TER HPC

Rui, Ammar

April 29, 2024

Overview

1. **Task-Based Runtime Systems**
   - Overview of Systems
     - StarPU
     - PaRSEC
   - Features and Benefits
2. **Implementation Code**
   - Matrix/Tiles Implementation
   - Integration with CUDA and MPI
3. **Optional Parts**
   - Out of Core Support
   - Parallel Worker Support
   - Hierarchical DAG
4. **Benchmark**

StarPU

- Task Based Runtime System
- Manage and optimized Heterogeneous machine combining CPU/GPU
- Efficient for cluster
- Data Management via Handle
- DAG task system

ParSEC

- Parametrized Task graph
- More compact Representation of task dependency
- More flexibility in thread utilisation
- unlike StarPU Support communication network like UCX or LCI

ParSEC

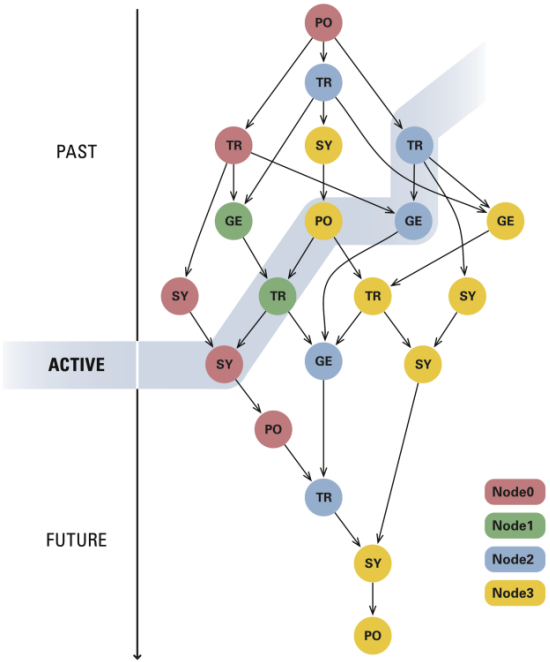


Figure 1: The PaRSEC runtime walks the DAG using a concise representation that instantiates only the relevant tasks at each computing node. Only the active, local tasks need to be stored and considered.

Matrix and Tile implementation

Matrix

- Vector of Tile
- Size of Matrix, Arrangement and size of tile

Tile

- A vector of double/float that is allocate with starPU
- A handle to manage the vector
- Size of Tile

Gemm

Done in two step

- First do the Computation of $C = \beta * C$
- Second do the Computation αAB

Cuda Support

Add for cuda support

- Function that support cuda in kernels.cu
- add .cuda_funcs in the codelets to use cuda

MPI Support

- Init with Mpi starpu special function
- Put Tag to all tile to know which nodes control which tile
- Shutdown with starpu mpi special function

Reduction Support

In matrix.cpp in the gemm function

- We declared the reduction method + its initialisation for its data handle
- In task we used STARPU_REDUX for the handle that use reduction

Out of Core Support

There is Out of core support

- We declared a new disk with starpu_disk_register
- if it is full it will be flush out for new memory
- When we declared data in the data handle we put -1 to let starpu manage the memory disk

Parallel Worker Support

- We declare Parallel worker with starpu parallel worker function
- When calling a task use STARPU_POSSIBLY_PARALLEL
- Since we use gemm of openBLAS, we supposed that it will be divided between thread
- unregister Parallel worker with starpu parallel worker function

