

ASL Exercise 5

Measuring a baseline

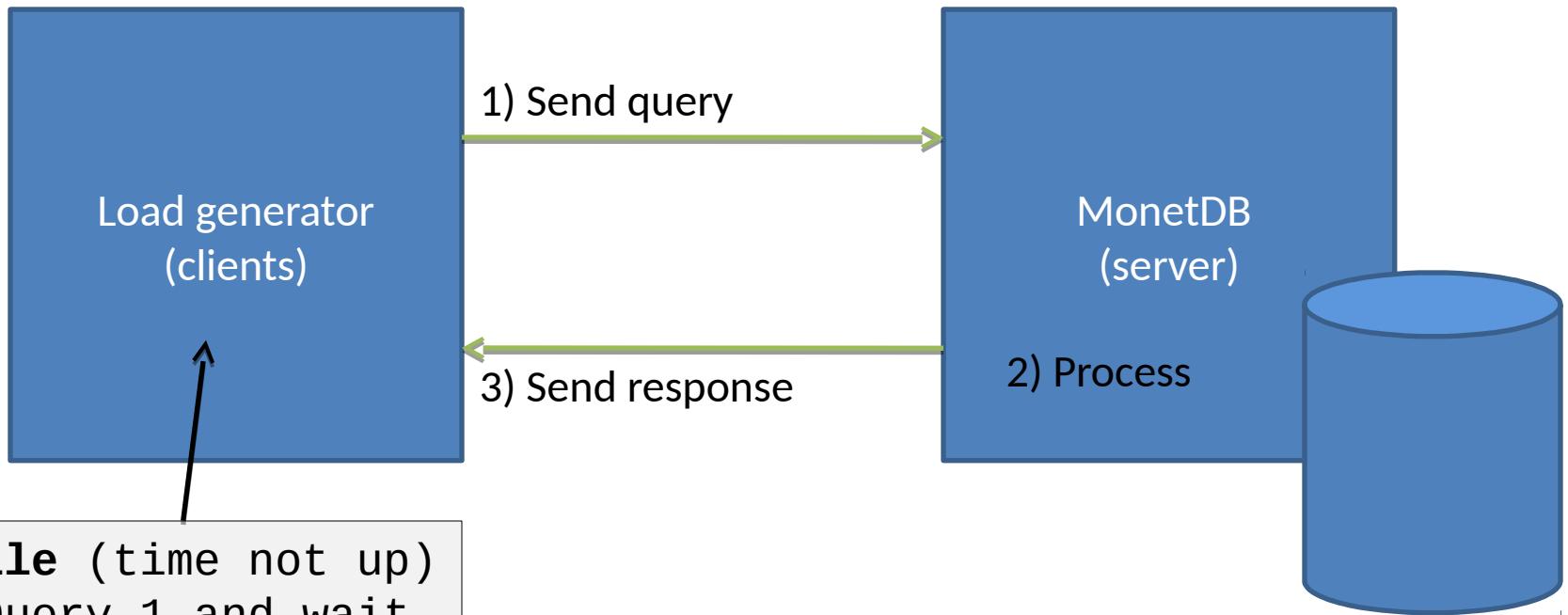
System under test

- MonetDB
 - Main-memory database (reads data off disk, keeps it in memory)
 - Best for analytical queries (no updates, etc.)
- Our clients
 - Simple scripts that run three types of queries from TPC-H
 - Log the output and runtime of each query

Testing methodology

- Experiment length
 - 4 minutes (without warmup and cooldown)
 - 3 repetitions
- Number of clients
 - Between 1 and 20
- MonetDB
 - Different multi-threading setups
- Machines
 - Two physical machines
 - 16 cores each

Testing setup




```
while (time not up)
  Query 1 and wait
  Query 2 and wait
  Query 3 and wait
end
```

$$T_{\text{network}} \ll T_{\text{processing}}$$

Collecting results

- mtX/ - X threads in MonetDB
 - pY/ - Y parallel clients
 - cZ/ - Data output by client number Z
 - rW/ - Repetition number W
- dump.dat



```
...
+-----+-----+-----+-----+
| supp_nation      | cust_nation      | l_year | revenue      |
+-----+-----+-----+-----+
| FRANCE           | GERMANY           | 1995   | 481994.2376  |
| FRANCE           | GERMANY           | 1996   | 629797.4221  |
| GERMANY          | FRANCE            | 1995   | 581829.6063  |
| GERMANY          | FRANCE            | 1996   | 642056.6361  |
+-----+-----+-----+-----+
4 tuples (639.316ms)
...
```

