# ECE 454
# Computer Systems Programming
## *Definition of Performance*

Ding Yuan
ECE Dept., University of Toronto
http://www.eecg.toronto.edu/~yuan

slides courtesy: Greg Steffan

## Before we go on…

- What do you exactly mean by "*performance*"?
  - Simple program: speed -- how fast your program runs
  - Unix "time" command
- Server program
  - Is "speed" the only important thing?
  - What is the "speed" for long running programs?
  - *Latency* vs. *throughput*

# Latency vs. throughput

- Latency
  - How fast the server respond my request?
    - Sometimes also called response time
- Throughput
  - Number of requests served/unit time
- Relationship?

# Positive correlation example

```
void dummy_server () {
  while (request = next_request ()) {
    respond (request);
  }
}
```

Latency for req. 1

Latency for req. 2

Latency for req. 3

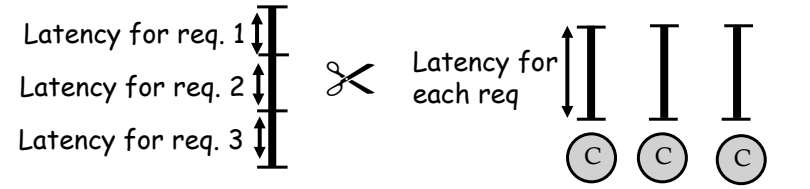
$$\text{Throughput} = \frac{3}{L1 + L2 + L3} = \text{req./sec.}$$

*If we have a faster CPU, both latency and throughput will improve (smaller latency, higher throughput)!*

# Negative correlation example

```
void dummy_server () {
  while (request = next_request ()) {
     thread_create(respond, request);
  }
}
```

*Latency will be worse (lower), why?*
*Throughput will be better (higher), why?*




*Before parallelization*

*After parallelization*

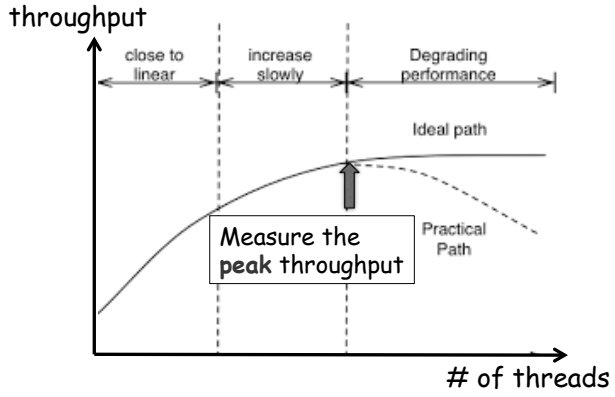
# Real life analogy

| Plane | Toronto to Paris | Speed | Passenger | Throughput (pmph) |
|---|---|---|---|---|
| Boeing 747 | 8 hours | 610 mph | 470 | 286,700 |
| Concorde | 4 hours | 1350 mph | 132 | 178,200 |

Which plane has higher performance?

# Parallelism vs. Throughput

- Will more *parallelism* always improve *throughput*?




# Performance measurement is a very complicated problem

- Other metrics: bandwidth, jitter, etc.
- Extra considerations: best case? worst case? average?
- Different applications have different requirements
  - Netflix
  - Google/Facebook/Amazon
  - Online gaming
- ACM special interest group on performance evaluation (SIGMETRICS)