L24: MapReduce and DFS

Distributed Systems for the masses

Jeff M. Phillips

April 15, 2019

Poster

Don't put too much, or small font!

- Succinct title (and names)
- What is the problem and data you worked on?
- What were the key ideas in your approach?
- What techniques from the class did you use?
- What did you learn?

Big Data

<u>Algo</u> data too big for one computer

(Stats: the limit is here)

<u>Algo</u> ⟳ count

(1) Data enormous
200 - 1000 machines

(2) Data <u>Static</u> (some appends)

Data Science Revolution

Classic Science

① Make hypothesis
② Gather Data
③ Check if data supports hypothesis

iid, used once

Stats

⓪ Precompute, gather / store Massive Data
① Find statistically significant patterns in data

Data Mining

MapReduce: @ Google
- Build inverted index
- Run PageRank

Distributed File System

<key, value> pair JSON

key: "unique" id

value:
raw, sometimes messy data

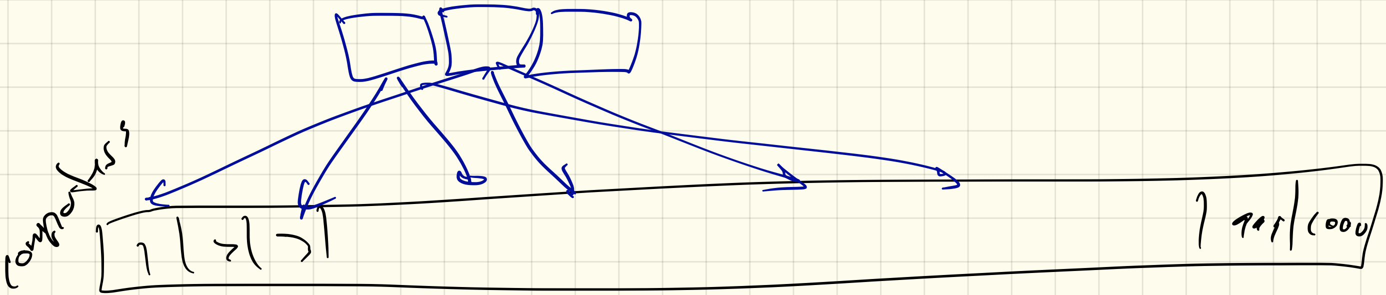
log id	actual log
web address	html, out-going links
doc id	list of words ↳ k-grams
word index	list of docs w/ that word.

Data in Blocks

Block = 64 MB

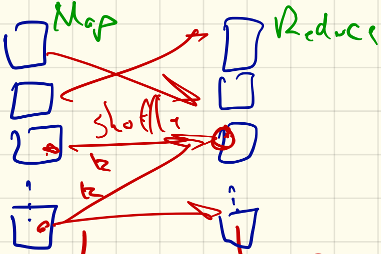
Each block is stored 2 or 3 times

get rid of locality



very resilient (to machine failure)

Map Reduce



1. Map : Read block from key-value store
   - pull out import info
   - determine "new locality" → identify new key

1.5 Combine : Merge <k,v> before shuffle

2. Shuffle group by key
   → same key, same new block

3. Reduce : Look at all k-v pairs w/ same new key, post. process (analysis)

Word Count

(Hello World)

Consider as input all of English Wikipedia stored in DFS. Goal is to count how many times each word is used.

Map: $\overset{7}{\text{doc id}} \to \{ \langle \overset{\text{it}}{\text{word}}, \overset{1}{\text{id}} \rangle, \langle \overset{\text{was}}{\text{word}}, \overset{1}{\text{id}} \rangle, \langle \overset{\text{good}}{\text{word}}, \overset{1}{\text{id}} \rangle$

Combine: $\{ \langle w, v_1 \rangle, \langle w, v_2 \rangle, \dots \} \to \langle w, \sum_i v_i \rangle$

Reduce: Input $\{ \langle \text{good}, \overset{v_1}{1} \rangle, \langle \text{good}, \underset{3}{\overset{v_2}{1}} \rangle, \dots \langle \text{good}, \underset{2}{\overset{v_n}{1}} \rangle$

Output $\langle \text{good}, \sum_i v_i \rangle$

"the" 7'2a

Input $\{ \langle \text{the}, 1 \rangle, \langle \text{the}, 1 \rangle, \dots \langle \text{the}, 1 \rangle$

Inverted Index

Consider as input all of English Wikipedia stored in DFS. Goal is to build an index, so each word has a list of pages it is in.

Map: article id $\rightarrow \langle word_1, id \rangle, \langle word_2, id \rangle$ ...