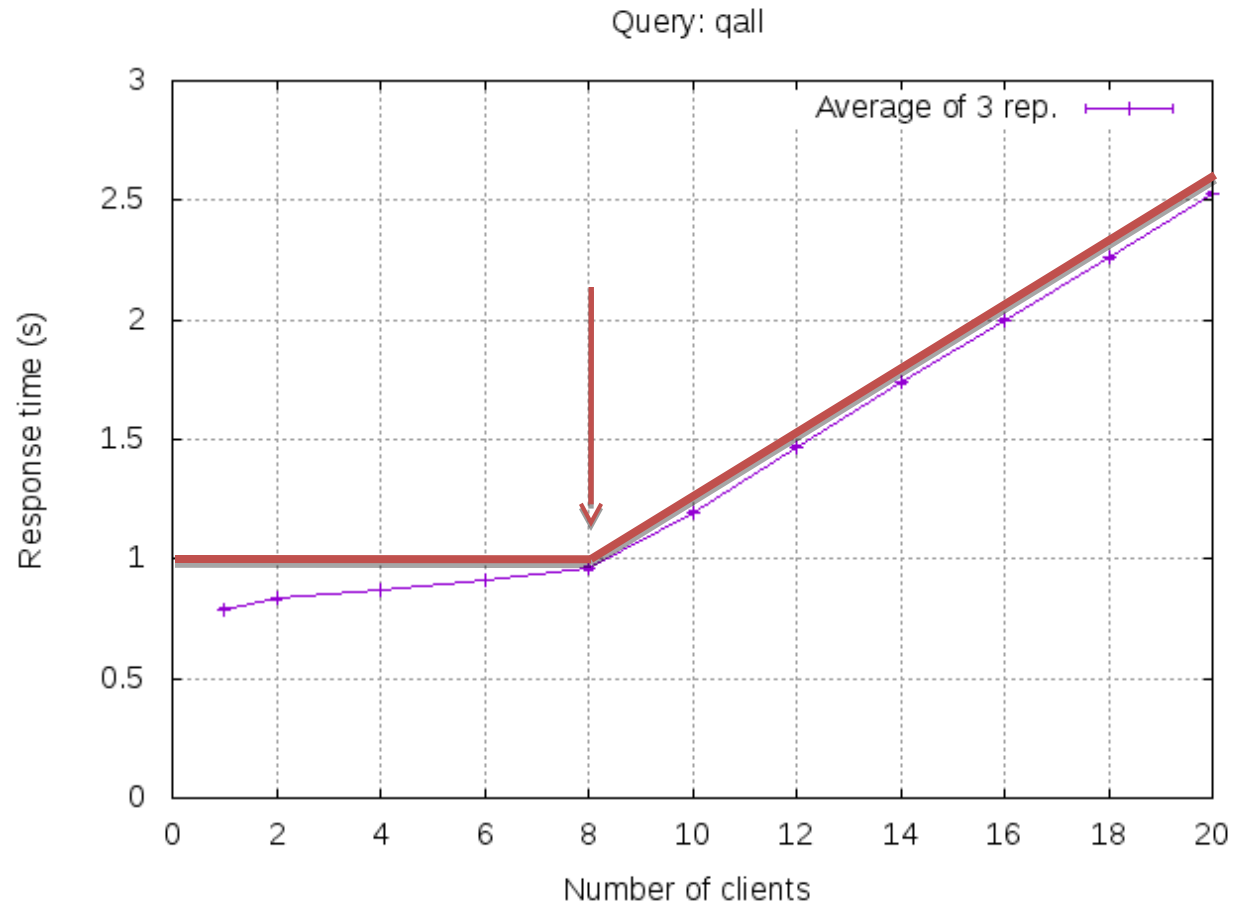
supp_nation	cust_nation	l_year	revenue
FRANCE	GERMANY	1995	481994.2376
FRANCE	GERMANY	1996	629797.4221
GERMANY	FRANCE	1995	581829.6063
GERMANY	FRANCE	1996	642056.6361

4 tuples (639.316ms)

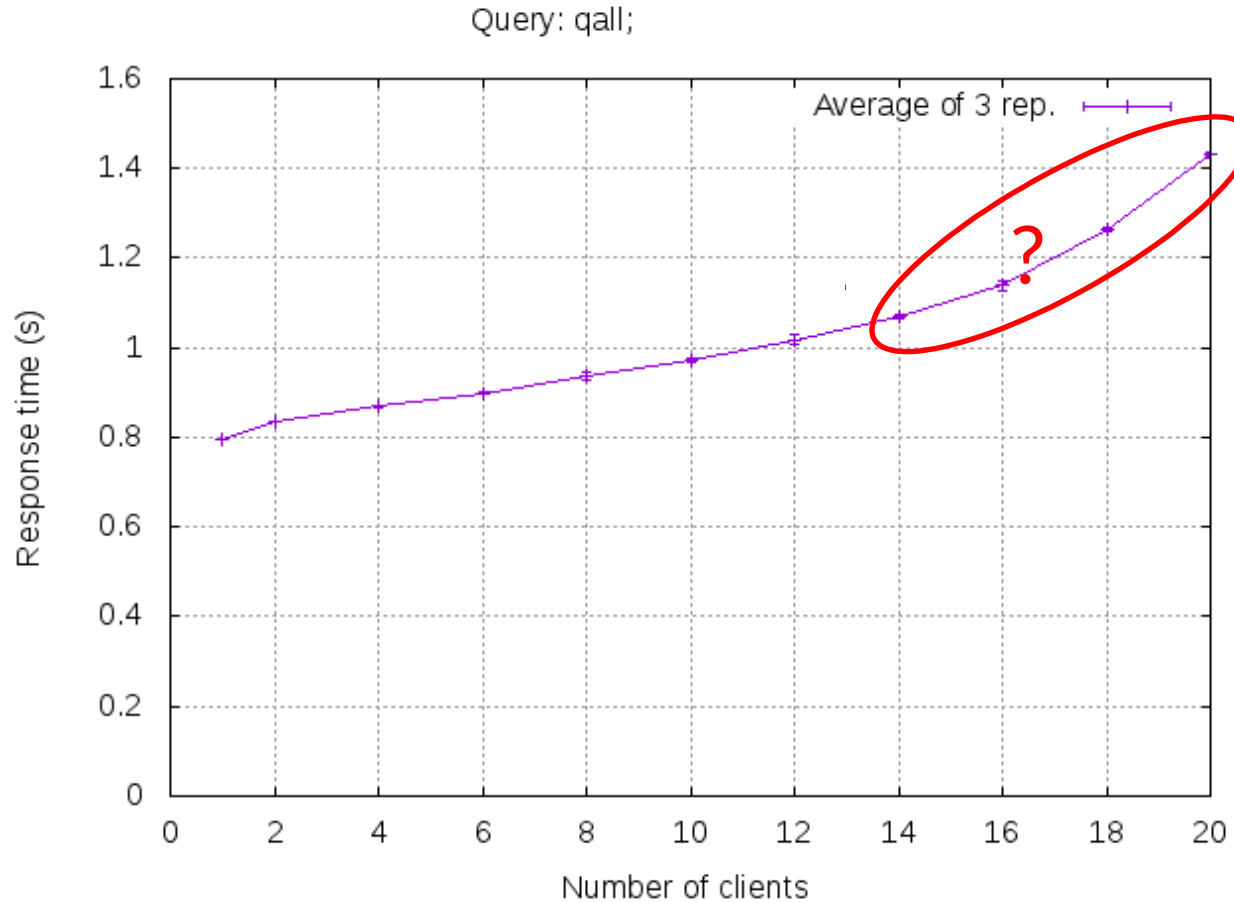
Script to transform results

- Response times
 - Collect all response times per repetition
 - Collect response times per query type per repetition
 - Compute average and standard deviation
- Throughput
 - Count returned queries from all clients / 4 minutes
 - Compute average and standard deviation

