10000 to 1

(tasks per second)

Ben Clifford
benc@hawaga.org.uk

Baseline A: my laptop, local threads without logging

10188

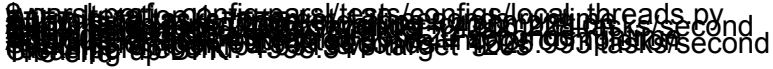
tasks/sec

Baseline B: NERSC Perlmutter with Work Queue

0.63

tasks/sec

Variant L: my laptop, local threads, default logging on



~~10188~~

4000

tasks/sec

Variant R: perlmutter Work Queue, 1 core per task

```
$ parsl-perf --resources '{"cores":1,
"memory":0, "disk":0}' …
```

(2.8x speedup on a host with 256 cores…)

~~0.62~~

1.68

tasks/sec

Variant C: Work Queue coprocesses

Persistent Python workers

== Task Vine "serverless" mode

~~1.68~~

400

tasks/sec

Variant M: Monitoring

~~400~~

basic monitoring

300

tasks/sec

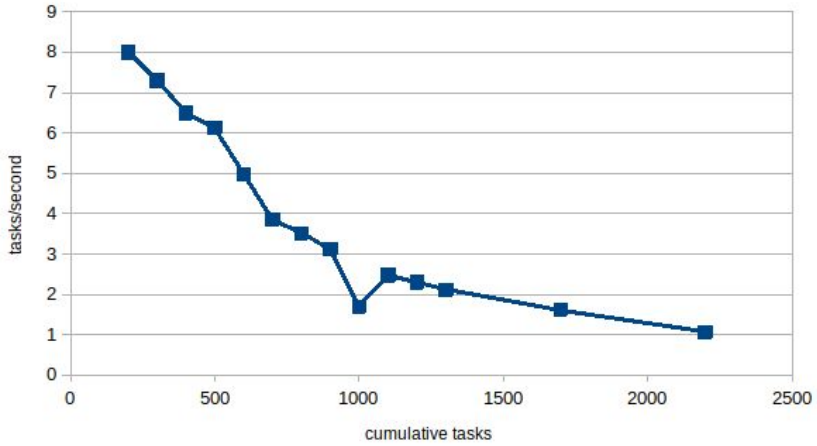
serialization
process forking
network message delivery

