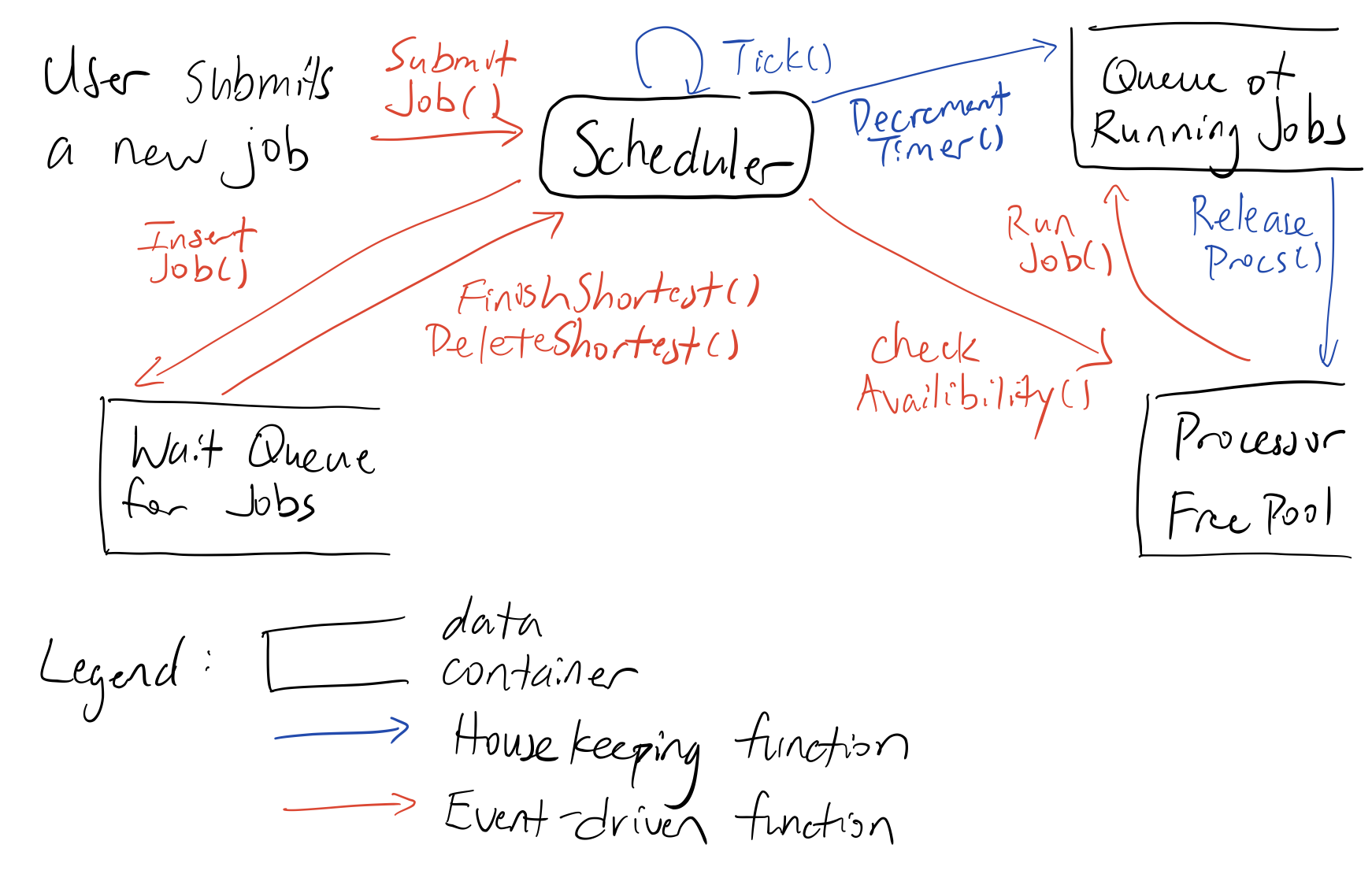
Eric Furukawa/Jimmy Zheng



|  |  |
| --- | --- |
| InsertJob | O(logn) |
| FindShortest | O(1) |
| DeleteShortest | O(1) |
| CheckAvailability | O(1) |
| RunJob | O(1) |
| DecrementTimer | O(n) |
| ReleaseProcs | O(1) |

A few shortcomings we found about this shortest-job-first strategy is that some jobs with shorts ticks might take a lot more processing power. Which in turn creates a shortage of processors for the other jobs causing a bottleneck. With no more processors in the free pool the cluster won’t be able to schedule any more jobs until the prior are finished. Another would be through the functionality point of view that this shortest-job-strategy won’t have great functionality when provided lot of short jobs, since it will never be able to move onto the jobs with more ticks if the user continues to input more jobs with less ticks.