Big Data Systems

CS 5965/6965 – FALL 2015

Today …

- General course overview
- Expectations from this course
- Q&A
- Introduction to Big Data
- Assignment #1

General Course Information

- Course Web Page
  - http://www.cs.utah.edu/~hari/teaching/fall2015.html
  - also on canvas
- Text & References
  - Mining of Massive Datasets, Leskovec, Rajaraman, Ullman
  - will use online resources and papers
- CHPC accounts → fill this form asap http://goo.gl/forms/nT8TkFtiJq
- TA – Vairavan Sivaraman
  - Email – vairavan.sivaraman@utah.edu
  - Office hours

Class Interaction …

- I strongly encourage discussions & interactions
- 10% for class interaction and online participation

Academic conduct

Adherence to the CoE and SoC academic guidelines is expected. Please read the following.

- College of Engineering Academic Guidelines from their web page
- School of Computing Misconduct Policy Please Read

TL;DR

NO CHEATING

Exams, Assignments & Projects

- Assignment 0 – Due in 1 week
- Assignments 1-2
  - one every week
  - Simple spark problems
- Assignments 3,4
  - Larger problems
  - 2 weeks each
- Final Project → submit proposal before start of fall break
  - Default project
- 2-3 students per group
- Mid-term exam after fall break

Things you should know

- No formal prerequisite
- Good Programming Skills (any language)
  - Python/Scala/Java
  - Make a case if you would like to use any other language
- Be prepared to learn new programming tools & techniques
- Linear Algebra
- Sequential algorithms, complexity analysis

What will we cover ?

- Hadoop & Spark
- MapReduce
- Parallel Algorithms
- Searching & Sorting in Parallel
- Randomized Algorithms - Graph Coloring
- PageRank
- Clustering Data
- Recommendation Systems
- Matrix Factorization
- Social Networks
- Graph Algorithms
- Large-Scale Machine Learning

Expectations

- The focus will be on understanding Big data algorithms and ensuring scalability for large scale problems
- While we will use Spark/Python, the skills acquired should be transferable to other platforms/frameworks
- While I will show you code examples, this is not a course to take if you want to learn to program in Java/Python/Scala/….
- Programming assignments will be frequent and hard. Additionally, the cluster is unreliable, so do not wait until the last evening to test your code.
- Choose the default project if you

