Computational Social Choice in the Cloud

Theresa Csar, Martin Lackner, Emanuel Sallinger, Reinhard Pichler

Technische Universität Wien
Oxford University

PPI, Stuttgart, March 2017
Cloud Computing Technologies

- MapReduce
- Pregel
- Giraph
- GraphX
- Spark
- Hadoop
MapReduce

Map Phase
The input data is mapped to (key,value)-pairs

Shuffle Phase
The (key,value)-pairs are assigned to the reduce tasks

Reduce Phase
Each reduce task performs a simple calculation on all its values
What’s an election?

Given as lists of preferences with n votes and m candidates.

We are interested in finding the best candidate, or the set of best candidates.
MapReduce and Elections by Example

Given a set of \( m = 3 \) candidates \( a,b,c \) and \( n \) voters. Each voter provides a ranking of candidates, e.g.: \( a > b > c \)

**Borda Scoring Rule:** The candidate ranked first receives \( m-1 \) points, the second \( m-2 \) points, etc.
MapReduce and Elections by Example – Borda Scoring Rule

\[
\begin{align*}
\text{Map:} & \quad a > b > c \\
& \quad b > a > c \\
& \quad c > a > b \\
\text{Shuffle:} & \quad (a, 2) \\
& \quad (b, 2) \\
& \quad (c, 2) \\
\text{Reduce:} & \quad (a, 4) \\
& \quad (b, 3) \\
& \quad (c, 2)
\end{align*}
\]
Performance Analysis of a Mapreduce Computation

• data replication rate (rr)
• number of MapReduce rounds
• number of keys / reduce tasks
• wall clock time (wct): the maximum time consumed by a single computation path in the parallel execution of the algorithm
• total communication cost (tcc): number of values transferred during the computation.
The scores of all candidates given a scoring rule can be computed using MapReduce with the following characteristics:

$rr = 1$, # rounds = 1, # keys = m, wct $\leq n$, and $tcc \leq mn$. 
Winner Determination in Elections

- **Scoring Rules**: Borda Scoring Rule, ...

- **Copeland Set**: The Copeland set is based on Copeland scores. The Copeland score of candidate $a$ is defined as $|\{b \in C : a > b\}| - |\{b \in C : b < a\}|$. The Copeland set is the set of candidates that have the maximum Copeland score.

- The **Smith set** is the (unique) smallest set of candidates that dominate all outside candidates.

- The **Schwartz set** is the union of minimal sets that are not dominated by outside candidates.
Winner Determination in Elections
Input Data

Preference-Lists (Scoring Rules)
- Number of Lists / Number of Votes
- Length of Votes / Number of Candidates

Dominance Graph (Smith Set, Copeland Set, Schwartz Set)
- Number of Candidates
Smith Set

Definition
Candidate a is in the Smith set if and only if for every candidate b there is a path from a to b in the weak dominance graph.

Brandt, Fischer and Harrenstein (2009) show that in the weak dominance graph a vertex t is not reachable from a vertex s if and only if there exists a vertex v such that \( D_2(v) = D_3(v) \), \( s \in D_2(v) \), and \( t / \in D_2(v) \).

→ In other words: We only need paths of length 3 to find the Smith Set.
Smith Set Algorithm-sketch

• Preprocessing step (create needed datastructure)
• 2 MR-Rounds: to find paths of length 2 und 3 (or 4)
• Postprocessing: find vertices contained in the Smith set
Smith Set – Vertex Datastructure

„Think like a vertex“

Each vertex saves three sets storing information on incoming and outgoing edges for a vertex $a$ as follows:

- the set `old` stores all vertices that have been found previously to be reachable from $a$;
- the set `new` stores all vertices that have been found in the last map-reduce round to be reachable from $a$;
- the set `reachedBy` stores all vertices known to reach $a$;
Vertex Data Structure – in action

MapVertex

Input | Output
--- | ---
Vertex A
new = {B} | (A,{{B},"old")
old = {} | (A,{{B},"new")
rB = {} 
Vertex B
new = {C} | (B,{{C},"old")
old = {} | (B,{{A},"rB")
rB = {A} 
Vertex C
... | ...

ReduceVertex A

Input | Output
--- | ---
(A,{{B},"old") | Vertex A
(A,{{C},"new") | new = {C}
old = {B} | rB = {}

ReduceVertex B

Input | Output
--- | ---
(B,{{C},"old") | Vertex B
(B,{{A},"rB") | new = {D}
(B,{{D},"new") | old = {C}
rB = {A}
Experimental Design

• Mapreduce Java Implementation
github.com/theresacsar/bigvoting

• Amazon Web Services (AWS) – Elastic Compute Cloud

• Synthetic Datasets
  • with varying number of candidates and edges in the dominance graph
    (m=7000 candidates and m2/10 edges)
  • up to 128 EC2 instances
Future Work – Exploring other Technologies

• Pregel (Giraph, GraphX) „Think like a vertex“
  • Pregel-like systems are better suited for iterative Graph computations

• Spark
  • Interactivity
  • Data is loaded in memory
Future Work

• Other Technologies
• Using real word data
  • Results from search engines
• Other Rules for Winner Determination
Thank you for listening!
ask questions now or send me an email csar@dbai.tuwien.ac.at 😊