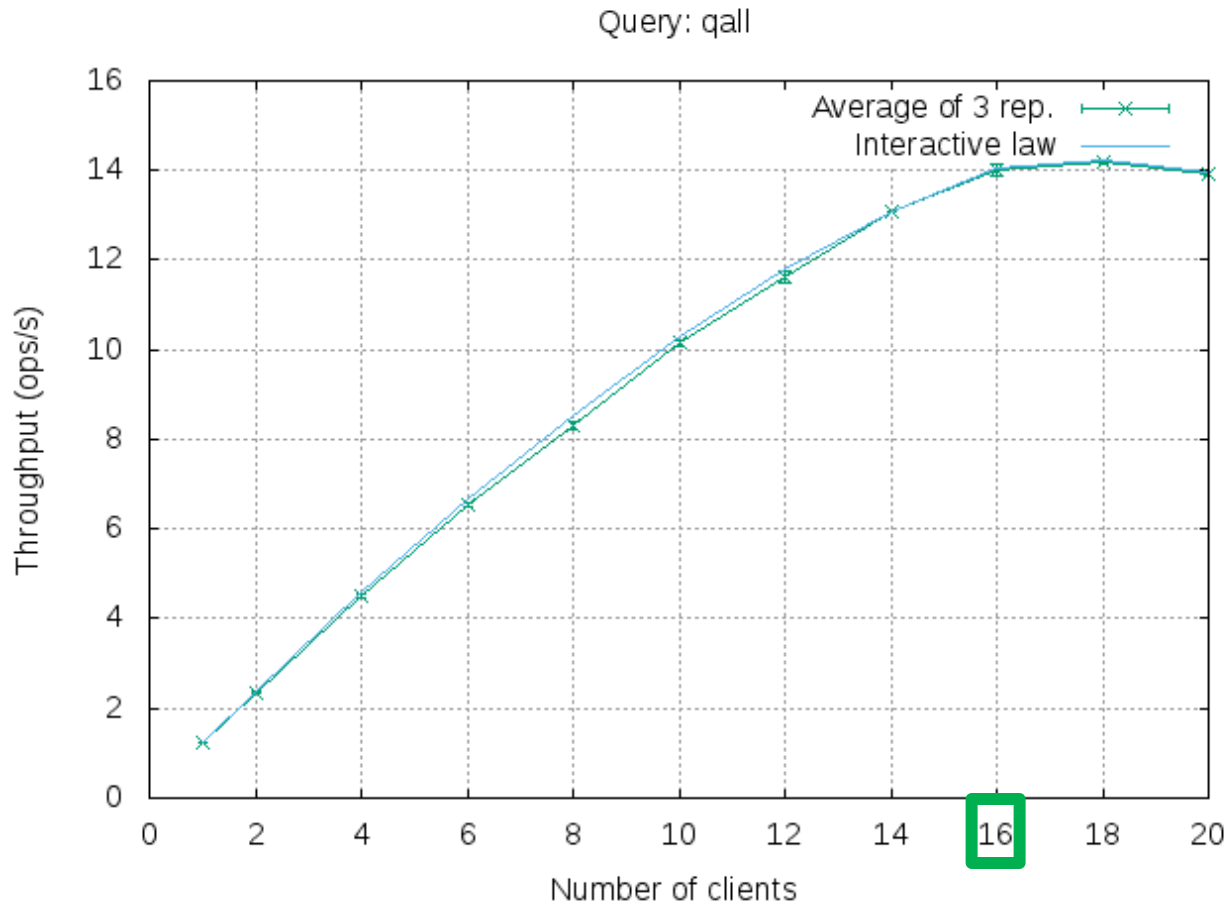
How many threads in MonetDB?



