Design Document

CptS 223 Project 4
Xiyu Xie

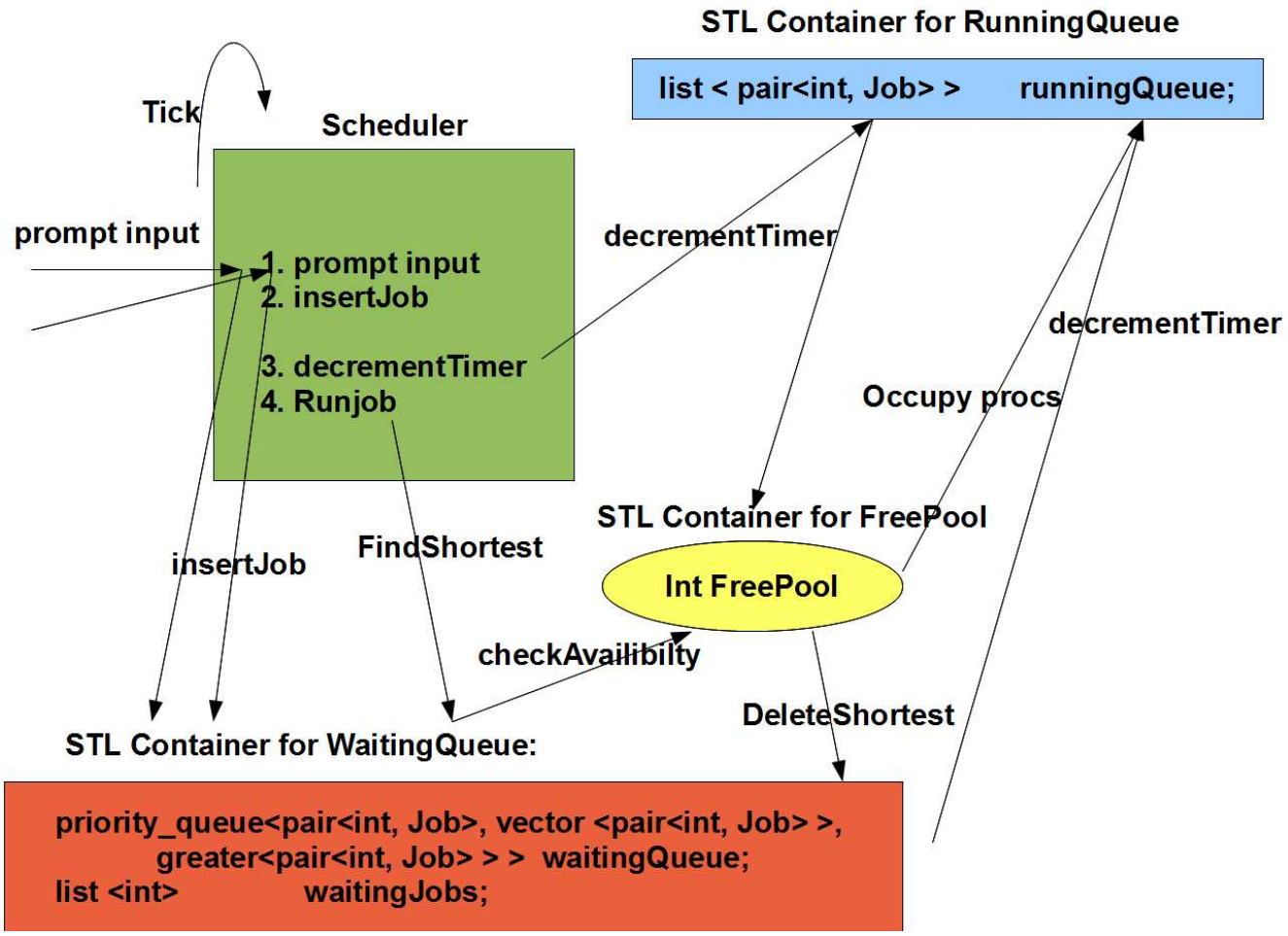

Figure 1: Organization of the Scheduler System for a parallel cluster

Table 1: Worst-Case Run-Time Complexity

	priority_queue waitingQueue	List waitingJobs	List runningQueue	Total
InsertJob	O(log n)	Θ(1) push_front		O(log n)
FindShortest	Θ(1)			Θ(1)
CheckAvailability	Θ(1)			Θ(1)
DeleteShortest	O(log n)	O(n) remove		O(n)
RunJob			Θ(1) push_front	Θ(1)
DecrementTimer && ReleaseProcs			Θ(n)	Θ(n)

3. Major Shortcomings of this Shortest-Job-first strategy

There are times when the shortest waiting job requires more processor then the ones in the freepool. That is there are times when some of the processors are not at work.

