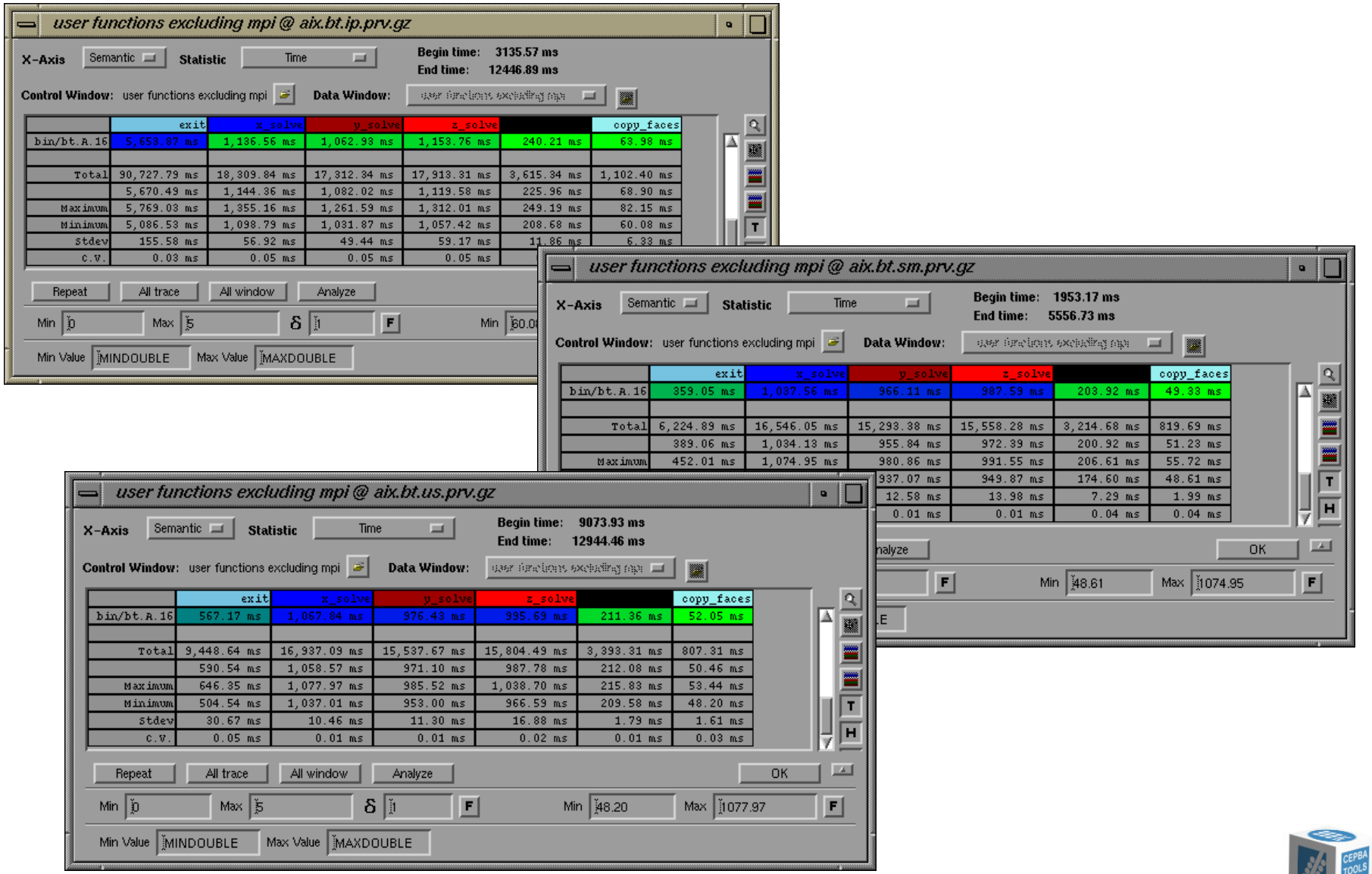
- **Environment**
  - NAS–BT, class A
  - Modified source code to instrument
    - ✓ Some user functions
    - ✓ All mpi calls
  - 16 tasks in a16-way node
  - 3 runs: SM, US, IP