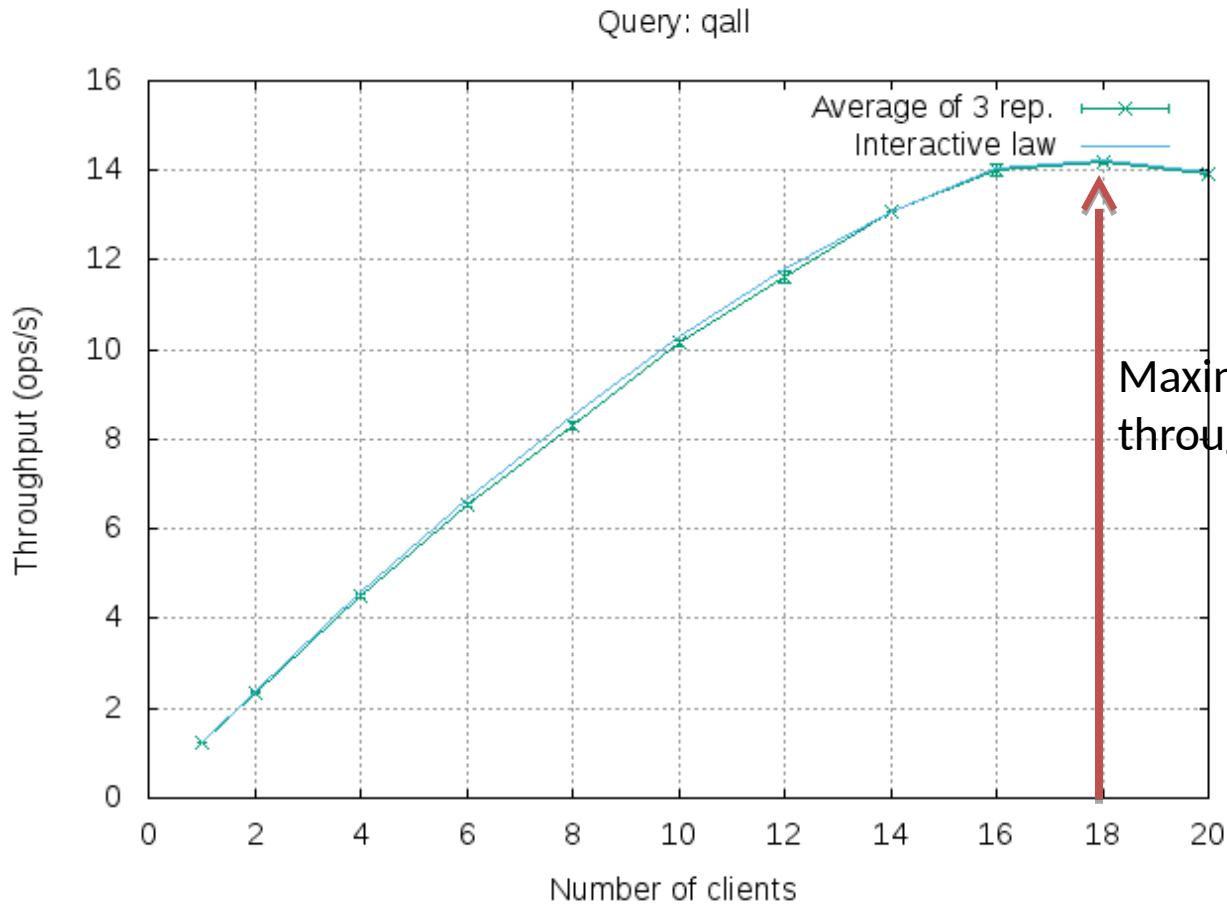
And now?



Maybe if we look at the TPUT graph?

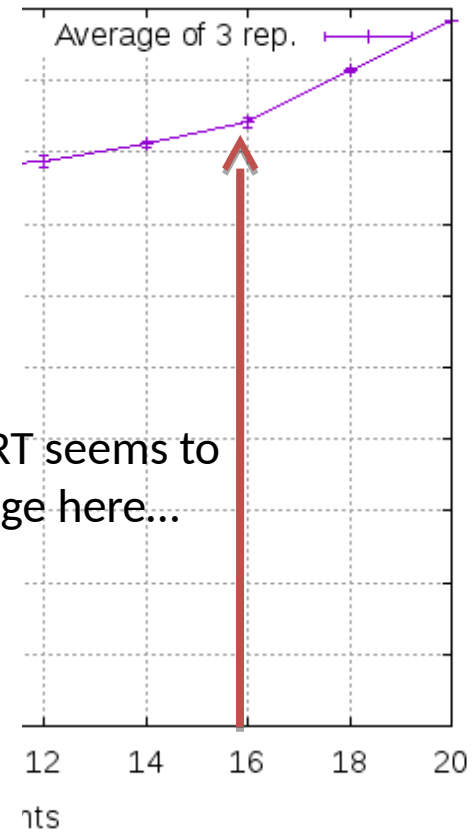


Maximum throughput?

