Performance Prediction for Coarse-Grained Locking

Vitaly Aksenov, ITMO University and INRIA Paris
Dan Alistarh, IST Austria
Petr Kuznetsov, Telecom ParisTech

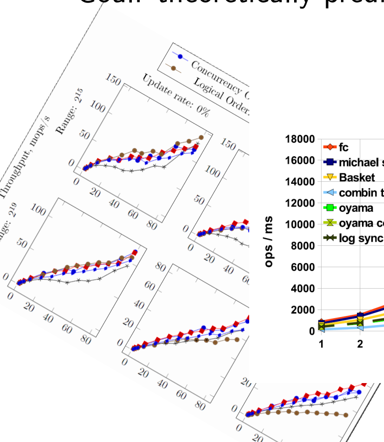
PODC 2018

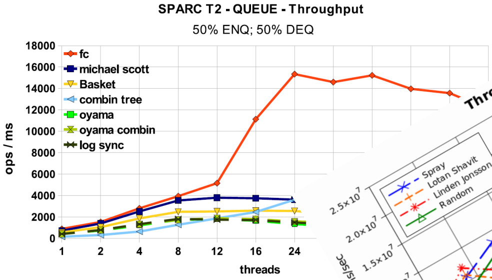
Measure Efficiency

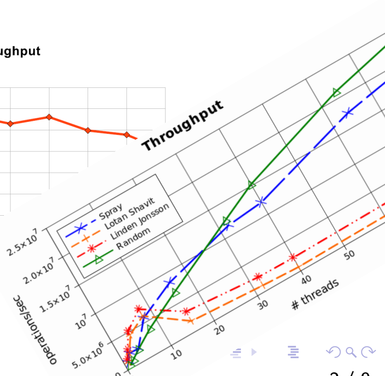
How to measure efficiency:

- Theoretically: Number of steps, Number of messages
- Experimentally: Throughput, Latency

Goal: theoretically predict Throughput!







Coarse-Grained Data Structure

Simple but non-trivial class. E.g., hash-tables.

```
operation():
  lock.lock()
  for i in 1..C:
    nop
  unlock.lock()
  for i in 1..P:
    nop
```

Assumptions

- Variant of MESI Protocol (`Modified`, `Shared`, `Exclusive` and `Invalid`)
- One-Layer Symmetric Cache: W_{st} and R_{st} (in cycles)
- Intel Xeon Machine. $W = W_M = W_S = W_E = W_I$ and uncontended swap takes W [David et al., 2013]
- Uniform Scheduler: at each unit of time each process makes a step
- CLH Lock [Craig, 1993]

CLH-based Coarse-Grained Data Structure

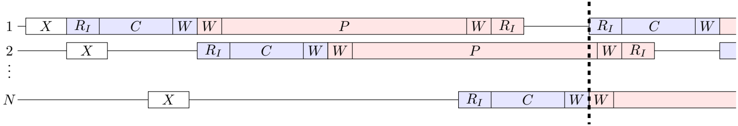
```
global Node head ← new Node()
local Node my_node ← new Node()

operation():
  Node next ← swap(&head, my_node)     // W or X
  while (next.locked) {}          // R_l or 2·R_l
  for i in 1..C:                  // C
    nop
  my_node.locked ← false          // W
  my_node ← next
  my_node.locked ← true           // W
  for i in 1..P:
    nop
```

Simple Analysis. Throughput. Nonempty Queue

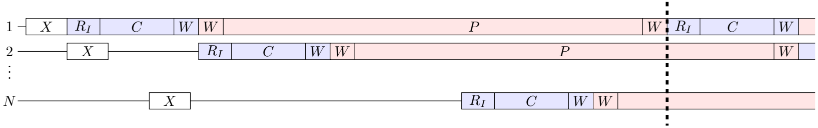
1. There is always somebody in the queue

$$P \leq (N-1) \cdot W \text{ then } \frac{\alpha}{R_I + C + W} \approx \frac{\alpha}{C}$$



2. The queue is always empty

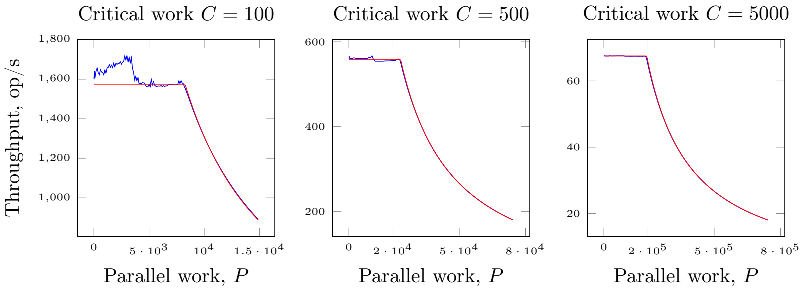
$$P \geq (N-1) \cdot W \text{ then } \frac{\alpha N}{(W+P+W)+(R_I+C+W)} \approx \frac{\alpha N}{P+C}$$



Results

40 processors = 4 chips Intel Xeon $\times$ 10 cores.

$$\alpha = 3.5 \cdot 10^5,\ W \approx 40,\ R_l \approx 80$$



Future Work

- Generic analysis — different scheduler
- Generic computational model
- Another lock implementations
- More complex types of programs