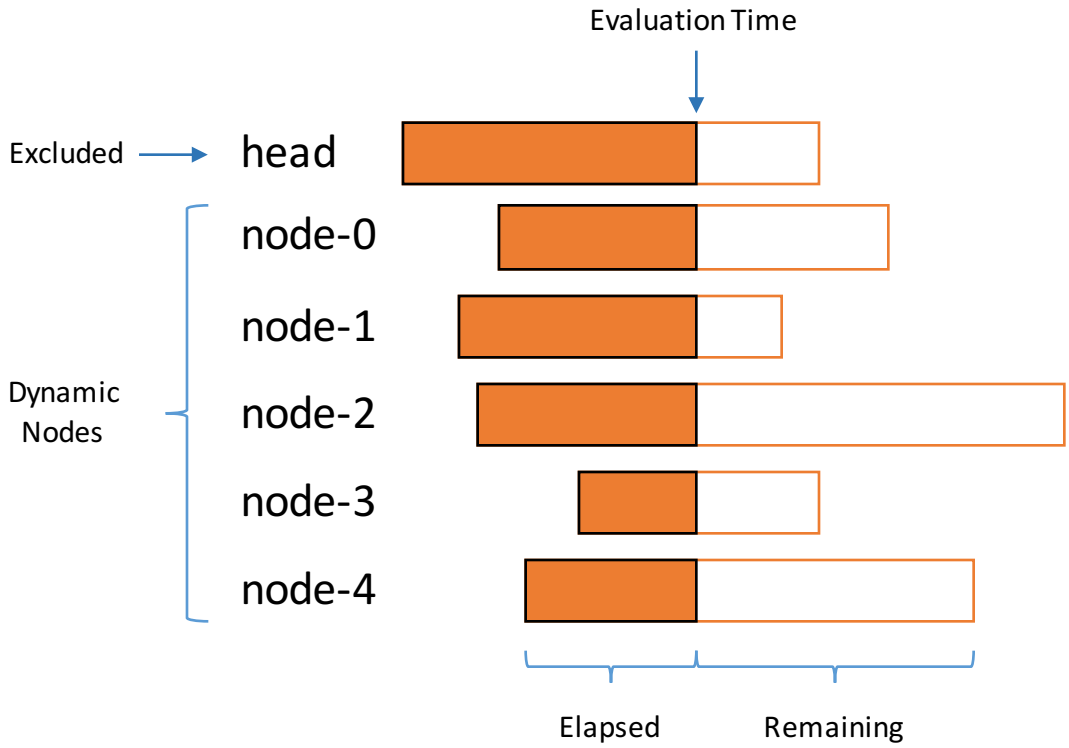
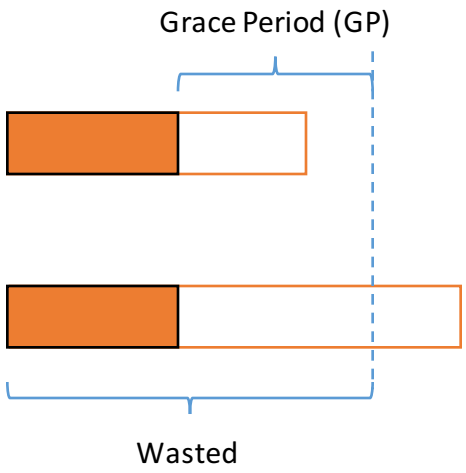
HTC (single-node jobs)



Preemption Vectors:

Policy	Random	LIFO	FIFO
Node Values	Ignore Elapsed	Sort Elapsed **Ascending**	Sort Elapsed **Descending**
Deleted Nodes	Select N nodes randomly	Select first N nodes	Select first N nodes