Analyzing MPI behavior – user functions

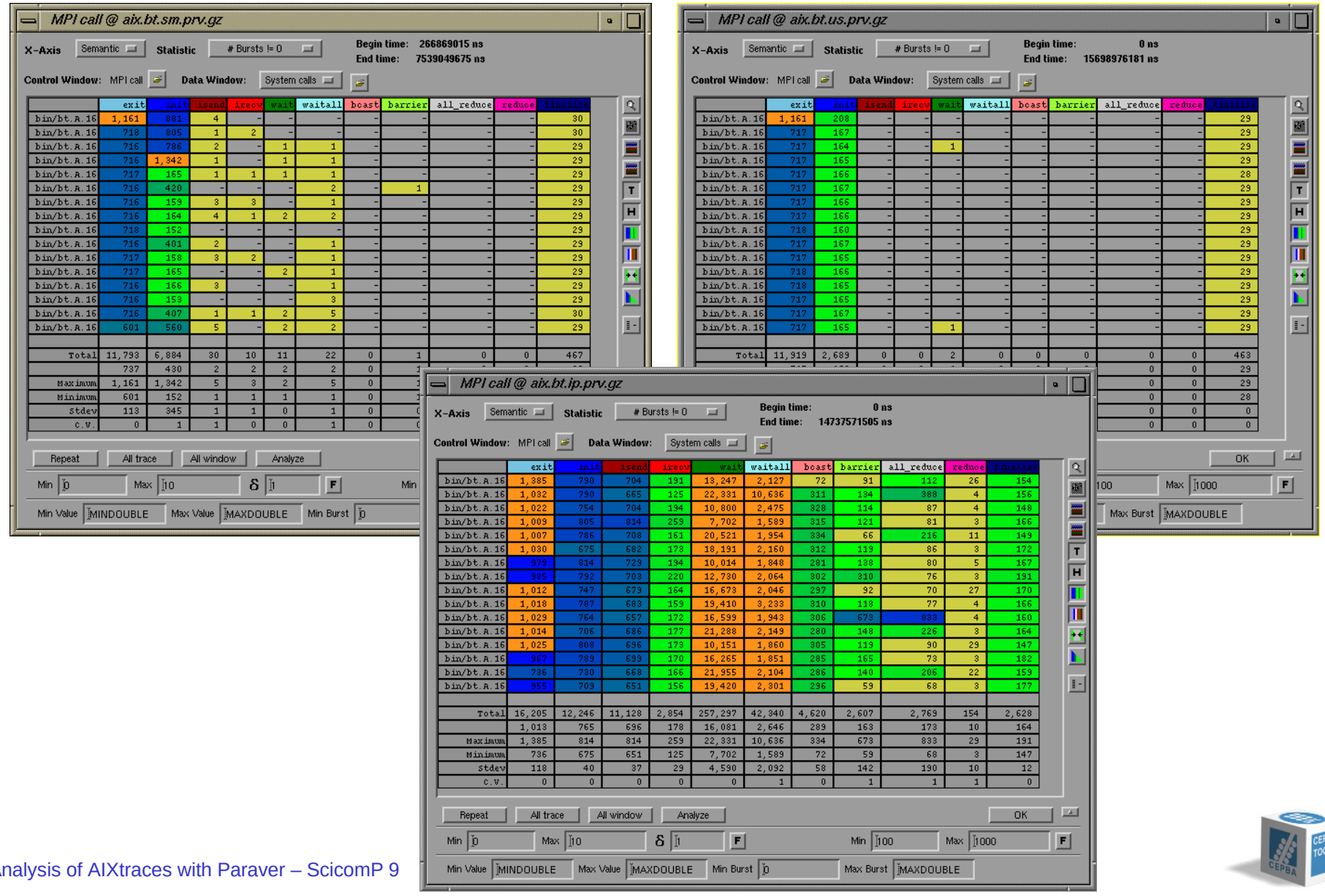


Analyzing MPI behavior – time distribution