Hierarchical DAG

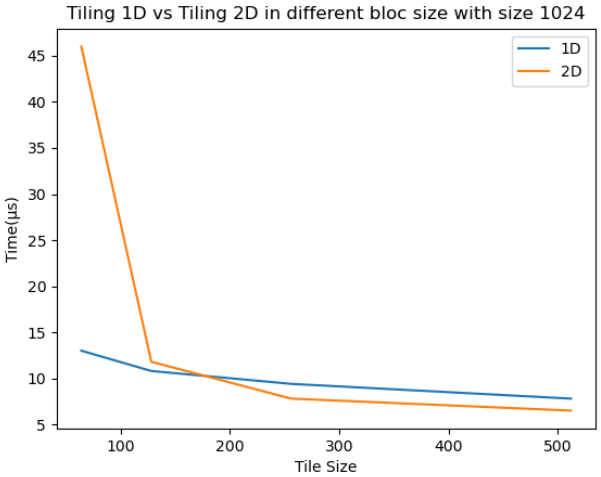
In matrix in gemm function

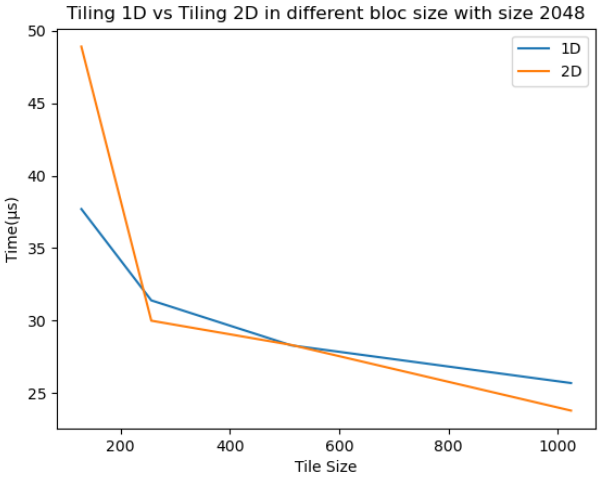
- We Partition and unpartition the Data Handle in 4 bloc

Then the bubble task

- We do a sort of 2D tiling in a 2x2 Grid

1D vs 2D Tiling





Scheduler comparaison

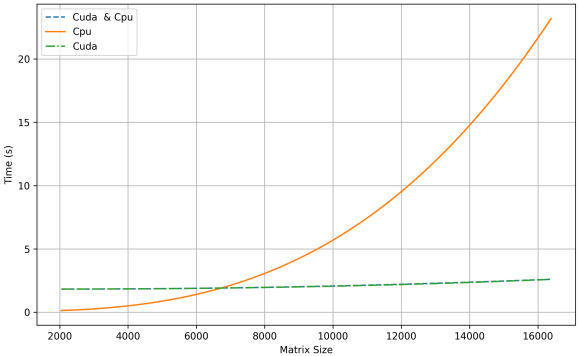


Figure 2: DM Scheduler.

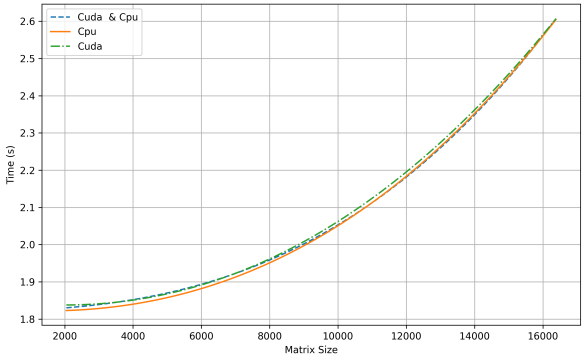


Figure 3: DMDA Scheduler.

Scheduler comparaison

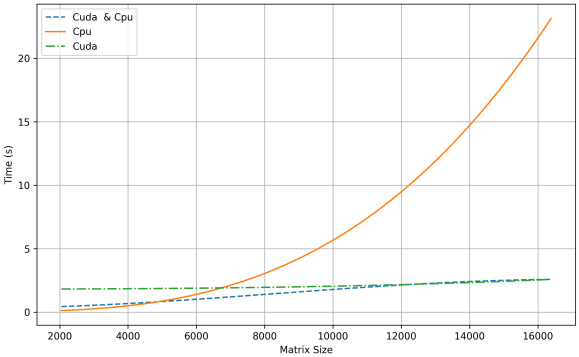


Figure 4: LWS Scheduler.

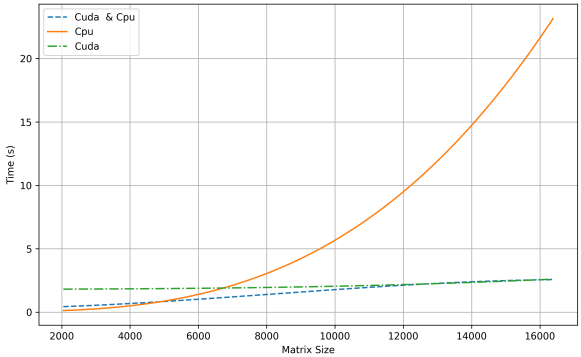


Figure 5: PRIO Scheduler.