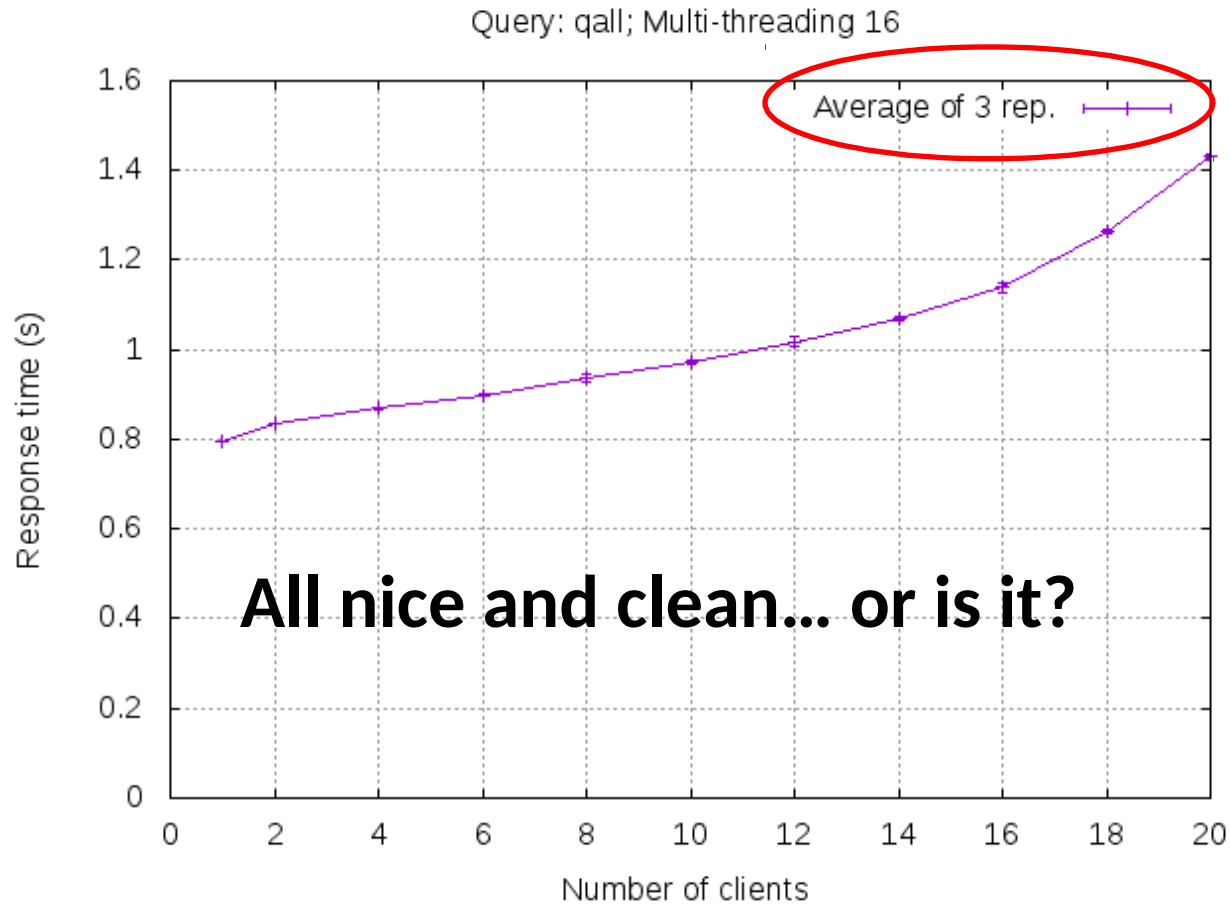


Maximum throughput?

But RT seems to change here...

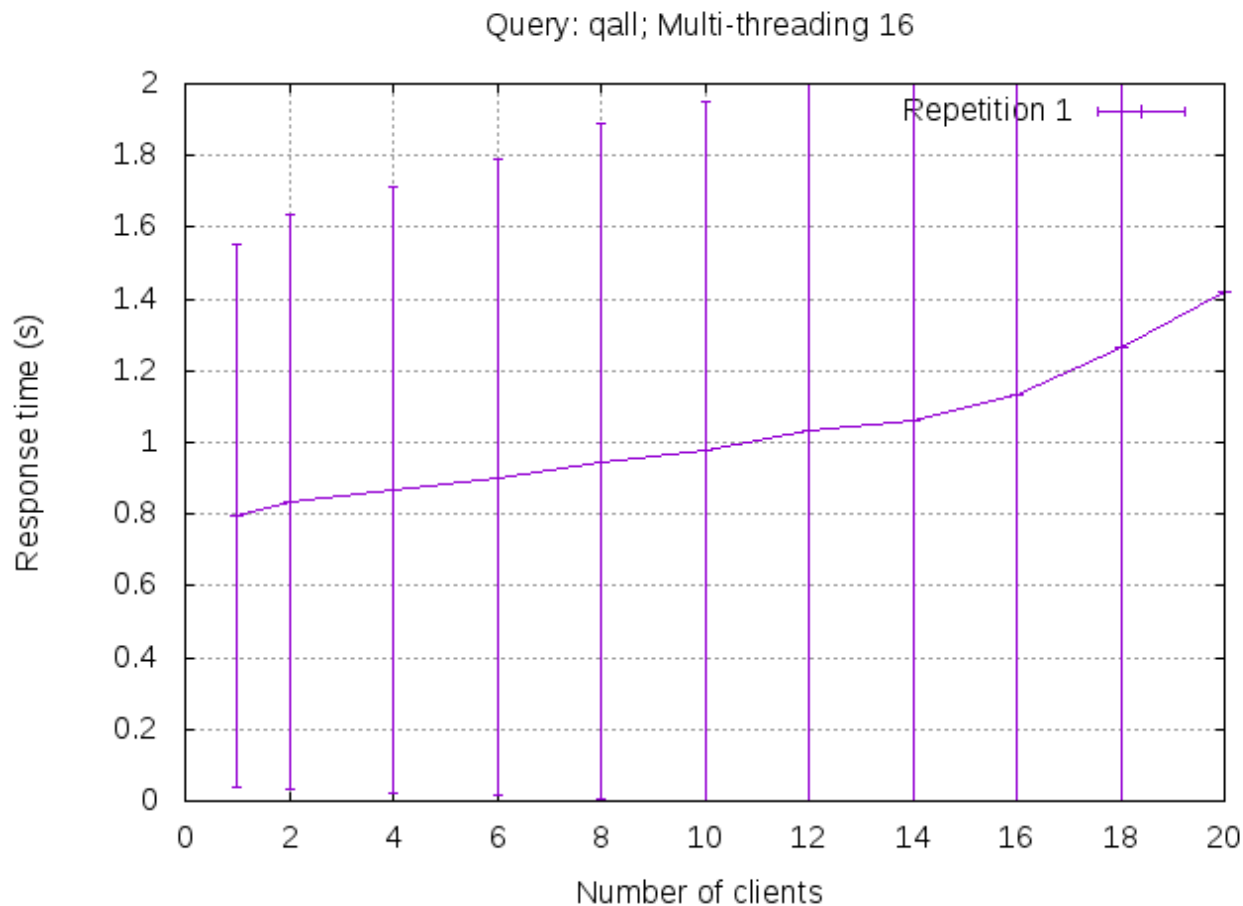


A word on standard deviation



A word on standard deviation (II)

Let's see **AVG** and **STDEV** inside a repetition...



What happened to our STDEV?