in-task resource monitoring

11

tasks/sec

Variant S: my serialization mess up

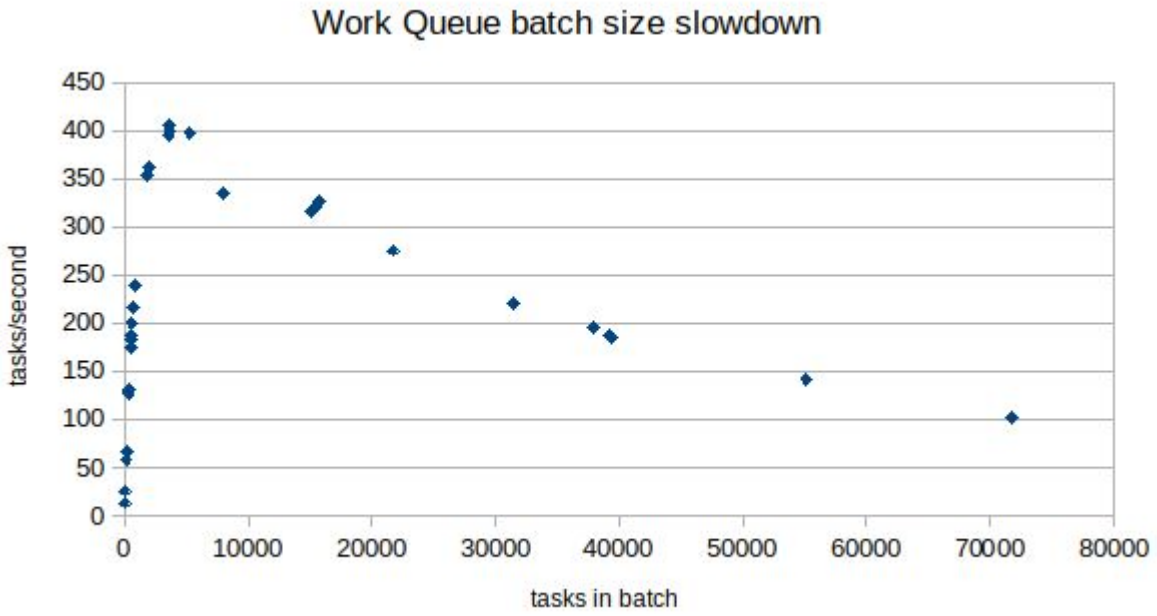


~~11~~

1

tasks/sec

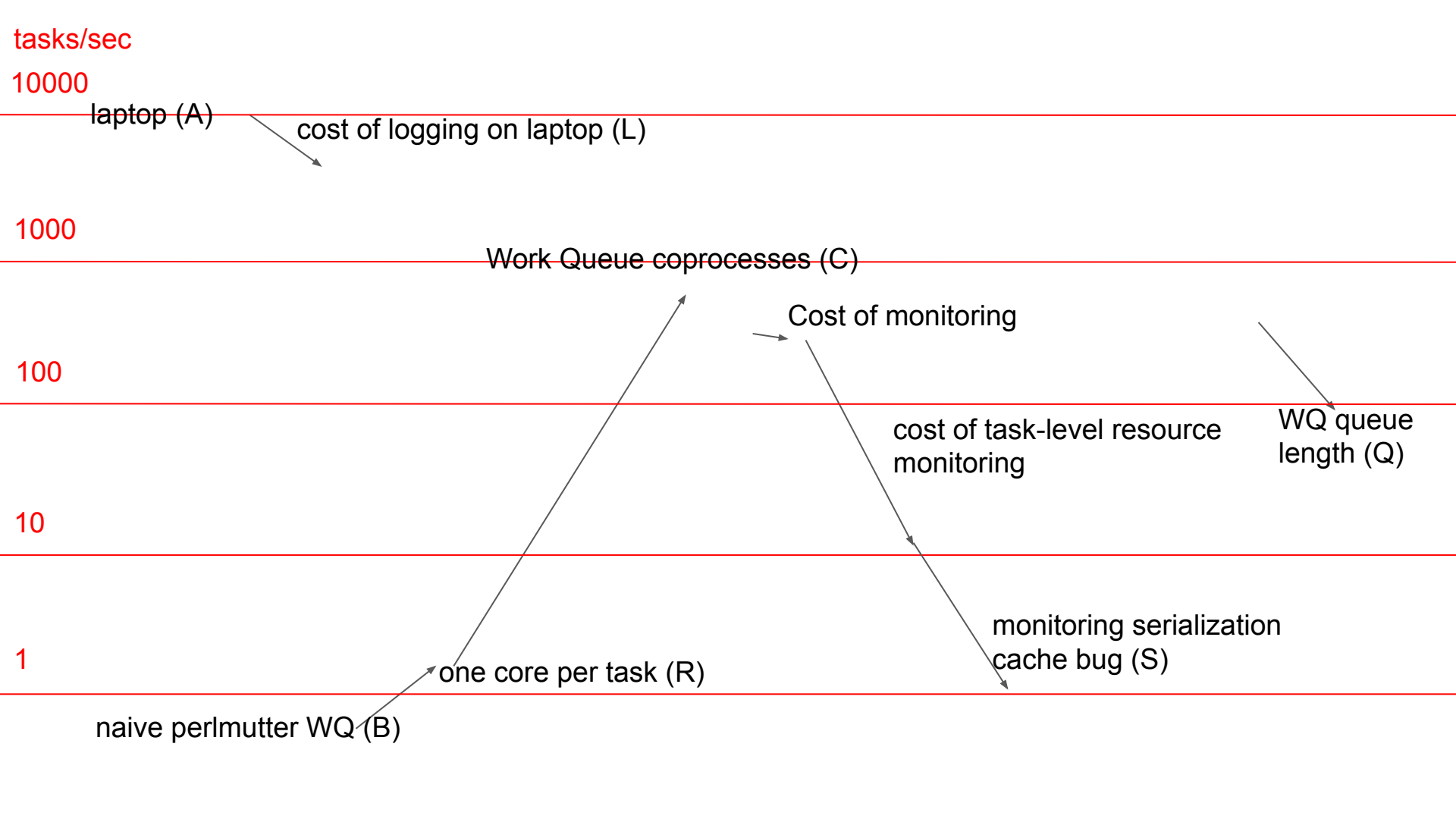
Variant Q: Work Queue queue length slow down



~~400~~

100

tasks/sec


tasks/sec
10000
laptop (A)
cost of logging on laptop (L)
1000
Work Queue coprocesses (C)
Cost of monitoring
100
cost of task-level resource monitoring
WQ queue length (Q)
10
monitoring serialization cache bug (S)
1
one core per task (R)
naive perlmutter WQ (B)