LWS Scheduler Data Receive and Send

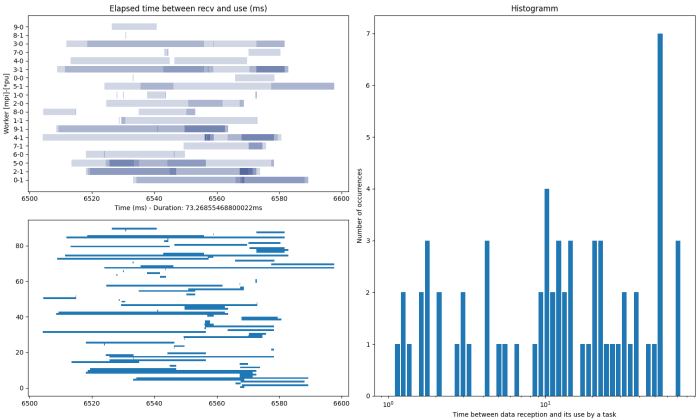


Figure 6: LWS Scheduler Data Receive.

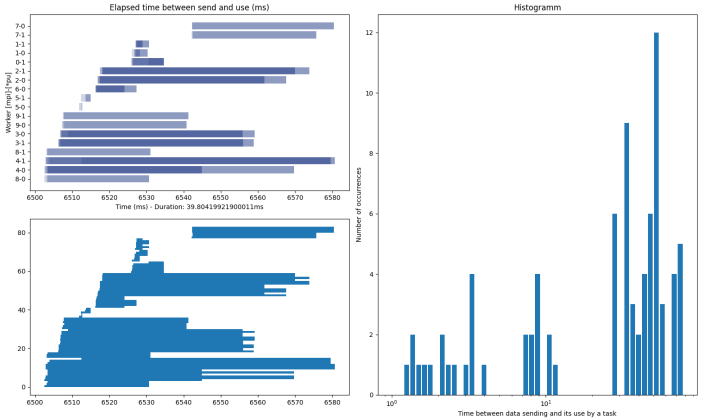


Figure 7: LWS Scheduler Data Send.

LWS Scheduler Data Receive and Send

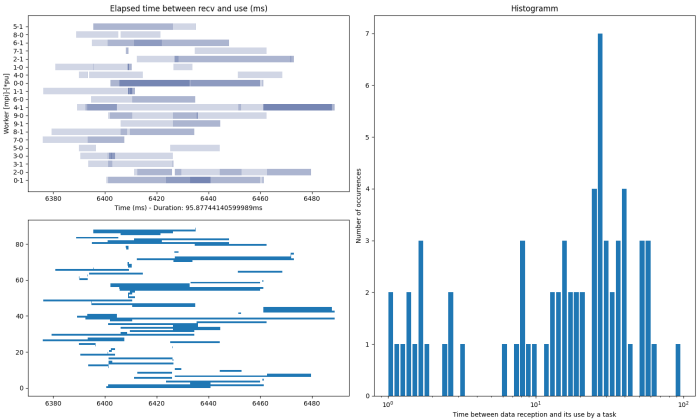


Figure 8: DMDA Scheduler Data Receive.

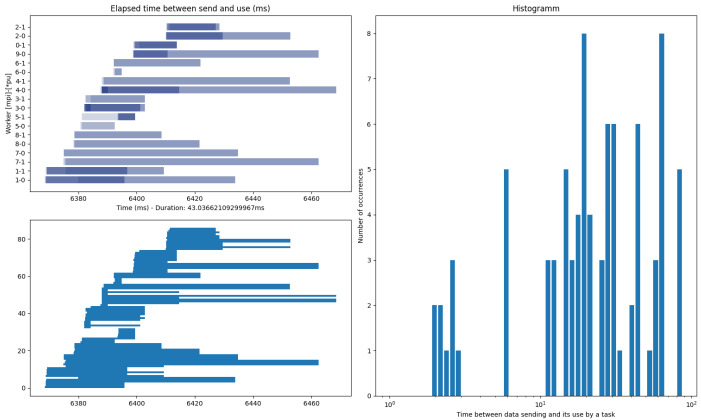


Figure 9: DMDA Scheduler Data Send.

LWS Scheduler Data Receive and Send

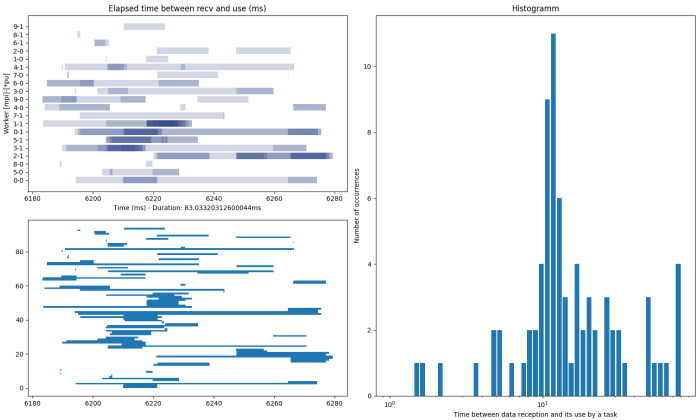


Figure 10: PRIO Scheduler Data Receive.