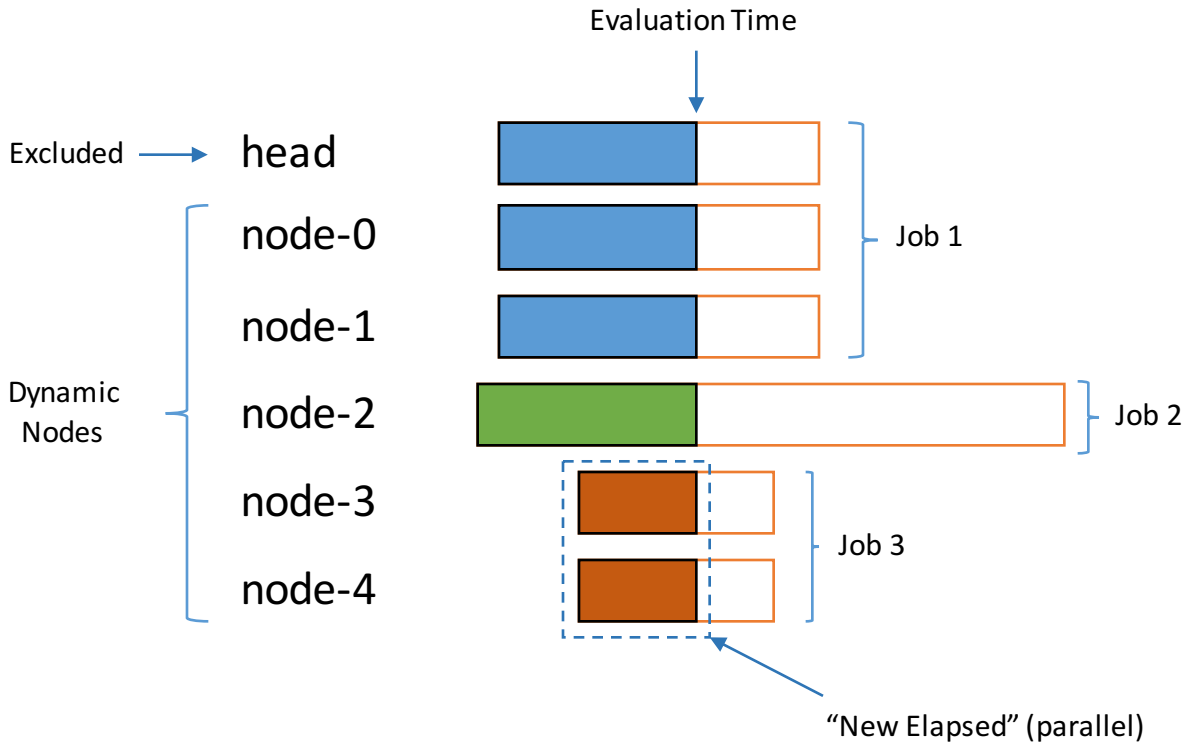
Wasted Cycles:



GP > Remaining (job will finish):
WC += 0

GP < Remaining (job will NOT finish):
WC += Elapsed + GP

HPC (multi-node jobs)

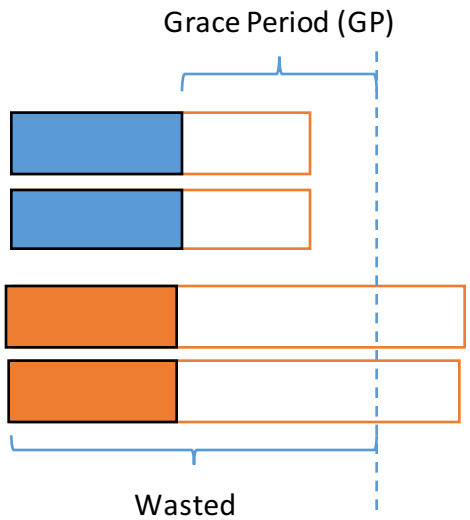


Preemption Vectors:

Random, LIFO, FIFO – same as in HTC case

Policy	PAP (Parallel-Aware Preemption)
Node Values	Sum of Elapsed for all nodes running the same job (sort ascending)
Deleted Nodes	Select first N (similar to LIFO)

Wasted Cycles:



GP > Remaining (job will finish):

WC += 0

GP < Remaining (job will NOT finish):

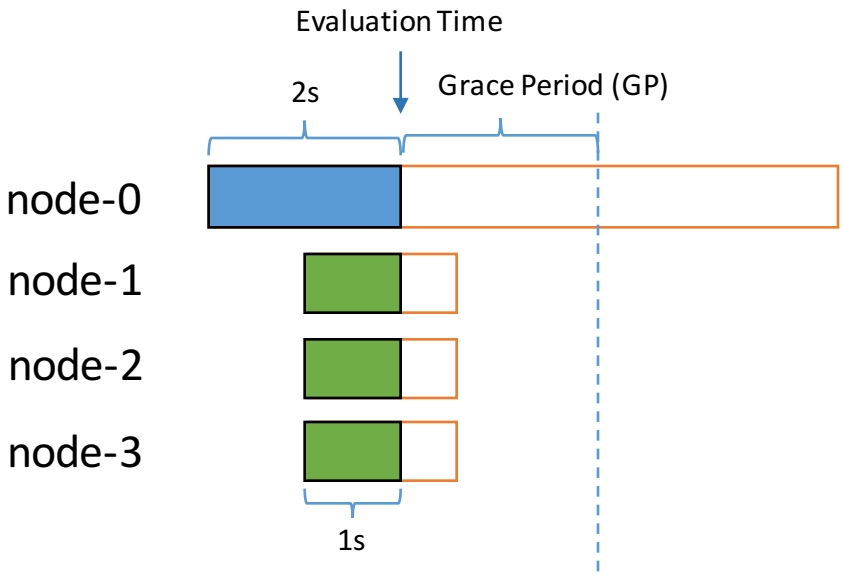
WC += (Elapsed + GP) * JN

where JN – number of node running this job

PAP versus LIFO

Assumption: PAP always performs better than LIFO because it uses information about specific nodes used by each job

Wrong. Example when it is not true:



LIFO: preserves node-0, terminates one of node-[1-3]

WC will not increase because green job will finish

PAP: preserves node-[1-3], terminates node-0

WC will increase by (2s + Grace Period) for terminating blue job