

Analysis of Hurry-Up Scheduling Performance

Daniel Araújo de Medeiros

¹Department of Computer Science
Federal University of Bahia
Salvador, Brazil

September 2018

Definitions: Types

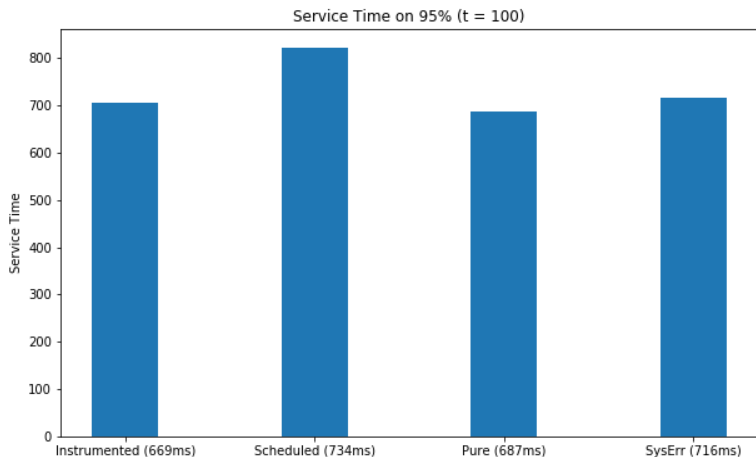
- **Pure**: The default version of Elasticsearch.
- **Instrumented**: Hurry-Up logic without up/down migrations.
- **Scheduled**: Hurry-up logic with up/down migrations.
- **SysErr**: Instrumented only with calls to SysErr (writes to RAMFS).

The used approach was:

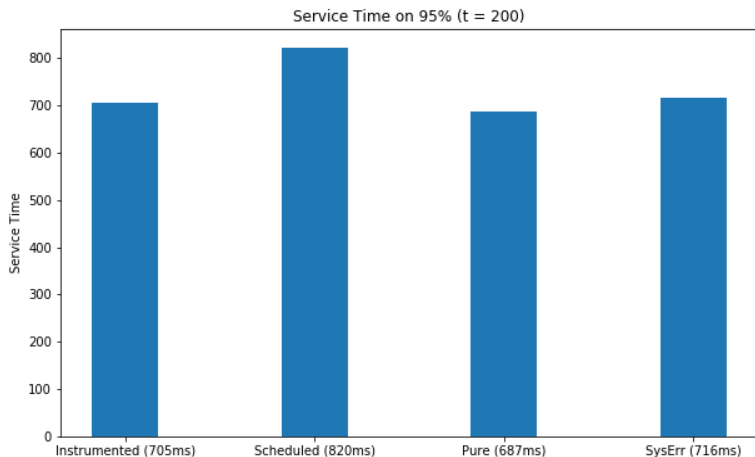
- **For SysErr:** 20.000 queries of all sizes (1-14).
- **For Pure:** 1.252 queries of all sizes (1-14). However, there's also results for low (1-4) and high (10-14) keywords length. Available on Github.
- **For Instrumented and Scheduler:** 1252 queries of all sizes. The scheduler (with and without migrations) ran every $t=100\text{ms}$ and every $t=200\text{ms}$.

1252 is the magic number as we'd get an overflow on the ringbuffer at query 1253.

Results for $t = 100\text{ms}$



Results for $t = 200\text{ms}$



- ➊ Reducing the number of sched calls also reduces the overhead generated by the scheduler. In both cases, the overhead is pretty minimal as Instrumented vs Pure has a difference of less than 5%.
- ➋ However, there's an associated cost to the migration - the "Scheduled" case is bigger than "Instrumented" in both ($t=100$, $t=200$) cases.
- ➌ SysErr has a bigger service time due to the more frequent number of activations (every in-out of functions, which has more frequent calls than Instrumented version).

Conclusions: Modifications to the new Scheduler

- 1 The instrumentation (e.g, SysErr) has an impact according to the number of calls. An (atomic) event-driven approach - which calls only when necessary - may reduce even more the overhead.
- 2 Up/down migrations introduces an overhead even when the migration time is done at 100ms or 200ms. This means that the scheduler must still keep the migrations to a minimum (hence the event-driven logic) and try to find a way to make an interaction between Scheduler - Juno without the Operating System layer.