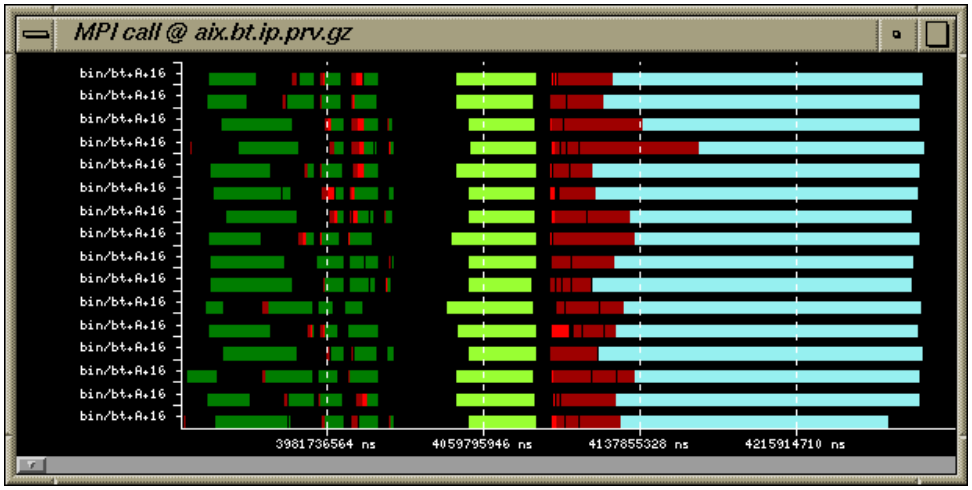
Analyzing MPI behavior – system calls

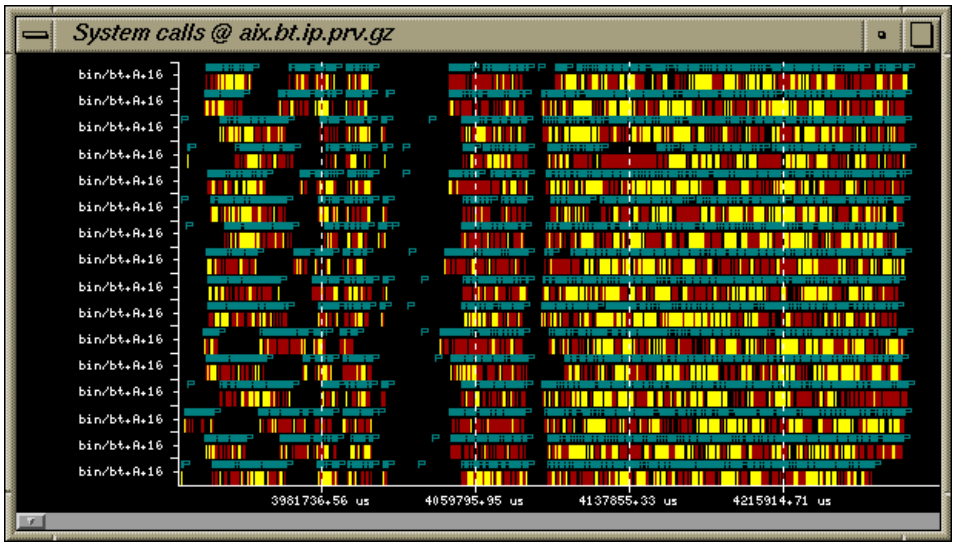
Analyzing MPI behavior – system calls