System will achieve best performance when every processors are at work at any time (except the start and end). We could maintain a priority queue for running jobs, based on the total n_ticks on this processor. Every time append new job to the processors with lowest total times. As shown in Figure 2.

No other data structure are needed.

We need to insert n_procs every time: Θ(log n)

decrementTimer is Θ(n).

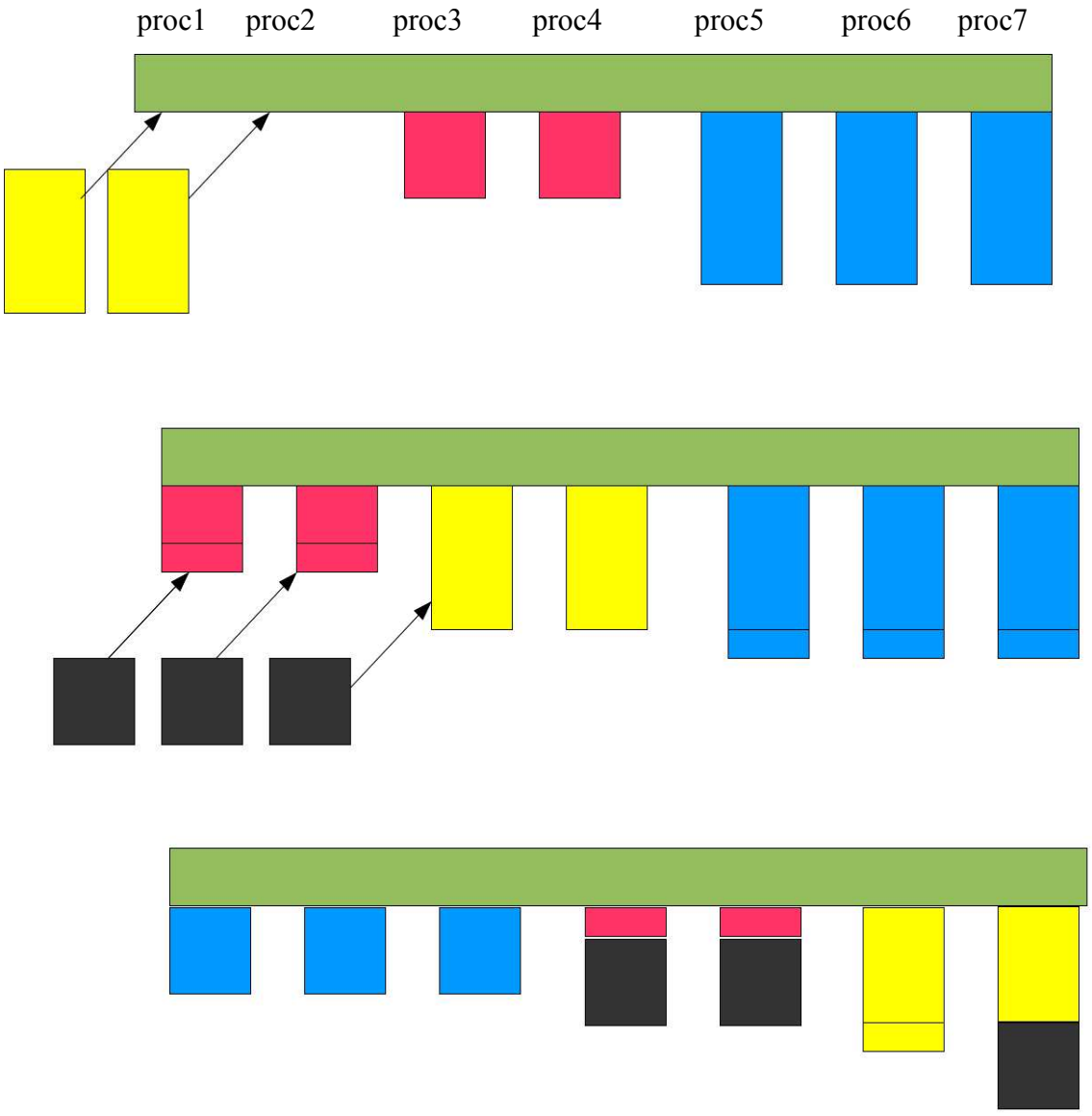

Figure 2. Priority Queue for Running Processors