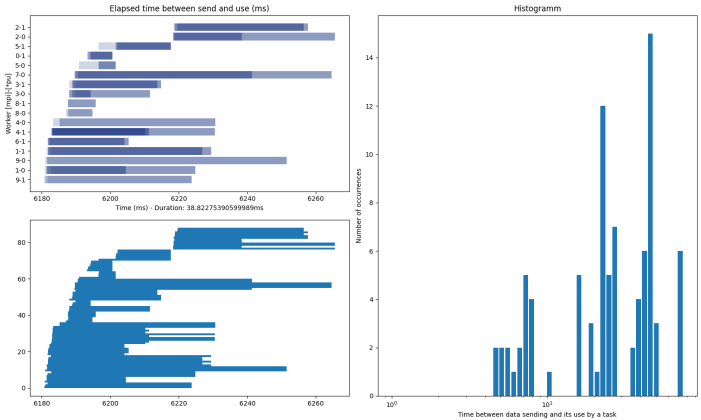


Figure 11: PRIO Scheduler Data Send.

Monitoring Activity LWS

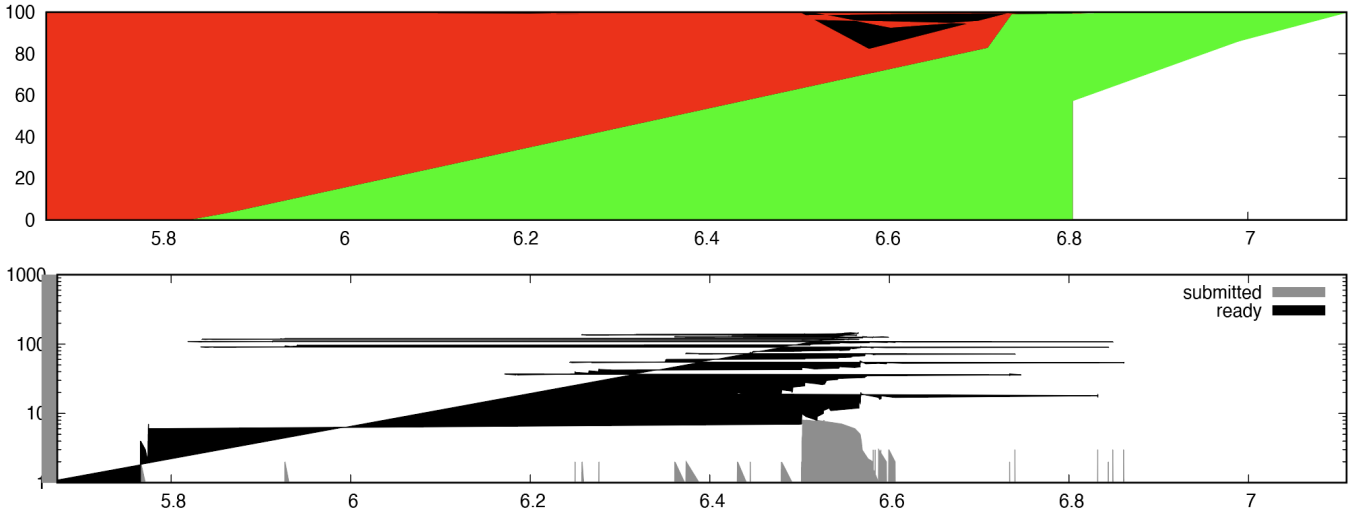


Figure 12: LWS Scheduler.

