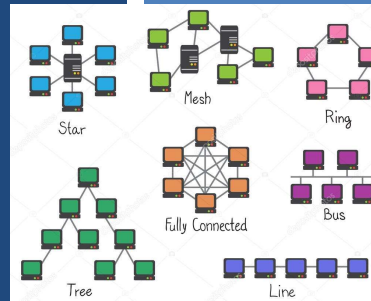


# TCSS 558: APPLIED DISTRIBUTED COMPUTING

## Virtualization, Clients and Servers

Wes J. Lloyd  
Institute of Technology  
University of Washington - Tacoma



**Don't Forget to Terminate (Shutdown)  
all EC2 instances for Assignment #0**

**Auction based instances:  
3 x m4.large instances @ ~3 cents / hour**

**\$2.16 / day  
\$15.12 / week  
\$64.80 / month**

**AWS CREDITS → → → → → → → →**



## OBJECTIVES

- In class Quiz. . .
- Assignment 0
- Assignment 1
- Feedback from 10/19
- Midterm exam on Thursday Nov. 2nd

October 24, 2017

TCCS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L8.3

## FEEDBACK – 10/19

- **What are the advantages and disadvantages of the many-to-one threading model?**
  - Operating system schedules only processes
    - Threads are created and managed with user code (not OS code)
    - User controls which threads have context (run) at any given time
    - Operating system schedules the entire process to run or wait
    - Thread context switching doesn't enter kernel (protected) mode
      - *Generally considered faster*
  - Application execution is all-or-nothing
    - Either **all** or **no** threads of an application can run
    - Contrast to one-to-one programming model: Fine grained elements of a process (a few threads) run while OS timeshares with many other apps
    - How many processes can the OS run at any given time ?

October 24, 2017

TCCS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L8.4

## FEEDBACK - 2

### ■ What are the advantages and disadvantages of the one-to-one threading model?

- Operating system schedules processes and threads
  - Threads are created and managed with kernel code (OS code)
  - Kernel controls which threads have context (run) at any given time
  - OS schedules individual threads and processes to run or wait
  - Thread context switching enters kernel (protected) mode
    - *Generally considered slower...*
- Application execution can be fine-grained
  - Any number of threads of an application can run at any time
  - Contrast to many-to-one programming model: All-or-nothing process execution - OS may timeshare with fewer total apps
  - How many processes/threads can the OS schedule?

```
top - 01:34:40 up 4
Threads: 1402 total
```

October 24, 2017

TCCS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L8.5

## FEEDBACK - 3

### ■ What are super peers?

- A node in a peer-to-peer network that operates as a server to a set of clients, and as an equal in a network of super-peers.
- Super-peer networks strike a balance between the efficiency of centralized search, and the autonomy, load balancing and robustness to attacks provided by distributed search.
- Super-peer networks take advantage of the heterogeneity of capabilities (e.g., bandwidth, processing power) across peers.

### ■ Super-peer networks paper:

- <http://faculty.washington.edu/wlloyd/courses/tcss558/papers/Designing%20a%20Super-Peer%20Network.pdf>

October 24, 2017

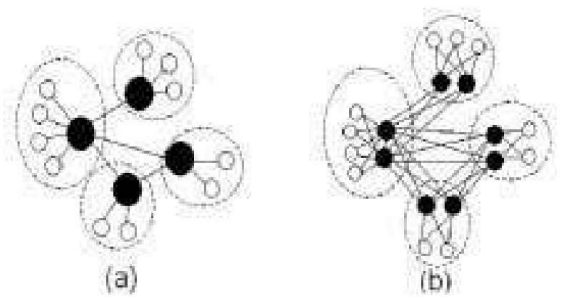
TCCS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L8.6

## FEEDBACK - 4

- **What are examples of super peers?**

- **Dashed lines are the circles...**



**Figure 1.** Illustration of a super-peer network (a) with no redundancy, (b) with 2-redundancy. Black nodes represent super-peers, white nodes represent clients. Clusters are marked by the dashed lines.