Analyzing MPI behavior – internals of MPI

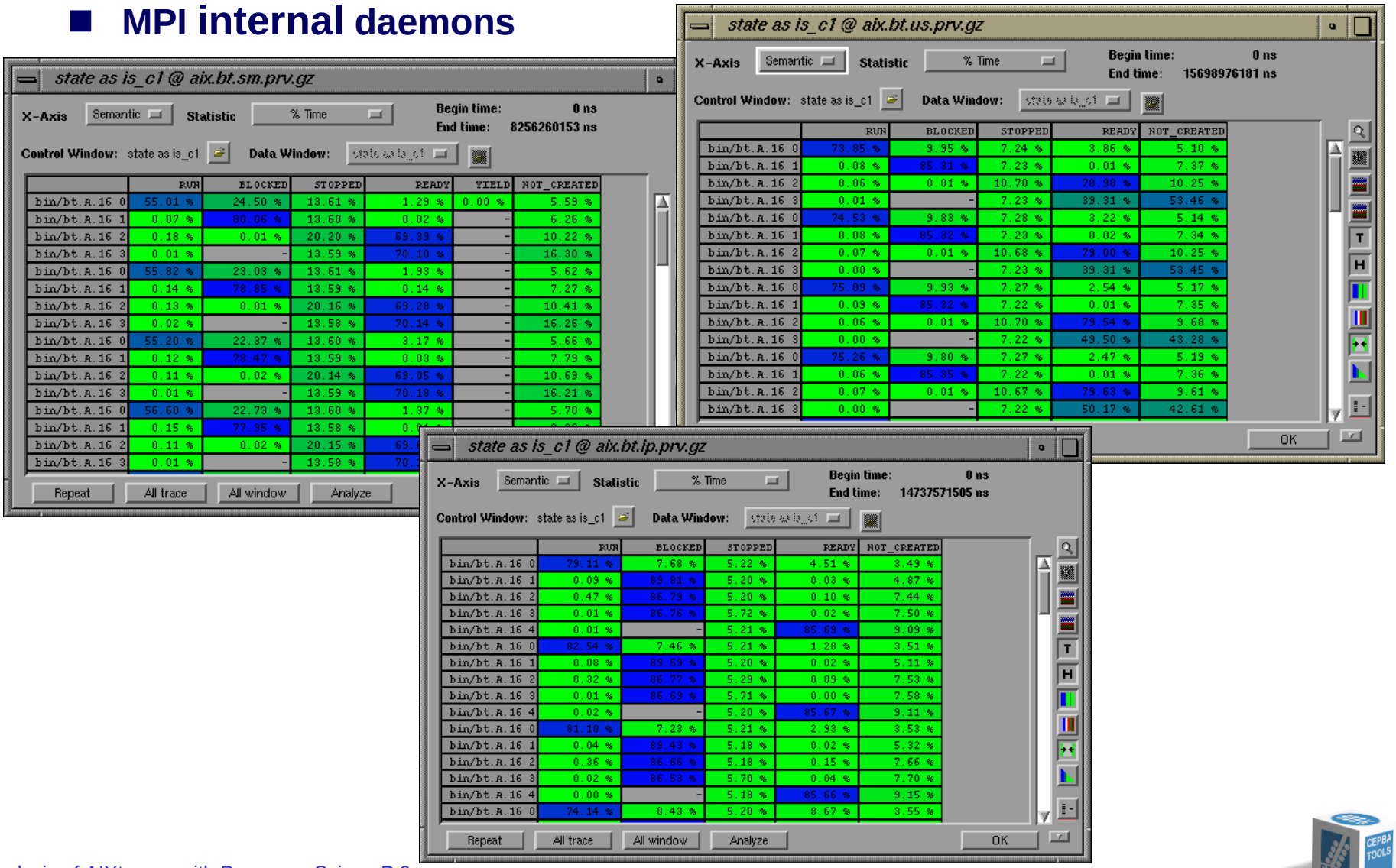
- **IP implementation**
  - MPI calls
  - System calls