Monitoring Activity Eager

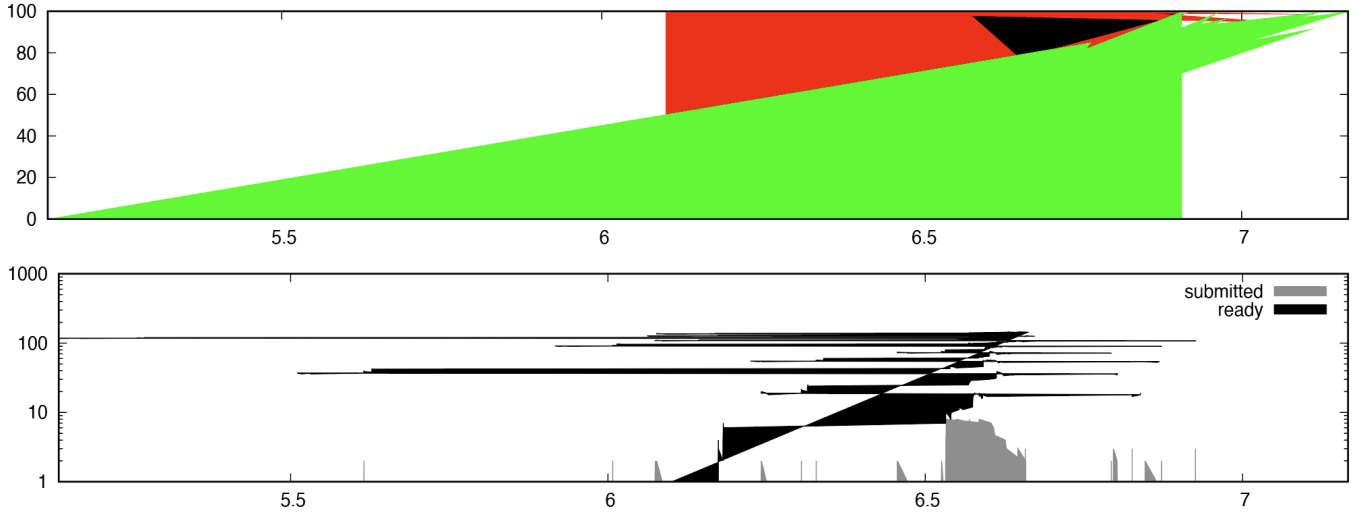


Figure 13: Eager Scheduler.

Monitoring Activity PRIO

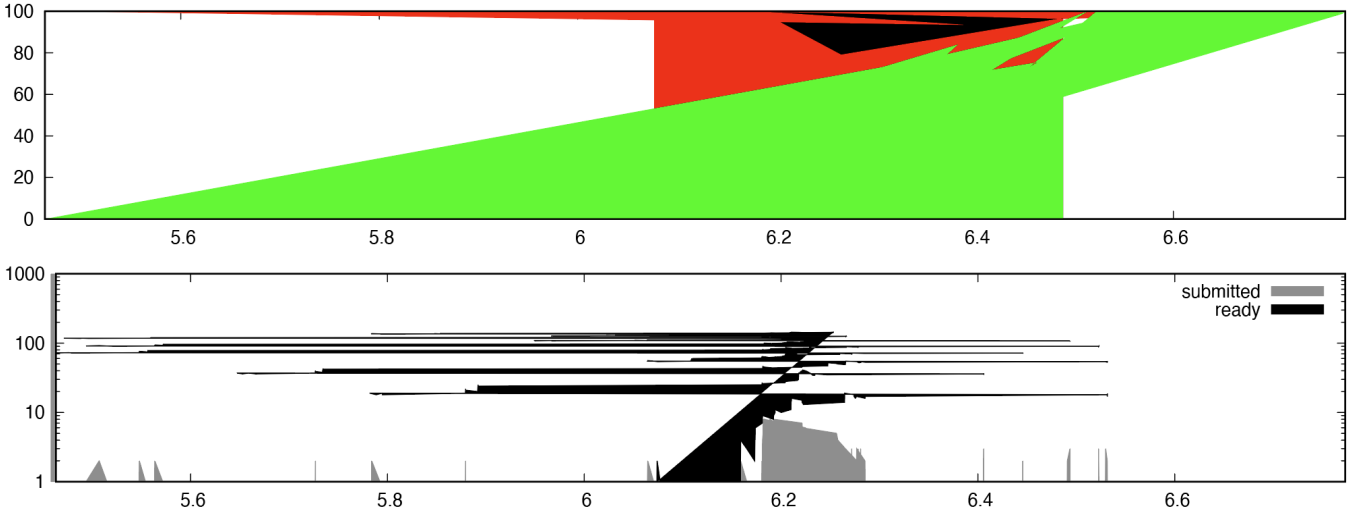


Figure 14: PRIO Scheduler.

