MPE: logging and post-mortem visualization toolset

Arquitectura de Computadores II
Universidad Carlos III de Madrid

What Is MPE/Jumpshot?

- The "quintessential" MPI logging and post-mortem visualization toolset
- MPE – Multi-Processing Environment
  - A software package for MPI programmers
  - Has three main parts:
    - A *tracing* library that outputs all MPI calls to stdout
    - A shared-display parallel X *graphics and animation* library
    - A *logging* library for logging events
    - *Note*: MPE/Jumpshot "logging" -> what we call tracing
- Jumpshot
  - A visualization tool for logfiles created by the MPE package
  - Programado en Java (crossplatform)
  - Provides a "time line" (GANTT) view of MPI and program events
  - Also has basic search and summary (histogram) functionality

Logfiles: What's In A Format?

- By default, MPE still outputs logfiles in CLOG
  - Low overhead
  - Can be easily converted to other formats as needed
- SLOG: "scalable" format
  - State-based logging format
  - Visualization tool: Jumpshot-3
    - Rewrite of Jumpshot-2 to use SLOG
    - Can scale to ~GB logfiles
- SLOG-2: Current logfile format
  - Next-generation SLOG file format
  - "Graphical" logfile format to speed logfile parsing
  - Visualization tool: Jumpshot-4

MPE Overview

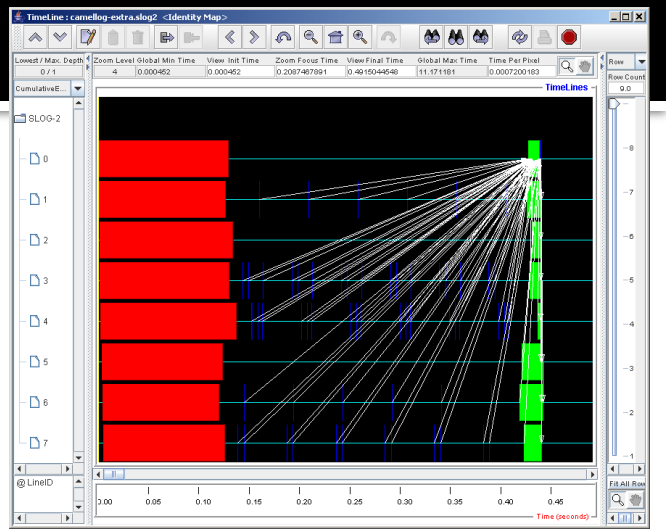
- Tracing capability
  - I nstrumentación automática: `mpicc -mpitrace`
    - Writes to stdout at every MPI call, eg
      - `[1] Starting MPI_Send with count = 28, dest = 0, tag = 0...`
      - `[1] Ending MPI_Send`
  - Equivalent "manual" method: printf
  - Very simple & intuitive
- Parallel graphics ability
  - Automatic instrumentation: `mpicc -mpianim -L/usr/X11R6/lib -lX11 -lm`
    - Displays graphics on one machine
    - Circle for each process, arrow indicate sends/receives
    - Slows down execution considerably
  - Graphics are also available via library calls
    - Calls seem relatively easy to use: MPE_Draw_string, MPE_Draw_circle, MPE_Update, etc
    - Probably not all that useful

MPE Overview

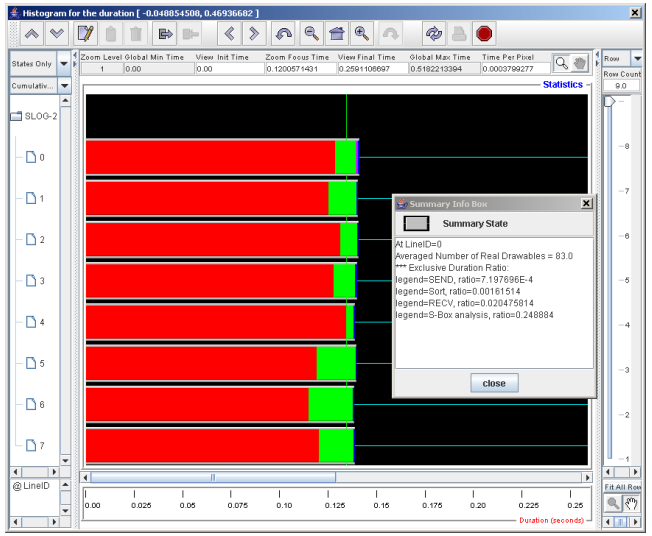
- Logging ability
  - Automatic instrumentation: `-llmpe -lmpe`
  - Logs start and stop of events
  - Can overlap starting and stopping of events
  - Can add "custom" events
    - Easy to do using library calls
      - MPE_Log_get_event_number: crea un nuevo evento
      - MPE_Describe_state: gives name and color to event
      - MPE_Log_event: records event in logfile, uses MPI_Wtime to get global time
    - Custom events show up in Jumpshot-4 just like events from automatic instrumentation
  - Conventions
    - Automatic instrumentation uses all caps (SEND, RECV)
    - Manual instrumentation uses mixed case