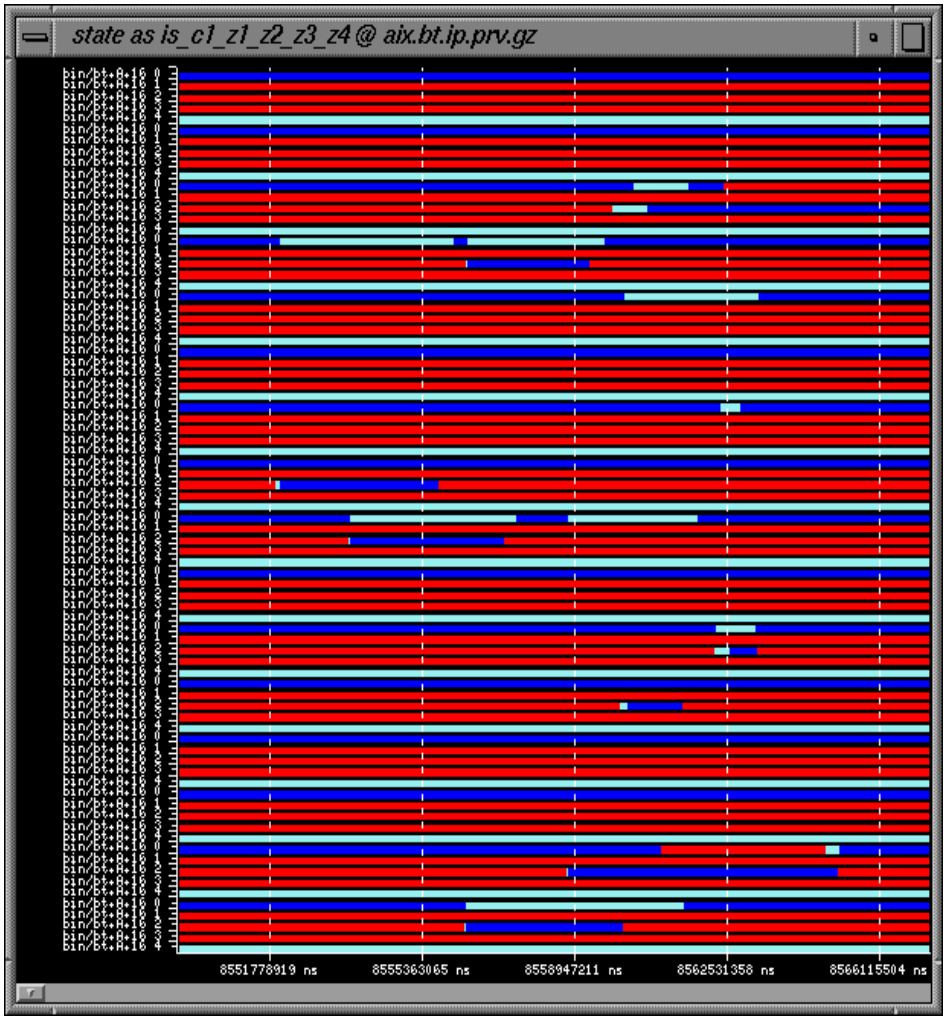
Analyzing MPI behavior – internals of MPI

- **MPI internal daemons**



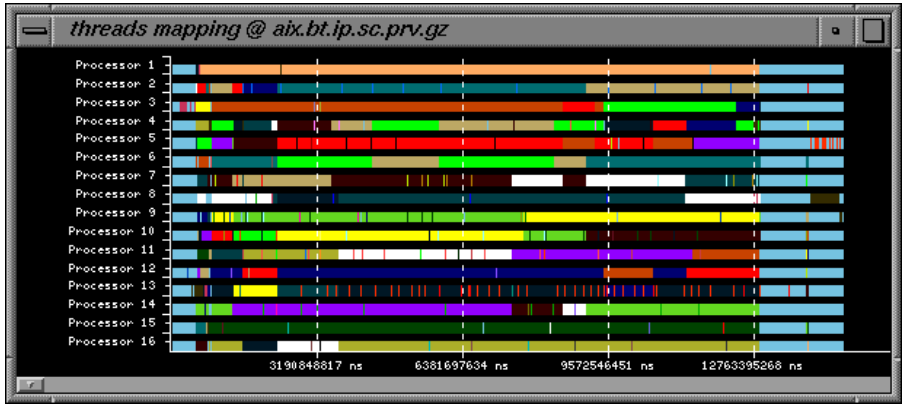
Analyzing MPI behavior – internals of MPI

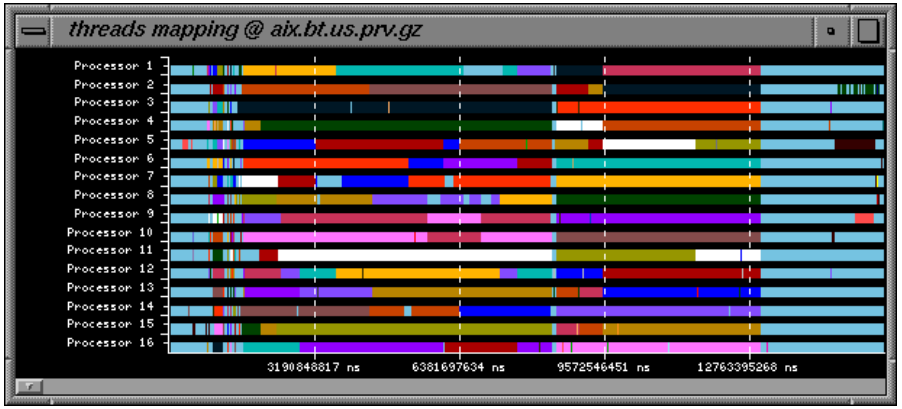
- **MPI internal daemons**
  - Sometimes interfere their own MPI task
  - Sometimes interfere other MPI task