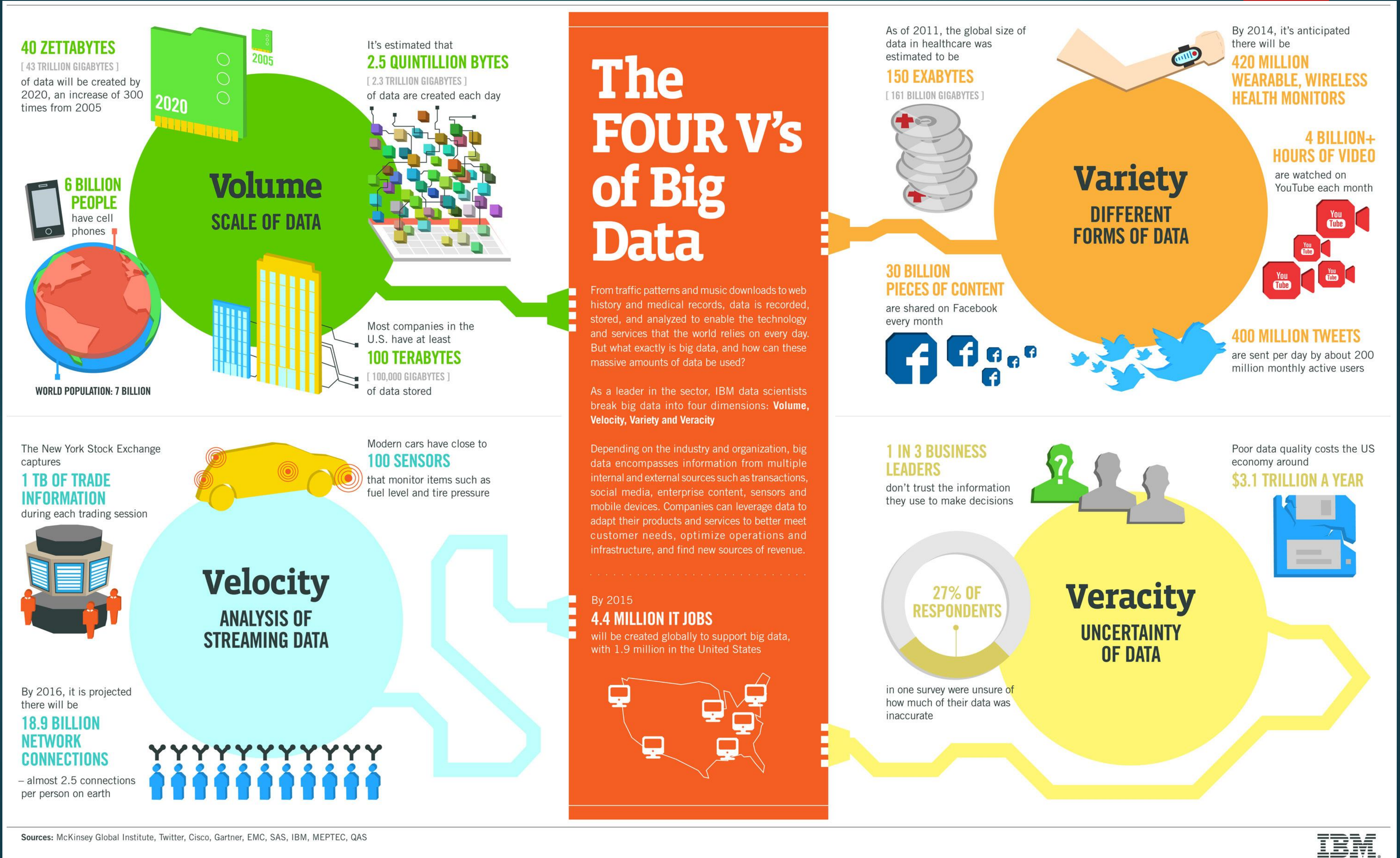
Where to look for help

- Course website / Canvas
- Email
- TA office hours → MEB 3XXX
- My office hours → MEB 3454 → Tue,Thu 2-3pm

Questions?

What is Big Data ?


DO WE HAVE ANY ACTIONABLE ANALYTICS FROM OUR BIG DATA IN THE CLOUD?
Dilbert.com DilbertCartoonist@gmail.com
YES, THE DATA SHOWS THAT MY PRODUCTIVITY PLUNGES WHENEVER YOU LEARN NEW JARGON.
1-9-13 © 2013 Scott Adams, Inc./Dist. by Universal Uclick
MAYBE IN-MEMORY COMPUTING WILL ACCEL-ERATE YOUR APPLICA-TIONS.
PLUNGE, PLUNGE, PLUNGE.