Monitoring Activity DM

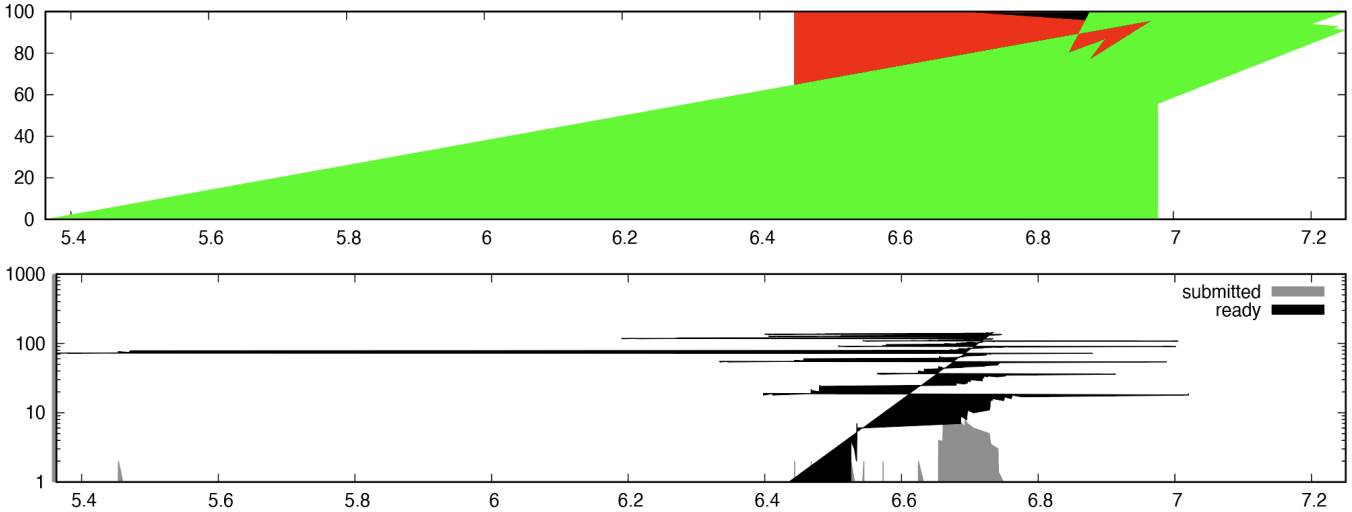


Figure 15: DM Scheduler.

Monitoring Activity DMDA

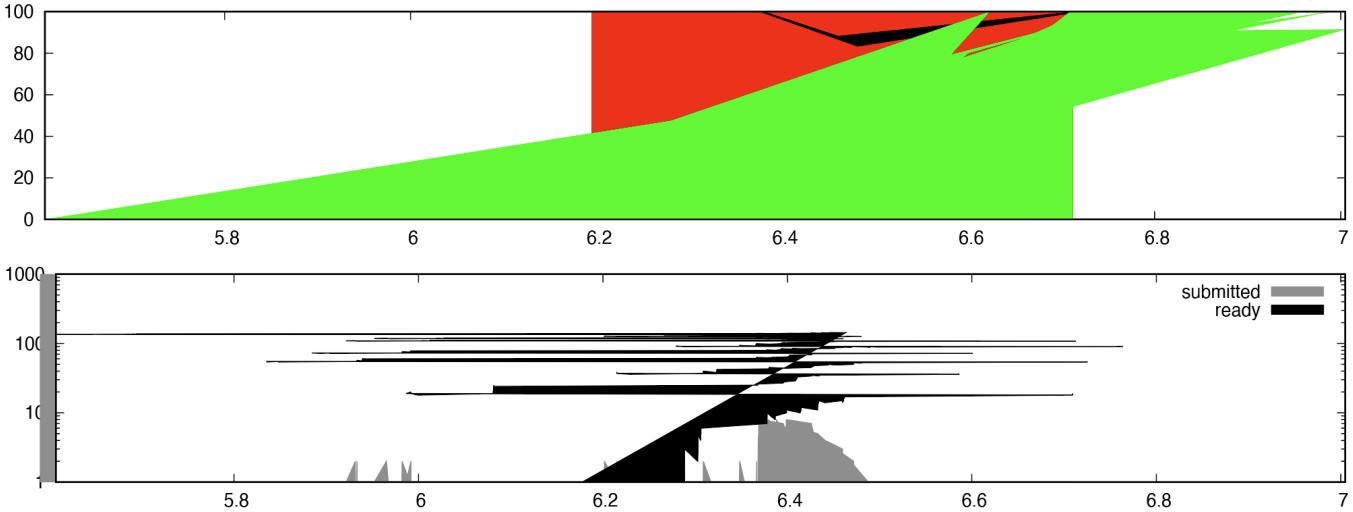


Figure 16: DMDA Scheduler.