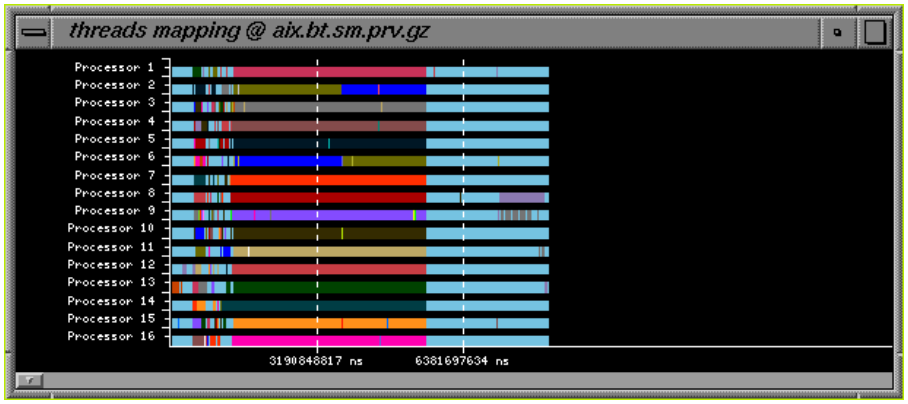


Analyzing MPI behavior – thread mapping

- **Process migration**
  - Initially very high
  - Not many in stable region

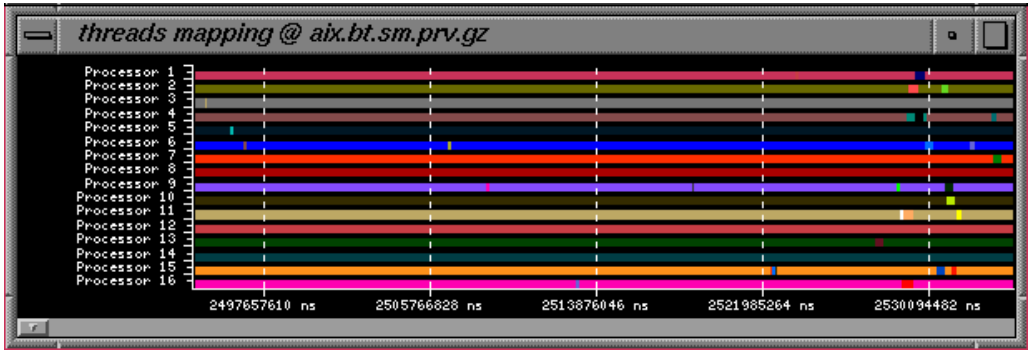




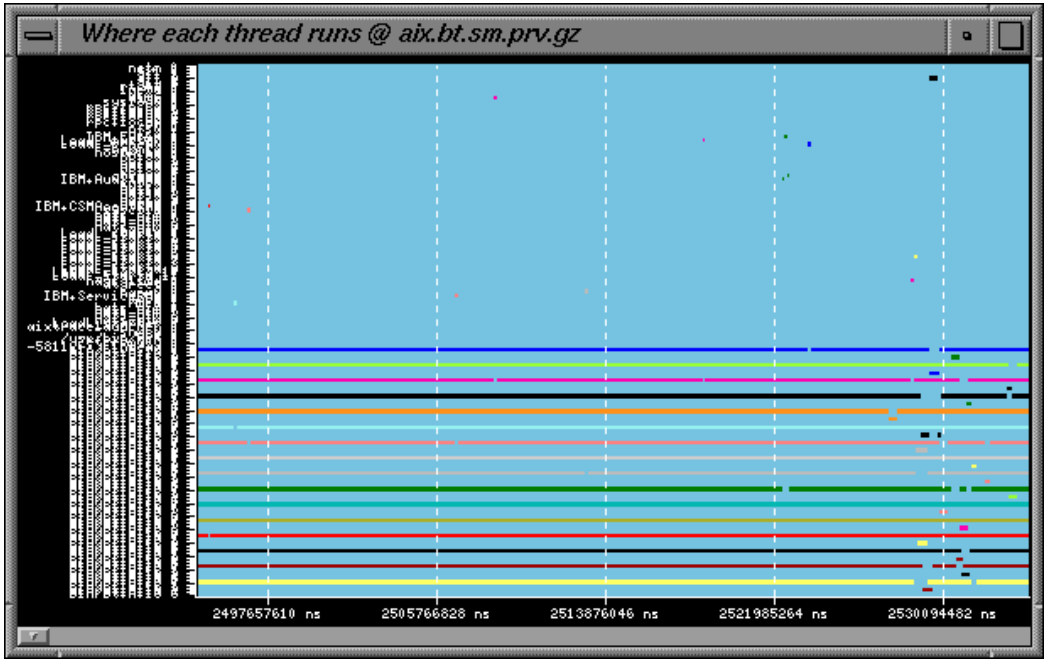