The FOUR V's of Big Data
From traffic patterns and music downloads to web history and medical records, data is recorded, stored, and analyzed to enable the technology and services that the world relies on every day. But what exactly is big data, and how can these massive amounts of data be used?
As a leader in the sector, IBM data scientists break big data into four dimensions: Volume, Velocity, Variety and Veracity
Depending on the industry and organization, big data encompasses information from multiple internal and external sources such as transactions, social media, enterprise content, sensors and mobile devices. Companies can leverage data to adapt their products and services to better meet customer needs, optimize operations and infrastructure, and find new sources of revenue.
By 2015
4.4 MILLION IT JOBS
will be created globally to support big data, with 1.9 million in the United States
Volume
SCALE OF DATA
40 ZETTABYTES
[43 TRILLION GIGABYTES]
of data will be created by 2020, an increase of 300 times from 2005
2020
2005
It's estimated that
2.5 QUINTILLION BYTES
[2.3 TRILLION GIGABYTES]
of data are created each day
6 BILLION PEOPLE
have cell phones
WORLD POPULATION: 7 BILLION
Most companies in the U.S. have at least
100 TERABYTES
[100,000 GIGABYTES]
of data stored
Variety
DIFFERENT FORMS OF DATA
As of 2011, the global size of data in healthcare was estimated to be
150 EXABYTES
[161 BILLION GIGABYTES]
By 2014, it's anticipated there will be
420 MILLION WEARABLE, WIRELESS HEALTH MONITORS
4 BILLION+ HOURS OF VIDEO
are watched on YouTube each month
You Tube
30 BILLION PIECES OF CONTENT
are shared on Facebook every month
400 MILLION TWEETS
are sent per day by about 200 million monthly active users
Velocity
ANALYSIS OF STREAMING DATA
The New York Stock Exchange captures
1 TB OF TRADE INFORMATION
during each trading session
Modern cars have close to
100 SENSORS
that monitor items such as fuel level and tire pressure
By 2016, it is projected there will be
18.9 BILLION NETWORK CONNECTIONS
– almost 2.5 connections per person on earth
Veracity
UNCERTAINTY OF DATA
1 IN 3 BUSINESS LEADERS
don't trust the information they use to make decisions
Poor data quality costs the US economy around
$3.1 TRILLION A YEAR
27% OF RESPONDENTS
in one survey were unsure of how much of their data was inaccurate
Sources: McKinsey Global Institute, Twitter, Cisco, Gartner, EMC, SAS, IBM, MEPTEC, QAS
IBM

The recognition that data is at the center of our digital world and that there are big challenges in collecting, storing, processing, analyzing, and making use of such data.

What is Big depends on the application domain

Kinds of Data

- Web data & web data accesses
- Emails, chats, tweet
- Telephone data
- Public databases – Gene banks, census records, …
- Private databases – medical records, credit card transactions, …
- Sensor data – camera surveillance, wearable sensors, seismograms
- Byproduct of computer systems operations – power signal, CPU events, …
- … … …

Data is valuable

- Google & facebook make money by mining user data
  - for revenue from advertisements
- Financial firms analyze financial records, real-time transactions and current events for profitable trades
- Medical records can be processed for better – and potentially cheaper – health care
- Roadways are monitored for traffic analysis and control
- Face detection in airports

Collection of Big Data

- The amount of data available is increasing exponentially
- But, it is still challenging to collect it
  - Difficult to get access
  - Redundancy
  - Noise

Processing & Analysis of Big Data

- Large datasets do not fit in memory
- Processing large datasets is time consuming
  - Parallel processing is necessary → challenging
- Message Passing Interface (MPI) – (1991)
  - Low(er) level C/Fortran API for communication
  - Powerful, hard(er) to code, !fault tolerant
- Mapreduce – Google (2004)
  - Originated from web data processing
  - ease of programming, fault tolerant
  - limited semantics
- Spark, GraphLab, Storm, ……

Storage & I/O

- Storage and I/O are critical for big data performance and reliability
- Hardware: disks, Flash, SSD, nonvolatile memory, 3D memory
- Parallelism: RAID, parallel data storage, DFS
- Data durability and consistency

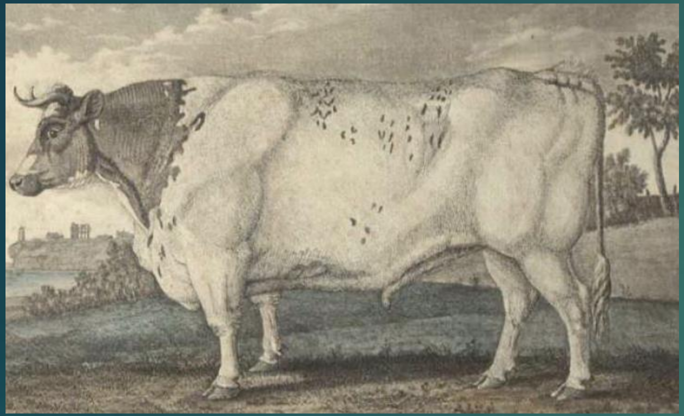
Data privacy & protection

- Misuse of big data is a big concern
  - A person’s online activities can reveal all aspects of the person’s life
- Systems need to provide clear guidelines on data privacy and protection
  - Sensitive clinical information
- Understand how the big data world operates
  - as an user
  - as a developer