October 24, 2017

TCCS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L8.7

## FEEDBACK - 5

- **In policy based search methods, where we create (store) history of our searches, what happens if a node (or nodes) is added to the adhoc network or removed from the network?**

- **New nodes have no initial knowledge about neighbors**

- Discovery takes time
- Nodes start accumulating history when they join
- History could be acquired from neighbors

- **When nodes join/leave the network, will the history (at the nodes) be adjusted for policy-based search implementations?**

- **And if so, wouldn't that be time consuming?**

- Accumulating new history is slow
- Deleting should be fast

October 24, 2017

TCCS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L8.8

## UPDATE

- **Assignment 0 Questions**
- **Introduction of Assignment 1**
- **Due to the in class quiz and introduction of Assignment 1, lecture coverage was limited on October 24, 2017**
- **The midterm exam will be inclusive of all course content covered through October 31.**

October 24, 2017

TCSS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L8.9

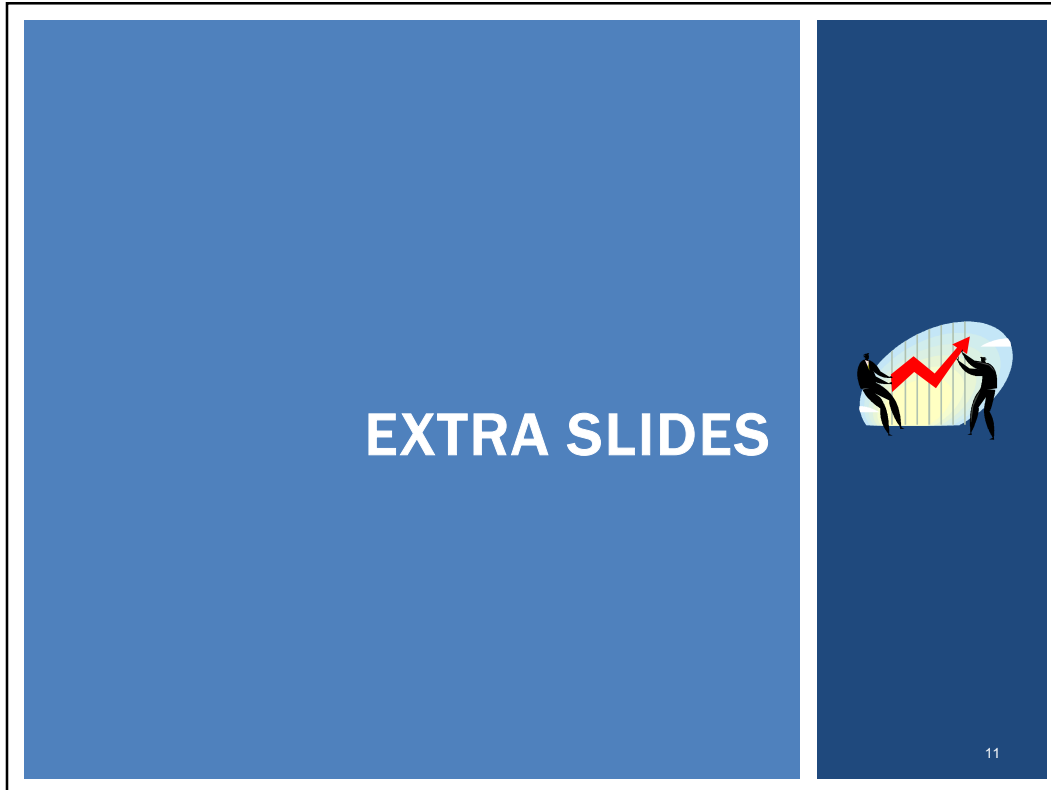
## QUESTIONS



October 24, 2017

TCSS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L8.10



**EXTRA SLIDES**

11