Reduce: $\langle word_i, id_1 \rangle, \langle word_i, id_2 \rangle, \ldots$

$$\rightarrow \langle word_i, \bigcup_j id_j \rangle$$

Phrases

Consider as input all of English Wikipedia stored in DFS. Goal is to build an index, on 3-grams (sequence of 3 words) that appears on exactly one page, with link to page.

Map : doc id : $\langle word_1, word_2, word_3, id \rangle$

$\langle word_2, word_3, word_4, id \rangle$

$\vdots$

Reduce: $\langle (w_i, w_{i'}, w_{i''}), id_1 \rangle, \langle (w_i, w_{i'}, w_{i''}), id_2 \rangle$

$\rightarrow \langle (w_i, w_{i'}, w_{i''}), \bigcup_j id_j \rangle$

Label Propagation (Graph)

Consider a large graph $G = (V, E)$ (e.g., a social network), with a subset of notes $V' \subset V$ with labels (e.g., {pos, neg}). Each node stores its label (if any) and edges.

Assign a vertex a label if (a) unlabled, (b) has ≥ 5 labeled neighbors, (c) based on majority vote.

node id — another node id — label

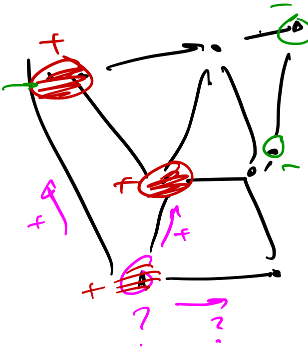
Map $\langle id, (+, e_1, e_2 \ldots e_k) \rangle$

if (lab $\neq \phi$) $\langle e_1, lab \rangle$, $\langle e_2, lab \rangle$

$\rightarrow \langle id, (+, \text{fixed}) \rangle$

Reduce: $\langle id, lab_1 \rangle$, $\langle id, lab_2 \rangle$

if (no fixed) then if #lab > 5 $\langle id, maj(lab_1, lab_2 \ldots) \rangle$



Label Propagation (Embedding)

Consider a data set $X \subset \mathbb{R}^d$, with a subset of points $X' \subset X$ with labels (e.g., {pos, neg}). Implicitly defines graph with $V = X$ and E using $k = 20$ nearest neighbors.

Assign a vertex a label if (a) unlabled, (b) has ≥ 5 labeled neighbors, (c) based on majority vote.

LSH, in 2 rounds

Example PageRank

1 block

$$M = \begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 1 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

one web page

$$q^* = P^* q_0$$

$$= P(P(P\dots\underbrace{(P q_0)}_{\text{mat-vec}}))$$

1 round

≤ 50 rounds

Example PageRank

$$
\begin{array}{cc}
 & \begin{array}{cccc} q(1) & q(2) & q(3) & q(4) \end{array} \\
M = & \begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 1 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}
\end{array}
$$

$\langle id, q(1)\cdot\frac{1}{3}\rangle$

Stripes:

$$
M_1 = \begin{bmatrix} 0 \\ 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \quad M_2 = \begin{bmatrix} 1/2 \\ 0 \\ 0 \\ 1/2 \end{bmatrix} \quad M_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad M_4 = \begin{bmatrix} 0 \\ 1/2 \\ 1/2 \\ 0 \end{bmatrix}
$$

These are stored as $(1 : (1/3, 2), (1/3, 3), (1/3, 4))$, $(2 : (1/2, 1)(1/2, 4))$, $(3 : (1, 2))$, and $(4 : (1/3, 1), (1/2, 2))$.

Example PageRank

$$\begin{array}{cc} & \begin{array}{cccc} g(1) & g(2) & & \end{array} \\ M = & \left[\begin{array}{cccc} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 1 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{array}\right] \end{array}$$

id_4

Blocks:

$$M_{1,1} = \left[\begin{array}{cc} 0 & 1/2 \\ 1/3 & 0 \end{array}\right] \quad M_{1,2} = \left[\begin{array}{cc} 0 & 0 \\ 1 & 1/2 \end{array}\right] \quad M_{2,1} = \left[\begin{array}{cc} 1/3 & 0 \\ 1/3 & 1/2 \end{array}\right] \quad M_{2,2} = \left[\begin{array}{cc} 0 & 1/2 \\ 0 & 0 \end{array}\right]$$

These are stored as $(1 : (1/2, 2))$, $(2 : (1/3, 1))$, as $(2 : (1, 3), (1/2, 4))$, as $(3 : (1/3, 1))$, $(4 : (1/3, 1), (1/2, 2))$, and as $(3 : (1/2, 4))$.

$\langle id_4, g(1) \cdot \frac{1}{3} + g(2) \cdot \frac{1}{2} \rangle$