Jumpshot Overview

- Jumpshot-4 supports two types of visualizations for metrics
  - Timeline (right, top)
  - Histogram (right, bottom)
- Visualization is dependant on SLOG-2 format and Data model
  - Real drawables
    - State – Single timeline ID, start/end timestamp
    - Arrow – Pair of timeline IDs, start/end timestamp
    - Event – Single timeline ID, single timestamp
  - Preview drawables
    - Amalgamation of real drawables
    - One corresponding type for each of the real drawables
    - Serve to optimize performance of visualization



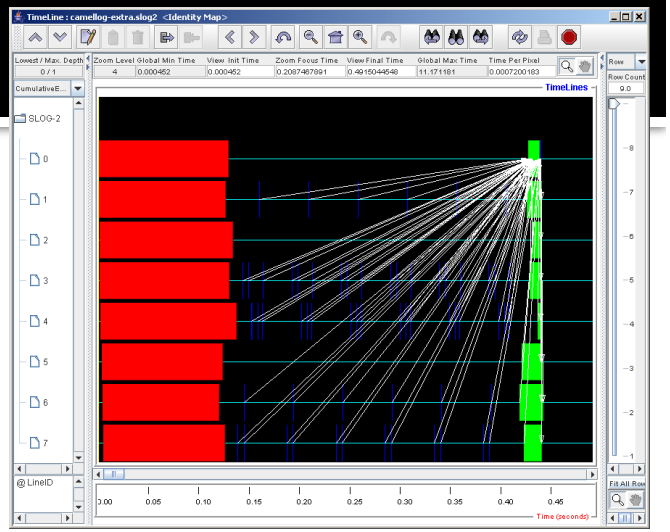
Timeline view



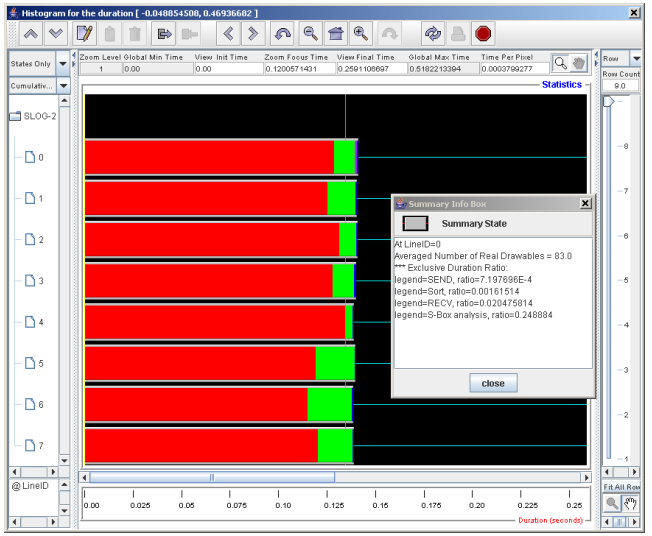
Histogram view

Jumpshot Overview

- Emphasis on providing useful profile analysis from
  - High-level (entire program execution) view
  - Low-level (individual events) view
- Nice features
  - Intuitive interface
  - Automatically converts from CLOG to SLOG-2
  - Very good support for zooming and scrolling
  - User manual very thorough
- Things that could use improvement
  - Java application -> uses a lot of memory (~70-100MB during typical runs)
    - Memory uses seems to scale nicely with logfile size though
  - No direct support for non-event-based data (running averages, time-varying histograms for cache miss numbers, etc)
  - Documentation a little unclear/excessively technical in some places



Timeline view



Histogram view