

Projects description

- Project 1: Job and Task inter-arrival time
 - **Team: Carpi, Scalia**
 - Input: <trace>, <event>, <condition>
 - Output: Inter-arrival time between different <event>
 - Statistics on arrival time of jobs (at the cluster) and task (at a machine). The tool must accept every event that can be associate to a job/task. Multiple conditions (mainly in AND logical conjunction) must be allowed.
 - Different window sizes must be used for the analysis.
 - Report in which you describe the work done and analyze the results obtained.
- Project 2: Task/Job and Task/Machine
 - **Team: Botti, Schiavi**
 - Input: <trace>, <condition>
 - Output: number of tasks per job and number of task per machines
 - Statistics on number of tasks per job and number of task that can be worked by the same machine simultaneously. Multiple conditions (mainly in AND logical conjunction) must be allowed. JobID and/or MachineID can be specified as conditions by the user.
 - Different window sizes must be used for the analysis.
 - Report in which you describe the work done and analyze the results obtained.
- Project 3: Jobs and Tasks response time
 - **Team: Brusamento, Donetti**
 - Input: <trace>, <start_event>, <stop_event>, <condition>
 - Output: time between <start_event> and <stop_event>
 - Statistics on the response time of jobs and tasks. The tool must accept every event that can be associate to a job/task. Multiple conditions (mainly in AND logical conjunction) must be allowed.
 - Different window sizes must be used for the analysis.
 - Report in which you describe the work done and analyze the results obtained.
- Project 4: How to set up a right window size for other experiments?
 - **Team: Gazzetta, Malberti**
 - Input: <trace>, <event>, <WS>
 - Output: statistics on <event> based on the specified <WS> (#events, %events wrt. the whole trace, average/variance among different window sizes, ecc.)
 - Statistics on different events and different traces in order to understand what is the best window size to use based on the trace and the event considered.
 - Different window sizes must be used for the analysis.
 - Report in which you describe the work done and analyze the results obtained.
- Project 5: Machine life cycle
 - **Team: Fiorenti, Samà**
 - Input: <trace>, <conditions>, <event_1> [, <event_2>]
 - Output: MTTF, MTTR, frequency of <event_1> in cluster/machine
 - Statistics on machine life wrt. different moment of the day/month. In this case, could be useful to consider the whole trace (29 days). Note that <event_2> is not mandatory. It must be specified to compute MTTF and MTTR, but it is useless if you are analyzing the frequency of <event_1>. Multiple conditions (mainly in AND logical conjunction) must

- be allowed.
- Different window sizes must be used for the analysis.
- Report in which you describe the work done and analyze the results obtained.

Tool description

Even if you can use every kind of tool to extract a smaller trace from the whole one provided by Google, we developed a script that you can use for that purpose. It can be downloaded from BeeP. This is the help that you can print with the command “java -jar MergeTraces.jar --help”:

```
--help : Print this message
-S, --start : Start time [usec]
-F, --finish : End time [usec]
-f, --file : Files to be merged (*). They must be passed in chronological order.
```

The period () can be set with:*

```
-u, --micro : Sample period in microseconds
-m, --milli : Sample period in milliseconds
-s, --sec : Sample period in seconds
-M, --min : Sample period in minutes
-h, --hour : Sample period in hours
-d, --day : Sample period in days
```

Other parameters:

```
-D, --directory : Directory where the output files are saved
```

** is for mandatory parameters*

File(s) and **period** are mandatory options. Note that the file(s) cannot be passed as an archive.

Before using the script you have to *extract* it/them. Moreover, file(s) must be passed *sorted* in a chronological way.

The **Period** can be specified with different time measure. If you want only one output file, you can use “-d 30” (since in the trace there are logs for 29 days); in that case you’ll miss events that were not finished within the 29th day.

Start Time and **End Time** are set by default to 0 and $2^{63}-1$, respectively. You can change them with -S and -F options. Note that they must be specified in microseconds.

Finally, you can also specify a **Target Folder** where the output will be saved.

About traces

As said in class, you can start working on May 2nd, 2011. You can obtain that sub-trace using these parameters (independently from the script used for that purpose):

- start time = 186000000000 (May 2nd, 2011 – 00:00 EDT)
- stop time = 1050000000000 (May 2nd, 2011 – 23:59 EDT)
- files = from part_3 to part_20 for **job** and **task** events, part_0 for **machine** events.

Please, note that the start time (May 1st, 2011 – 19:00 EDT) is 600,000,000 μ s. Use this value to compute other relevant timestamps.