Analyzing MPI behavior – preemptions

- Zooming into stable zone of SM run



- Who ?

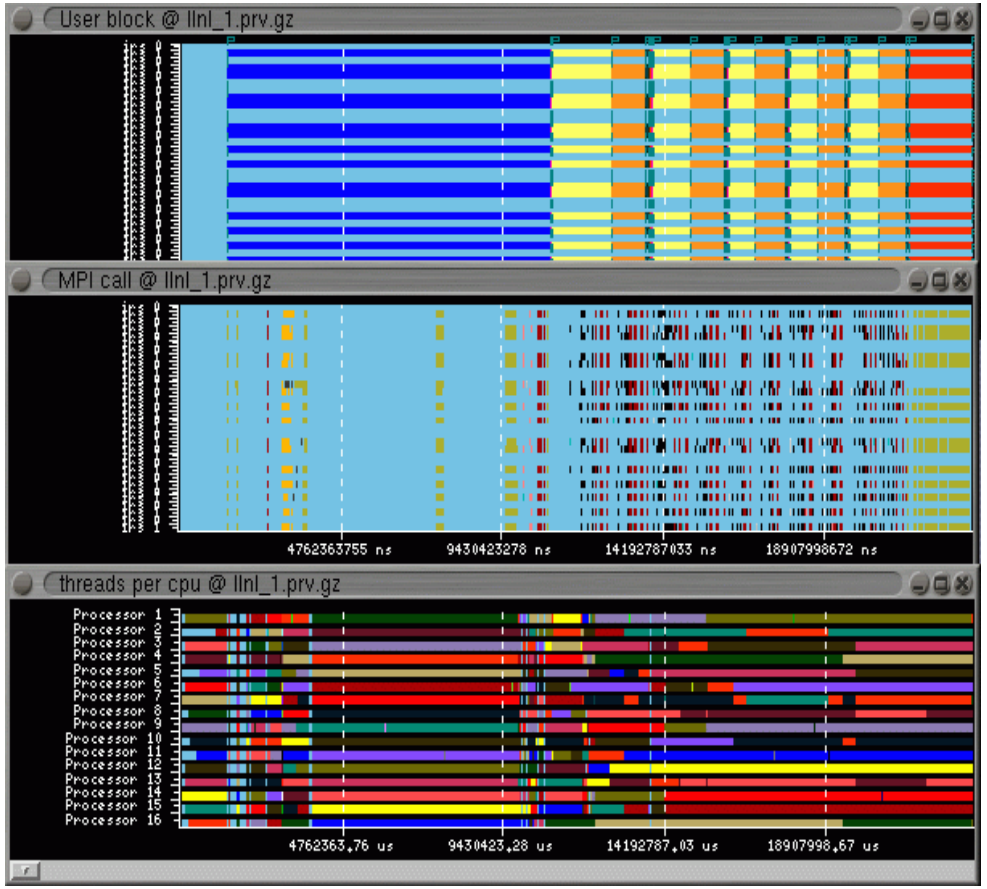


Index

- **Motivation**
- **AIXtrace2paraver**
- **Some Examples**
  - System interferences
  - Analyzing MPI behavior
  - IRS run @ LLNL
- **Conclusions**