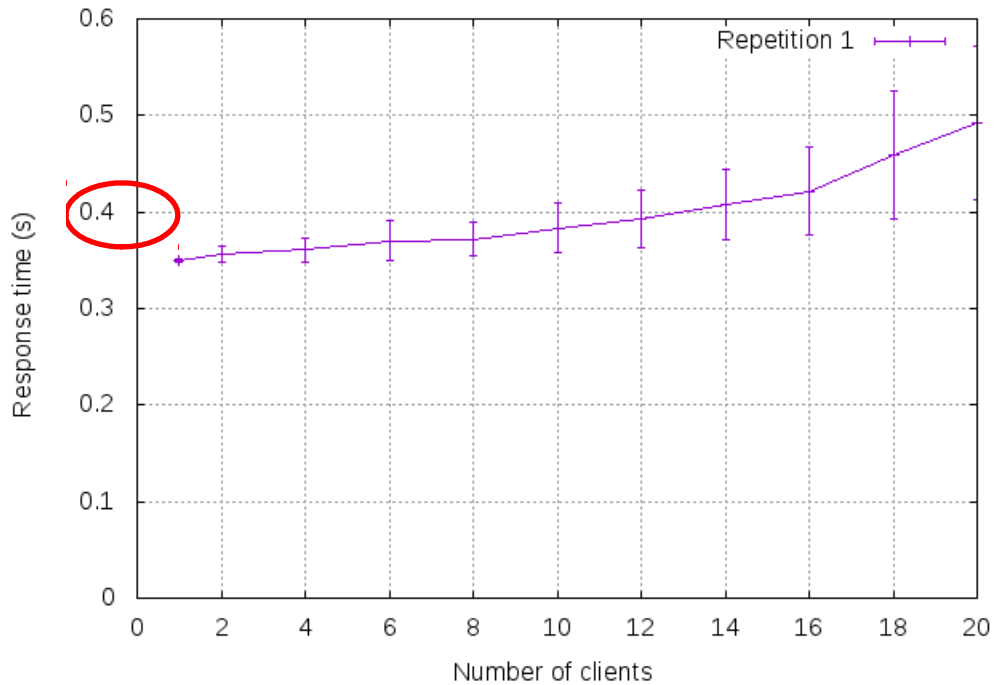
- What could be the reason?

What happened to our STDEV?

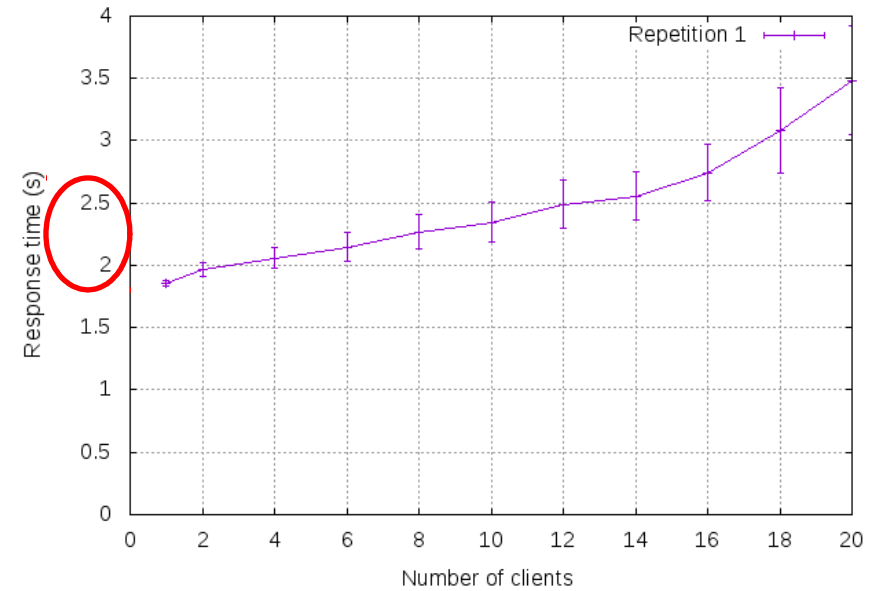
- What could be the reason?
- Recall: Each client sends three types of queries...
- What if their response times are different?