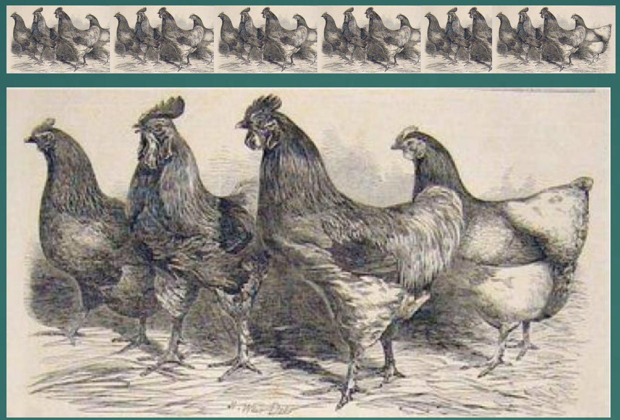
Parallel Thinking

THE MOST IMPORTANT GOAL OF TODAY'S LECTURE

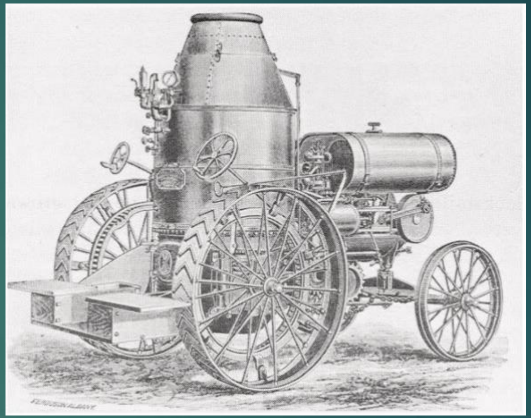
Parallelism & beyond …



1 ox: single core performance



1024 chickens: parallelism



tractor: better algorithms

If you were plowing a field, which would you rather use? Two strong oxen or 1024 chickens?