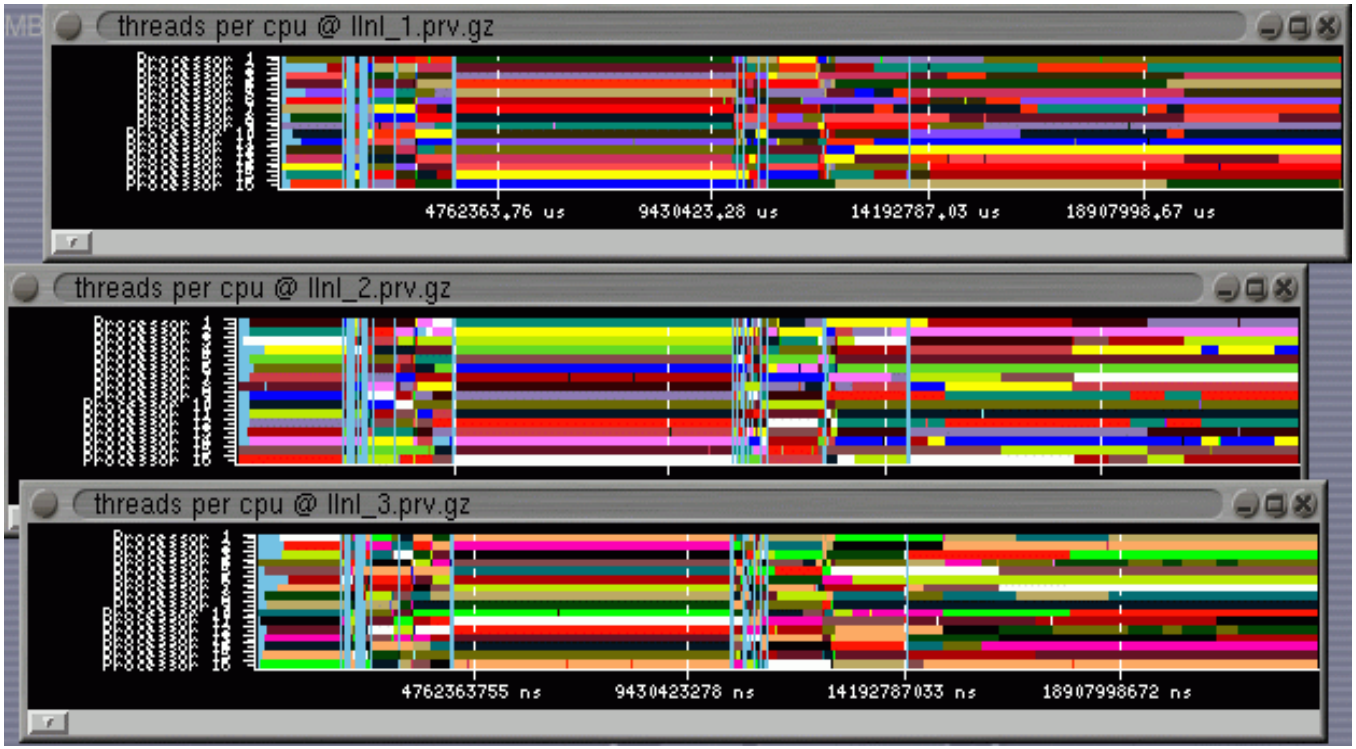
IRS run @ LLNL

- **Environment**
  - IRS run on 22 nodes @ LLNL
  - Trace obtained
    - ✓ without aixtracelauncher
    - ✓ without dumping switch clock
  - Different mapping of the user events



IRS run @ LLNL

- **Multiple nodes view**
  - with manual alignment
  - Synchronized scheduling effects ?



Index

- Motivation
- AIXtrace2paraver
- Some Examples
- Conclusions

Conclusions

- **Description of the translator AIXtrace2prv developed under support from LLNL (Contact: Terry Jones)**
- **Shown the huge potential of combining**
  - The extraordinary amount of data captured by AIX trace
  - The extraordinary flexibility and processing power of Paraver to extract information from raw performance data
- **Porting to 64-bit kernels…?**
- **Mechanism to automatically insert user events…**
- **Available to Paraver users or trough an evaluation license (www.cepba.upc.es/paraver)**