Data Analytics over Decentralized Architectures
From Clusters to the Edge

Davide Frey
davide.frey@inria.fr
Inria Rennes
ASAP Team @ Inria Rennes

**DISC'15**
**SODA'15**

Information dissemination over social networks

**Europsys'14**
**Middleware'14**

Cloud computing meets P2P

**ATC'15**

Scalability & Privacy through Decentralization

**TCS'13**
**DSN'15**

Privacy-aware decentralized computation

**PODC'14**
**STOC'15**

Computability and efficiency of distributed Systems

**ICDE'16**
**VLDB'16**

Scalable KNNs graphs & queries

**DISC'15**

Principles of Distributed Algorithms

**Theory**

**Practice**

**App**

Privacy-aware Affordable Personalization

Tech Transfer

Paris, January 6, 2017

Davide Frey - davide.frey@inria.fr
Decentralized Data Analytics
Why Distribute Computation

- Speed/Parallelization
- Scale
- Privacy / Cost /Energy

- Parallelize for Performance
- Decentralize for Simplicity
Are we Already Done?
Outline

• Brief SOTA
  • Map Reduce / Hadoop
  • Data Parallelism
  • Model Parallelism

• ASAP’s Focus
  • Massively Decentralized Data
  • Privacy
MapReduce Example: G-Means

G-Means as a collection of map-reduce jobs

Algorithm 2 KMeansAndFindNewCenters Mapper

<table>
<thead>
<tr>
<th>Input: point (text)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output:</td>
</tr>
<tr>
<td>centroid (long) ⇒ coordinates (float[]), 1 (int)</td>
</tr>
<tr>
<td>centroid + OFFSET (long) ⇒ coordinates (float[]), 1 (int)</td>
</tr>
</tbody>
</table>

procedure MAP(key, point)

Find nearest center
Emit(centroid, point)
Emit(centroid + OFFSET, point)
end procedure

Algorithm 3 TestClusters Mapper

<table>
<thead>
<tr>
<th>Input: point (text)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output: vectorid (int) ⇒ projection (double)</td>
</tr>
</tbody>
</table>

procedure SETUP

Build vectors from center pairs
Read centers from previous iteration
end procedure

procedure MAP(key, point)

Find nearest center
Find corresponding vector
Compute projection of point on vector
Emit(vectorid, projection)
end procedure

Algorithm 4 TestClusters Reducer

| Input: vectorid (int) ⇒ projection (double) |

procedure REDUCE(vectorid, projections)

Read projections to build a vector
Normalize vector (mean 0, stddev 1)
ADTEST(vector)
if normal then

Mark cluster as found
end if
end procedure

Algorithm 5 TestFewClusters Mapper

<table>
<thead>
<tr>
<th>Input: point (text)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output: vectorid (int) ⇒ A^2 (double)</td>
</tr>
</tbody>
</table>

procedure SETUP

Build vectors from center pairs
Read centers from previous iteration
end procedure

procedure MAP(key, point)

Find nearest center
Find corresponding vector
Compute projection of point on vector
Add projection to list vectorid
end procedure

procedure CLOSE

for Each list do

Read projections to build a vector
Normalize vector (mean 0, stddev 1)
Compute A^2 = adtest(vector)
Emit(vectorid ⇒ A^2)
end for
end procedure

Scalable KNN computation

Exploit greedy solutions

Data Parallelism: Parameter Servers

- Workers share common model
- Treat different portions of the data
- (Independently) update parameters

Stale Synchronous Parallel model

- Commutative associative parameter updates: $\theta \leftarrow \theta + \delta$

Model Parallelism: Google DistBelief

Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc’Aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, Quoc V. Le and Andrew Y. Ng "Large Scale Distributed Deep Networks". Advances in Neural Information Processing Systems 2012.
Model Parallelism: STRADS

Massive Decentralization

- Lot
- Set-top boxes
- Edge
Epidemics as a Tool

Gossip-based dissemination
Epidemics as a Tool

Gossip-based dissemination
Epidemics as a Tool

Gossip-based dissemination
Epidemics as a Tool

Gossip-based dissemination
Epidemics as a Tool

Gossip-based dissemination
Epidemics as a Tool

Gossip-based dissemination
Epidemic Recommendation

Exploit epidemic clustering to build KNN

The Case of News Items (and beyond)


http://www.mediego.com/
Making Recommendation Private: Obfuscation

Profiles kept locally

News item (received)

item profile

compact profile

filter profile

id, opinion, time

signature

mask of popularity

item profile

Profiles exchanged with others

News item (forwarded)

obfuscated profile

Making Recommendation Private: Landmarks

<table>
<thead>
<tr>
<th></th>
<th>L1</th>
<th>L2</th>
<th>L3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.55</td>
<td>0.2</td>
<td>0.9</td>
</tr>
<tr>
<td>B</td>
<td>0.42</td>
<td>0.6</td>
<td>0.18</td>
</tr>
</tbody>
</table>

A Slide to Bring Home

Decentralize computation on private data
- Cluster users by interest, privacy level, …
- Build local models
- Aggregate models at higher levels

Explore tradeoff spaces

Create networks of
- Browsers
- Set-top boxes
- Small/Tiny devices (plug-computers, smartphones, sensors)

Build Private Applications
- Recommendation
- Aggregation
- Personalized Services
- Inter-Silo Analytics