# 31268 Web Systems

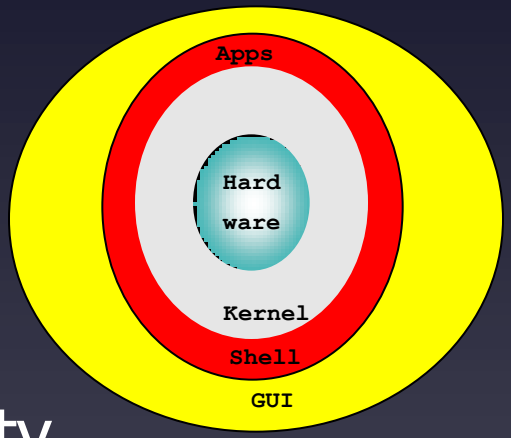
Week 2: Operating Systems

Part 4: Complexity

# 31268 Web Systems

UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Information Technology

## Week 2

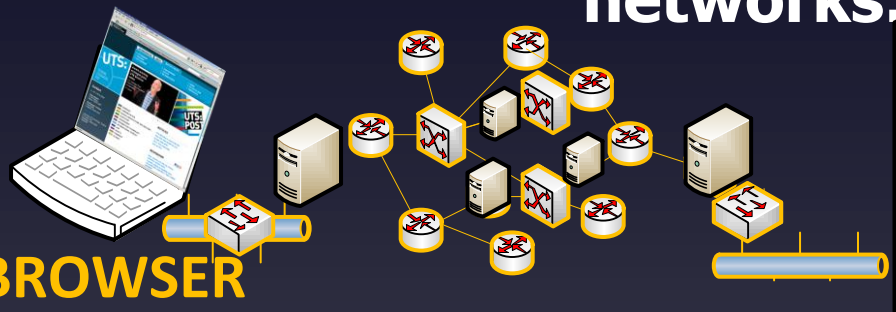



Operating Systems
File Storage - Complexity

# The web…

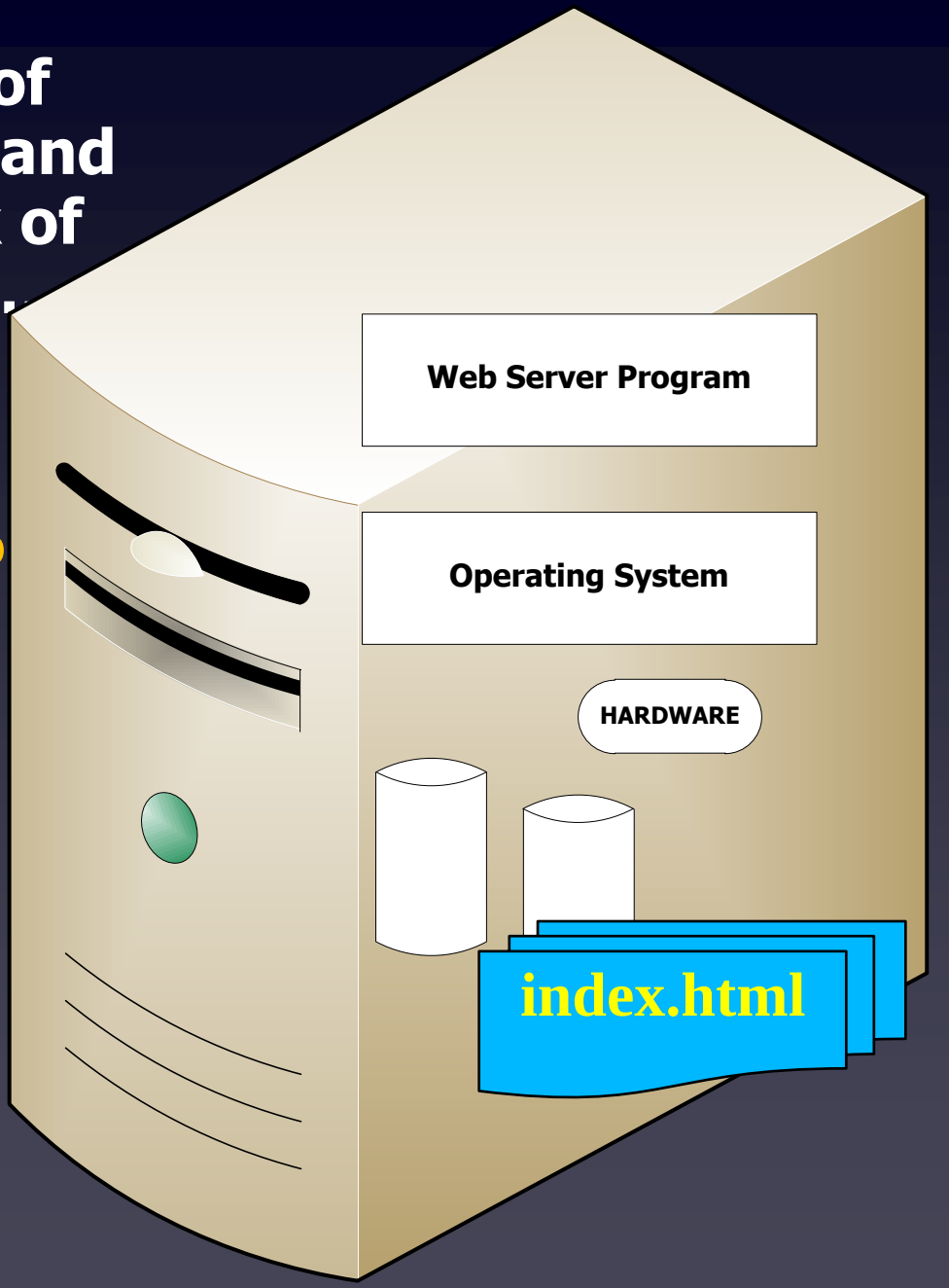
- Is It …

A bunch of computers and a network of networks.



BROWSER

… and a whacking **big** computer running the **web site program** on an **operating system** running on **HARDWARE**

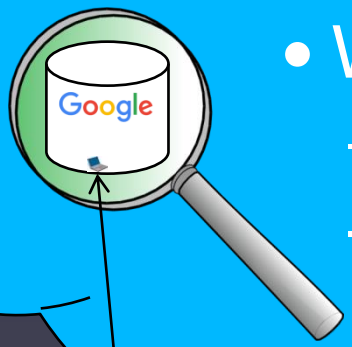


# The Web

WWW ++

- How big is the internet??
  - 1.003 Billion web sites? ++
    - http://news.netcraft.com/archives/category/web-server-survey/

→ But each website consists of many html pages and images??




- What about Google?
  - Est 2014 indexed 200Tb, 2016??
  - But this is estimated 0.004% of the internet

- What about Laptop?

***Chris' Laptop: 400,311 files/directories = 178Gb***

# The Web

- So many files!
- Question: Does the web, or google, or UTS, or your laptop store all the files in 1 single directory?

```
C:\> dir   web:\
934,856,356,384,959,437,893,947,373,248,094,
837,833,417,456,885,789,347,567,890  file(s)
1 dir(s)
∞ bytes free
```

# File Allocation Methods

UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Information Technology

Do what's the "BEST" way to store file's data?

➔ three common types of file allocation:

1. **Contiguous** Allocation
2. **Chained** or **Linked** Allocation
3. **Indexed** Allocation (e.g inode)

# File Allocation Methods

UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Information Technology

Do what's the "BEST" way to store file's data?

➔ three common types of file allocation:

1. **Contiguous** Allocation
2. **Chained** or **Linked** Allocation
3. **Indexed** Allocation (e.g inode)

➔Assume "BEST" == efficiency

➔ spacewise or timewise?

# Efficiency: → Complexity Theory

UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Information Technology

- What's complexity theory?
- Broadly speaking:
  → How much does it take to solve this problem?
-
- http://en.wikipedia.org/wiki/Computational_complexity_theory

# Efficiency: → Complexity Theory

UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Information Technology

- What's complexity theory?
- Broadly speaking:
  → How much does it take to solve this problem?
- Our question:
  → How much time (how many "reads") are required to find a particular block of the file?
- http://en.wikipedia.org/wiki/Computational_complexity_theory