RT by Query type

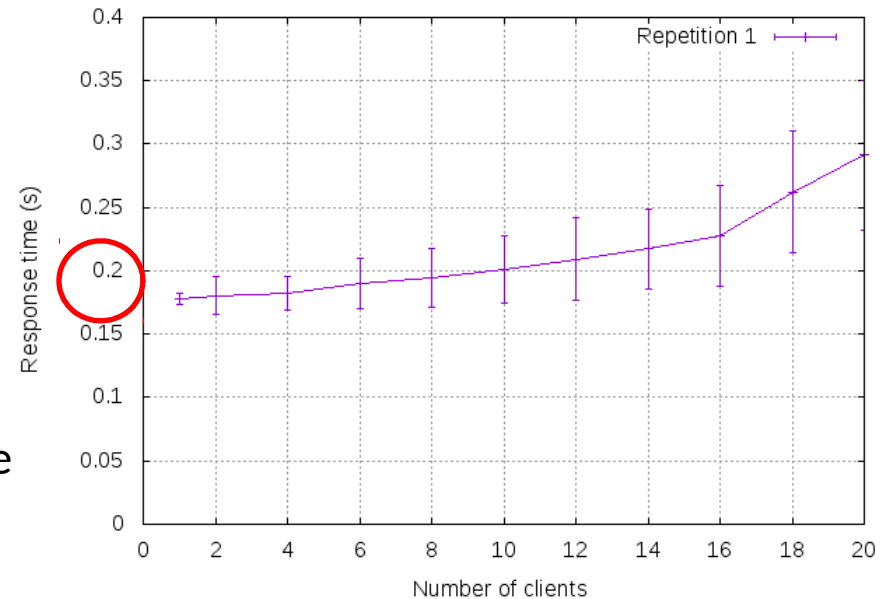
Query: q1; Multi-threading 16



Query: q2; Multi-threading 16



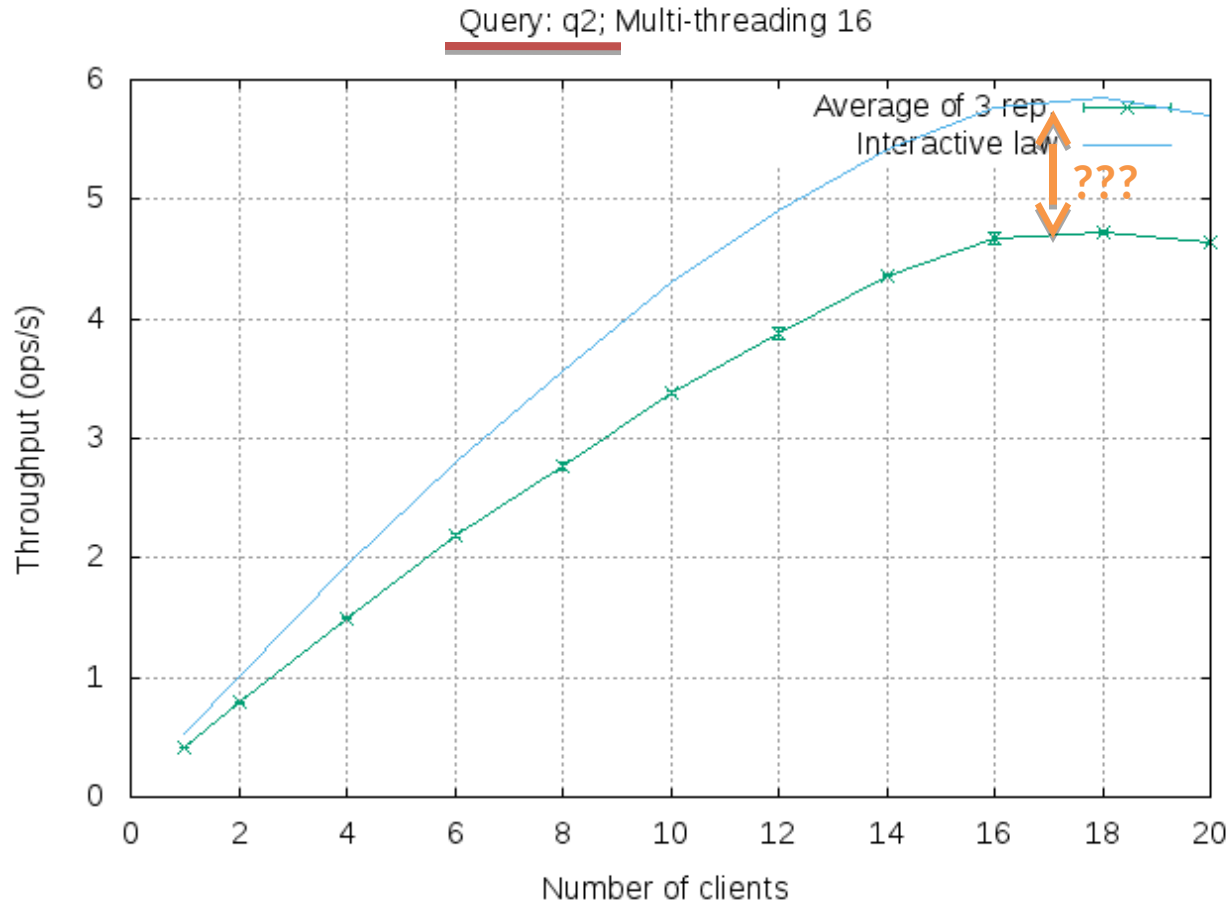
Query: q3; Multi-threading 16



If we average over three classes of queries, the standard deviation is going to be high

Interactive law

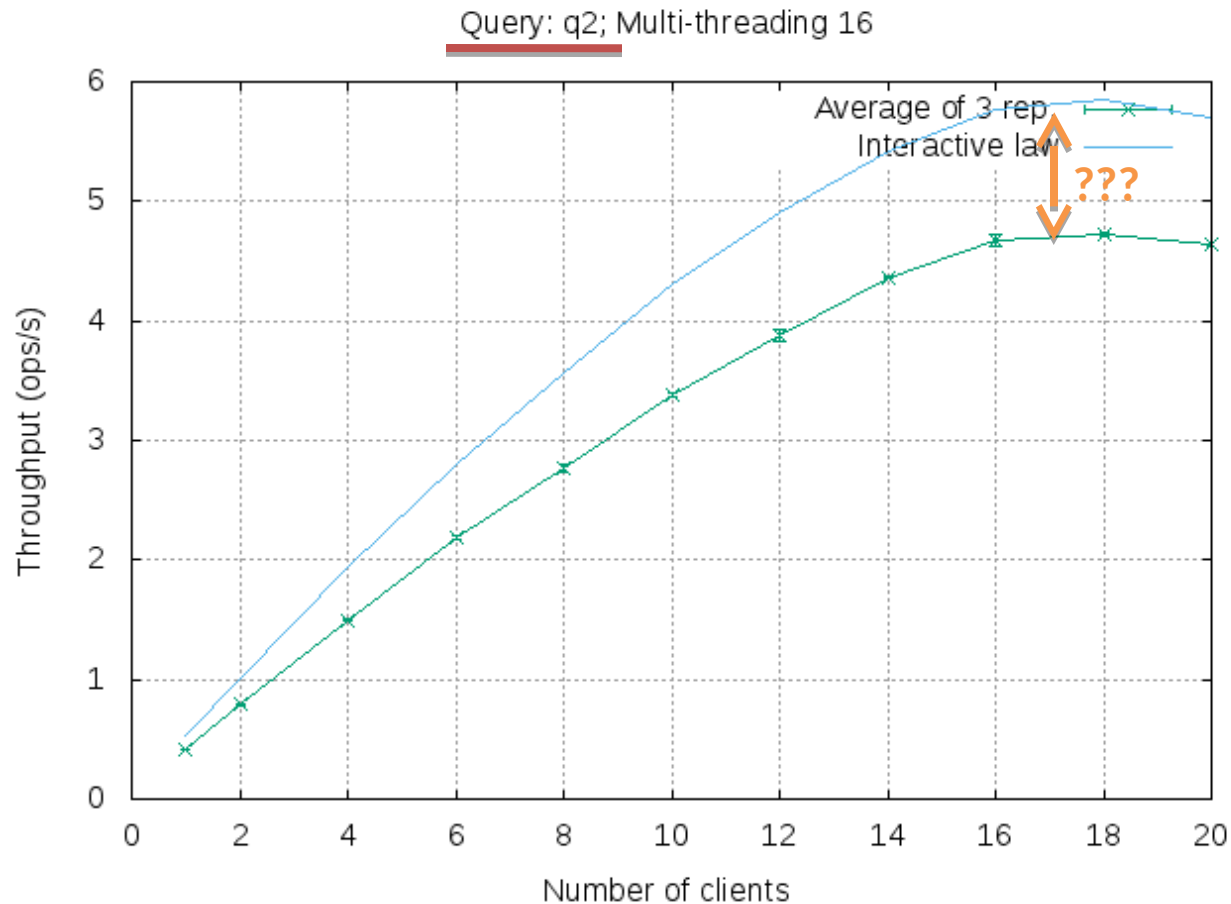
$$\text{TPUT} = 1/(\text{RT}+\text{Z}) * \#\text{Clients}$$