Seymour Cray

Credit: Phillip Stanley-Marbell

Consider an array A with n elements,

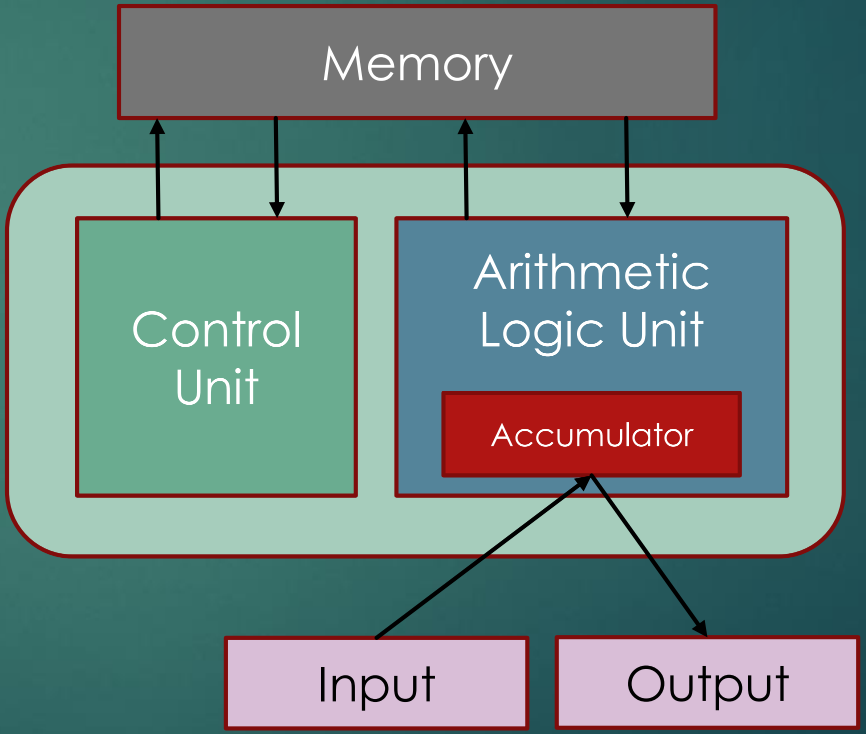
Goal: to compute,

$$x = \sum_{1}^{n} \sqrt{A_i}$$

Machine Model
Programming Model
Performance analysis

Von Neumann architecture

- Central Processing Unit (CPU, Core)
- Memory
- Input/Output (I/O)
- One instruction per unit/time
- **Sequential**



Characterizing algorithm performance

- O-notation
  - Given an input of size n, let $T(n)$ be the total time, and $S(n)$ the necessary storage
  - Given a problem, is there a way to compute lower bounds on storage and time → **Algorithmic Complexity**
  - $T(n) = O(f(n))$ means

    $T(n) \leq cf(n)$, where c is some unknown positive constant

    compare algorithms by comparing $f(n)$.

Scalability

- Scale Vertically → scale-up
  - Add resources to a single node
  - CPU, memory, disks,
- Scale Horizontally → scale-out
  - Add more nodes to the system

Parallel Performance

- Speedup

  best sequential time/time on p processors

- Efficiency

  $\text{speedup}/p, \qquad (< 1)$

Scalability

Amdahl's Law

- Sequential bottlenecks:

  Let s be the percentage of the overall work that is sequential

- Then, the speedup is given by

$$S = \frac{1}{s + \frac{1-s}{p}} \leq \frac{1}{s}$$

Gustafson

Sequential part should be independent of the problem size

Increase problem size, with increasing number of processors

Strong & Weak Scalability

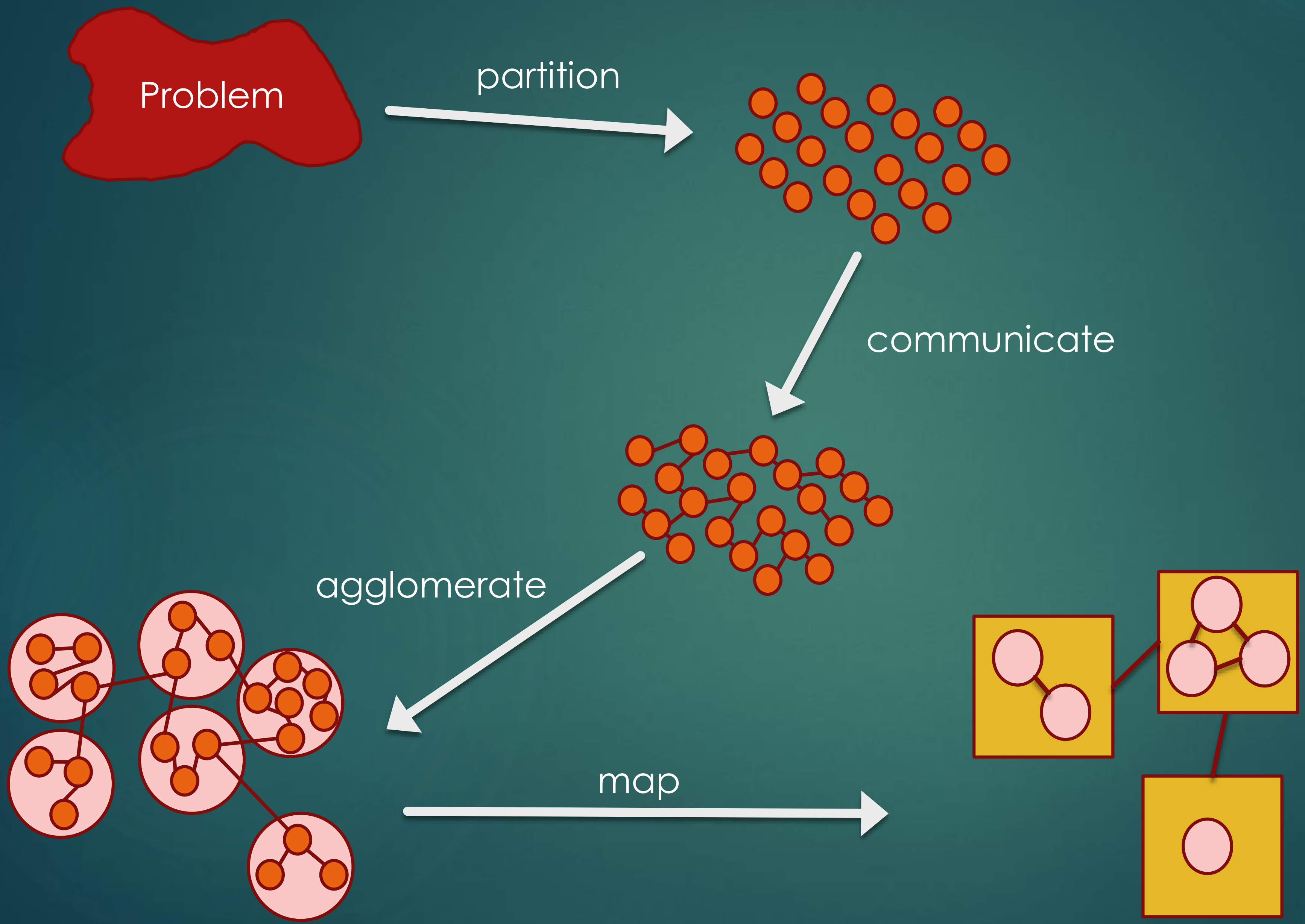
Increasing number of cores

Strong (fixed-sized) scalability — keep problem size fixed

Weak (fixed-sized) scalability — keep problem size/core fixed

Parallel Programming

- Partition Work   Data & Tasks
- Determine Communication
- Agglomeration to number of available processors
- Map to processors
- Tune for architecture