# Complexity Theory (2)

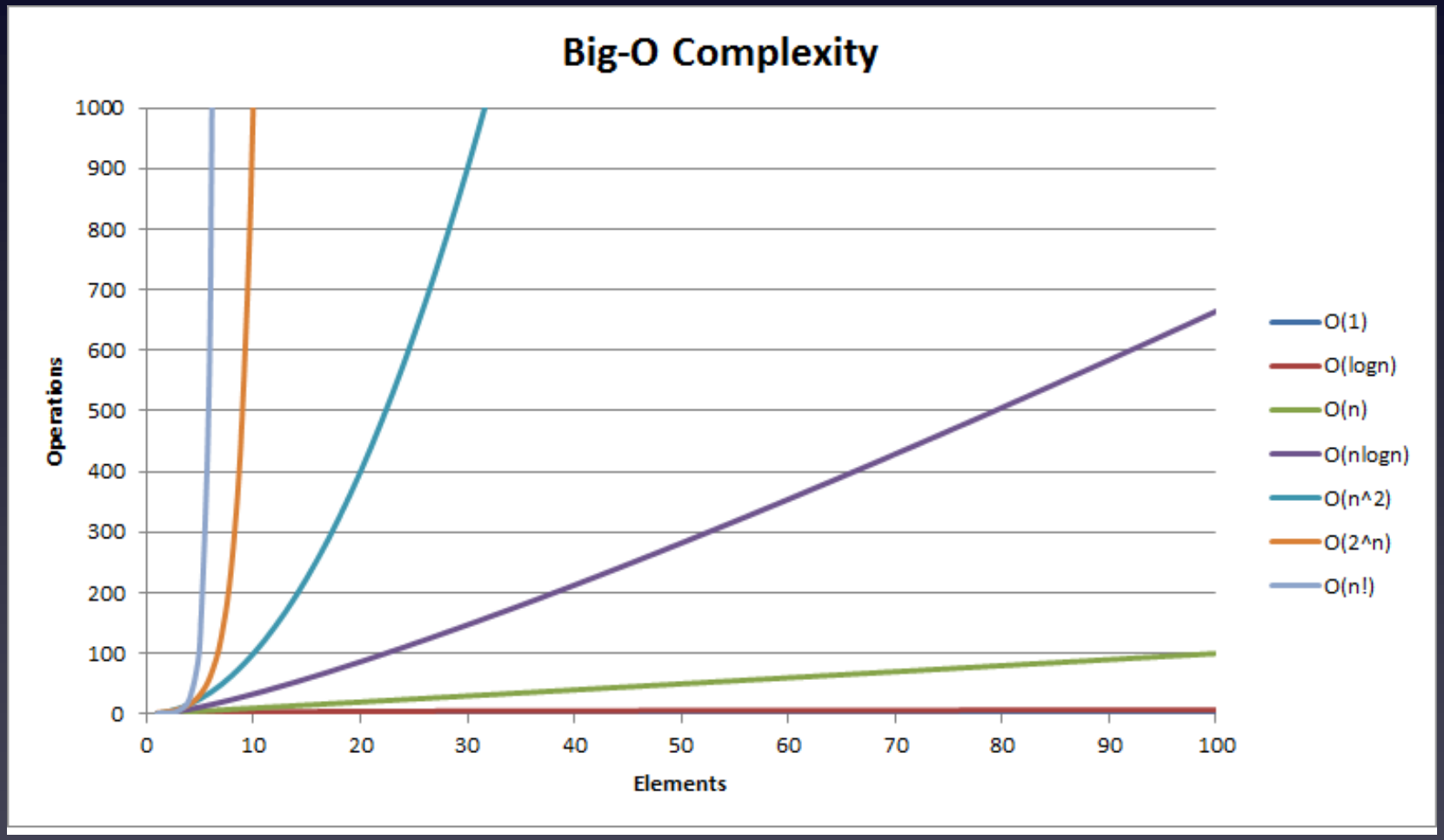
UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Information Technology

- Once more…
- Let *n* be the number of blocks in the file. To find a particular block it takes how many disk accesses?
- Contiguous: O(1) "about 1"
- Chained/Linked: O(n) "roughly n blocks"
- Indexed/Inode: O(logk(n)) blocks

*See http://en.wikipedia.org/wiki/Big_O_notation*

*http://bigocheatsheet.com/*

# From http://bigocheatsheet.com/




*The lower the better …*

*NOTE: O(1) is the x-axis & is not visible…*

# Complexity Theory

- Try this with Laptop sized filesystem
  - 400,000 files, **178 Gb**
- Contiguous: O(1) "about 1" (not likely but..)
  - Works on SSD quite well... *Except for fragmentation!*
- Chained/Linked: O(n) "roughly n blocks"
  - 1st block – need 1 read.
  - BUT 98 millionth block – need **98 million** reads!
- Indexed/Inode: O(logk(n)) blocks
  - $Log_{10}(178x10^9) \approx$ **11** reads needed!

# Complexity Theory

- Try this with Google sized filesystem
  - 15ExaByte = 15 x $10^{20}$ bytes = 150 Million TB

- Contiguous: O(1) "about 1" – *not likely*!
  - need to know exactly where that block is..

  *..... on over 10 million disk drives???*

# Complexity Theory

- Try this with Google sized filesystem
  - 15ExaByte = 15 x $10^{20}$ bytes = 150 Million TB

- Contiguous: O(1) "about 1" – *not likely*!
  - need to know exactly where that block is..

  ..... on over 10 million disk drives???

- Chained/Linked: O(n) "roughly n blocks"
  - 1st block – need 1 read.
  - BUT 987 billionth block → need 987 Billion reads!

# Complexity Theory

- Try this with Google sized filesystem
  - 15ExaByte = 15 x $10^{20}$ bytes = 150 Million TB
- Contiguous: O(1) "about 1" – *not likely*!
  - need to know exactly where that block is..

  ..... on over 10 million disk drives???
- Chained/Linked: O(n) "roughly n blocks"
  - $1^{st}$ block – need 1 read.
  - BUT 987 billionth block → need 987 Billion reads!
- Indexed/Inode: O(logk(n)) blocks
  - $Log_{10}(15x10^{20})$ ≈ ? reads needed!

# Complexity Theory

- Try this with Google sized filesystem
  - 15ExaByte = 15 x $10^{20}$ bytes = 150 Million TB
- Contiguous: O(1) "about 1" – *not likely*!
  - need to know exactly where that block is..

  ..... on over 10 million disk drives???
- Chained/Linked: O(n) "roughly n blocks"
  - 1st block – need 1 read.
  - BUT 987 billionth block → need 987 Billion reads!
- Indexed/Inode: O(logk(n)) blocks
  - $Log_{10}(15x10^{20}) \approx 21$ reads needed!

# Complexity Theory

- Obviously Indexed wins for big file systems.
- Big research issues on Internet WWW sized file systems though.....

# Todays Lab

- More on file systems
- Please complete "Linuxgym chapter 1" by 5pm Friday