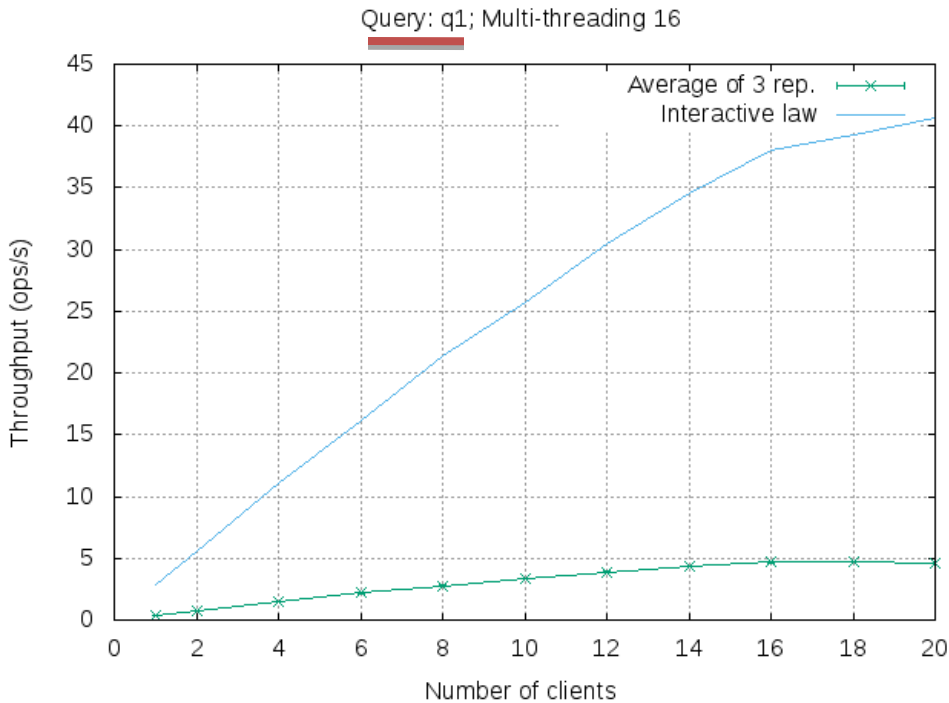
Interactive law (II)

$$\text{TPUT} = 1/(\text{RT}+\text{Z}) * \#\text{Clients}$$

- But we don't only run Q2!
- If we would run Q2 alone we would get the blue line. The green line is 1/3 the global throughput!

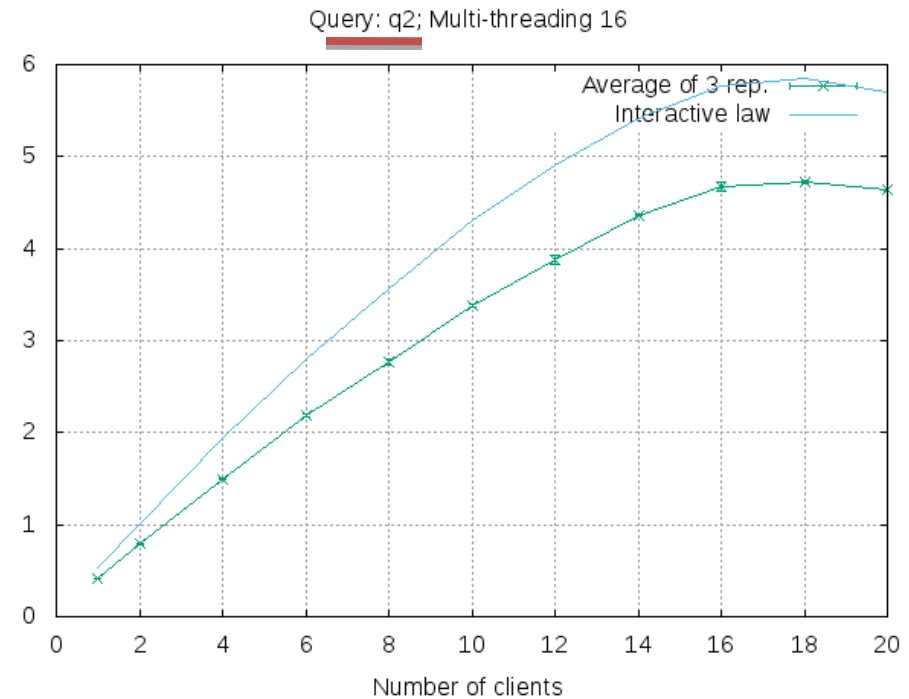


Interactive law (III)



Response time ~ 0.4s at 16 clients

Actual throughput ~5 at 16 clients



Response time ~ 2.75s at 16 clients

Actual throughput ~5 at 16 clients

Because system throughput (Q1+Q2+Q3) ~14 at 16 clients

Plotting Best Practices

- Start axis at zero, try and keep same range for related graphs
- Label both axis, state units clearly
 - Use Ops/s not Ops/minute, and other “exotic” units
 - Instead of 12000000 use 1.2 million
- Caution with logarithmic scales on axis
- Include error bars!
- Make sure system configuration is easily found