Problem
partition
communicate
agglomerate
map

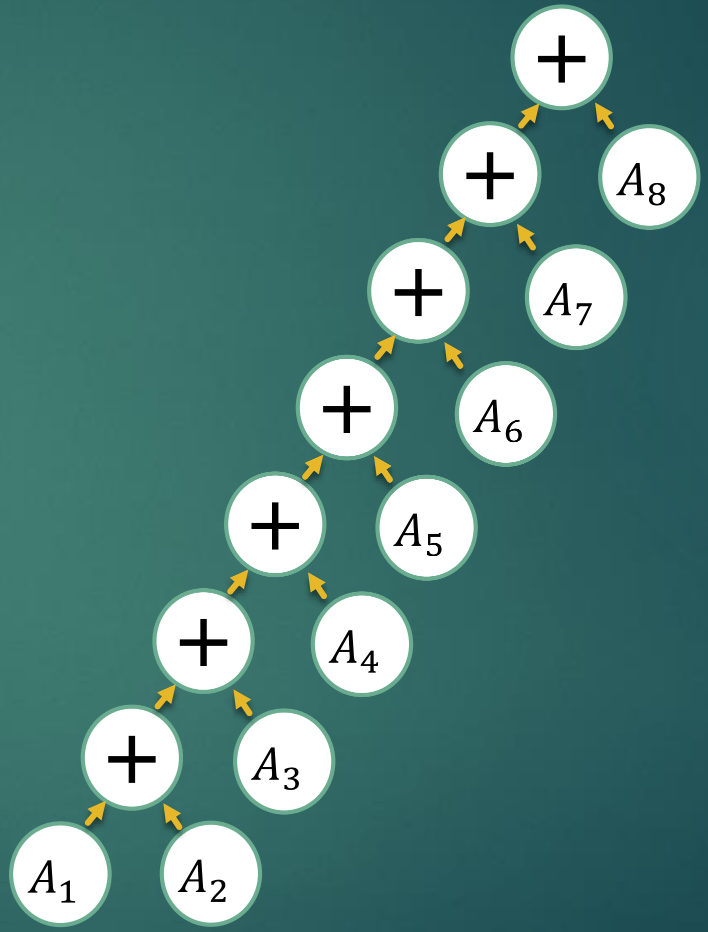
Consider an array A with n elements,

Goal: to compute,

$$x = \sum_{1}^{n} \sqrt{A_i}$$

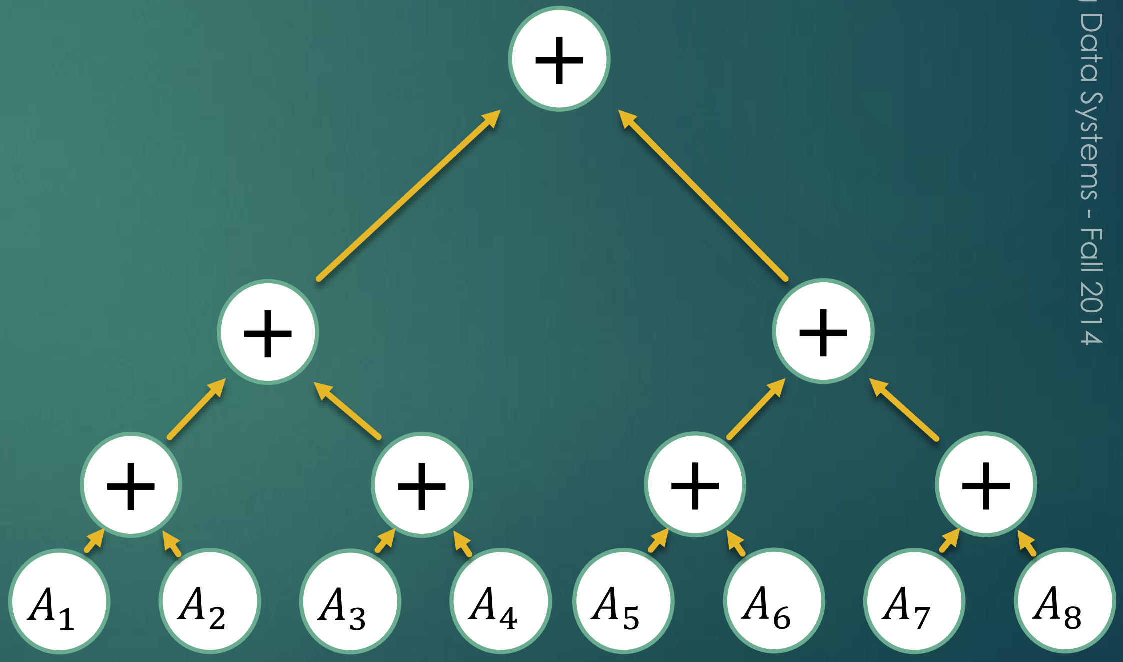
Work/Depth Models

- Abstract programming model
- Exposes the parallelism
  - Compute work W and depth D
  - D - longest chain of dependencies
  - $P = W/D$
- Directed Acyclic Graphs
- Concepts
  - parallel for (data decomposition)
  - recursion (divide and conquer)



Work/Depth Models

- Abstract programming model
- Exposes the parallelism
  - Compute work W and depth D
  - D - longest chain of dependencies
  - $P = W/D$
- Directed Acyclic Graphs
- Concepts
  - parallel for (data decomposition)
  - recursion (divide and conquer)



Sequential vs Parallel for

- Dependent statements
  - $W = \sum W_i$
  - $D = \sum D_i$
- Independent statements
  - $W = \sum W_i$
  - $D = \max(D_i)$

Parallel Sum – language model

```
// Recursive implementation
Algorithm SUM(a, n)
// Input: array a of length n = 2^k, k = log n
    parallel_for i ← [0,n/2)
        b(i) ← a(2i) + a(2i+1)
    return SUM(b); // W(n/2), D(n/2)
```

Complexity:

$$D(n) = D\left(\frac{n}{2}\right) + O(1) = O(\log n)$$

$$W(n) = W\left(\frac{n}{2}\right) + O(n) = O(n)$$

Questions?