Introduction to Distributed Computing Algorithms

Jeff M. Phillips

November 19, 2011
Many Unorganized Computers

I can't do this by myself!
And I don't really know where anyone else is!

Too much data processing for one computer.
Not part of an organized cluster.
Many Unorganized Computers

I can't do this by myself! And I don't really know where anyone else is!

Too much data processing for one computer. Not part of an organized cluster.

Could be huge job. Could be small computer.
Distribute computation out to friends.

Why won't this work?

Transferring big data very expensive!

Often more expensive than computation!
Many Unorganized Computers

Distribute computation out to friends.
Why won’t this work?
Many Unorganized Computers

Distribute computation out to friends. Why won’t this work?

Transferring big data very expensive! Often more expensive than computation!
Many Unorganized Computers

Goal:
Minimize Communication!
Distribute computation out to friends.
When might this work?
Many Unorganized Computers

Distribute computation out to friends.
When might this work?

Computation is Very Expensive. (Exponential)
Molecular dynamics
- typically very sequential
- many inaccurate and average
- explore different scenarios
Molecular dynamics
- typically very sequential
- many inaccurate and average
- explore different scenarios

Central Server: sends out work units. Nodes have fixed time to complete. Failures lead to shorted jobs.
Folding@Home

How large is it?

- 439,000 CPUs
- 37,000 GPUs
- 21,000 PS3s
- 6.7 petaFLOPS

Molecular dynamics

- typically very sequential
- many inaccurate and average
- explore different scenarios

Central Server: sends out work units. Nodes have fixed time to complete. Failures lead to shorted jobs.
BOINC

Berkeley Open Infrastructure for Network Computing

SETI@Home

- 451,000 CPUs
- 5.6 petaFLOPS

More restrictive protocol than Folding@Home.
Checks results, and often duplicates.
Each processor is connected to server.

- two-way communication.
- sometimes, data can originate on processor
- can stream in, or static
- server can be overloaded
Flat Model

Each processor is connected to server.

- two-way communication.
- sometimes, data can originate on processor
- can stream in, or static
- server can be overloaded
Random Sampling

Random Sample $t$ items from $k$ sites.

$O(k + t)$ communication.

1. Each node assigns a random variable $u_i$ to all its data $v_i$.
2. Sends top value $u_i$ to server as $(x_i, u_i)$.
3. Server keeps $x_i$ with top $u_i$.
4. Asks corresponding node for next top value.
5. Go to 3.
Random Sampling

Random Sample $t$ items from $k$ sites. $O(k + t)$ communication.

1. Each node assigns a random variable $u_i$ to all its data $v_i$.
2. Sends top value $u_i$ to server as $(x_i, u_i)$.
3. Server keeps $x_i$ with top $u_i$.
4. Asks corresponding node for next top value.
5. Go to 3.
Tree Model

Many processors connected to server through tree

- two-way communication.
- arbitrary topology (tree)
Tree Model

Many processors connected to server through tree

- two-way communication.
- arbitrary topology (tree)
- can stream in, or static
Tree Model

Many processors connected to server through tree

- two-way communication.
- arbitrary topology (tree)
- can stream in, or static
- less stress on server
- latency slower
- might multi-cast from server
- sometimes only pass summaries
Mergeable Summaries

Aggregation Network
- Each node $i$ has data $X_i$
- Creates summary $S_i = \sigma(X_i)$
- has $\varepsilon$-error, size $f(\varepsilon)$

Can merge two summaries:
- $S = \mu(S_1, S_2)$
- has $\varepsilon$-error on $S_1 \cup S_2$
- size $f(\varepsilon)$

Neither error nor size grows.
Can be used like `sum` or `max`. 
Clique Model

Many computers, all can talk (internet)

- may limit degree (10000+ nodes)
- central server may control

Distributed Hash Tables
- Stores data distributed (like GFS)
- Distribute files (Bitorrent)
- Minimize communication
tolerate failure
Clique Model

Many computers, all can talk (internet)
- may limit degree (10000+ nodes)
- central server may control
- may have no central server
Clique Model

Many computers, all can talk (internet)
- may limit degree (10000+ nodes)
- central server may control
- may have no central server

Distributed Hash Tables
- Stores data distributed (like GFS)
- Distribute files (BitTorrent)
Clique Model

Many computers, all can talk (internet)
  ▶ may limit degree (10000+ nodes)
  ▶ central server may control
  ▶ may have no central server

Distributed Hash Tables
  ▶ Stores data distributed (like GFS)
  ▶ Distribute files (Bitorrent)

Minimize communication tolerate failure
Lower Bounds

- $k$ computers: $i$
- each computer has $n$ bits: $X_i$

Compute $f(X_1, X_2, \ldots, X_k)$. 
Lower Bounds

- $k$ computers: $i$
- each computer has $n$ bits: $X_i$

Compute $f(X_1, X_2, \ldots, X_k)$.

Number-on-forehead
- See all data, but your own
Lower Bounds

- $k$ computers: $i$
- Each computer has $n$ bits: $X_i$

Compute $f(X_1, X_2, \ldots, X_k)$.

Number-on-forehead
- See all data, but your own

Blackboard
- Costs to write to BB, free to read
Lower Bounds

- $k$ computers: $i$
- each computer has $n$ bits: $X_i$

Compute $f(X_1, X_2, \ldots, X_k)$.

Number-on-forehead
- See all data, but your own

Blackboard
- Costs to write to BB, free to read

Multi-Party
- All-pair
- $f = \{\text{OR}, \text{XOR}, \ldots\}$
  $\Omega(nk)$ comm