Perspektiverende Datalogi

Internetalgoritmer

MapReduce

Gerth Stølting Brodal

MapReduce Implementationer

Dean, F. and Ghemawat, S. (2004) *MapReduce: Simplified Data Processing on Large Clusters*. In: Sixth Symposium on Operating System Design and Implementation (OSDI 2004): 137-150

MapReduce

Google™

Hadoop

Apache open source projekt

YAHOO!®

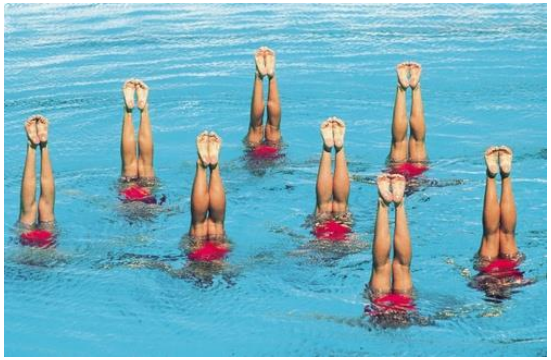
facebook

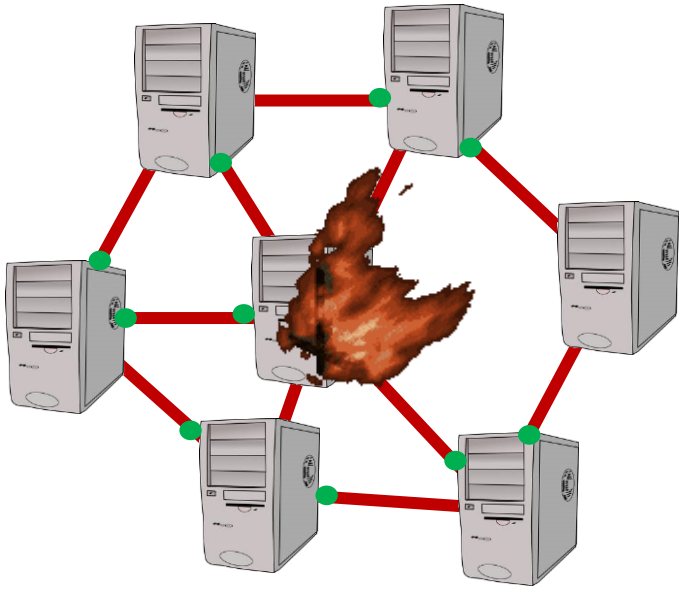
amazon.com®

Parallelle algoritmer – Teori vs Praksis

- Behandling af massiv data kræver parallelle algoritmer (fx Google)

- **Ideelle Teoretiske Verden**
  - mange maskiner samarbejder synkront
  - kan kommunikere i hvert beregningsskridt
  - utallige teoretiske algoritmer udviklet

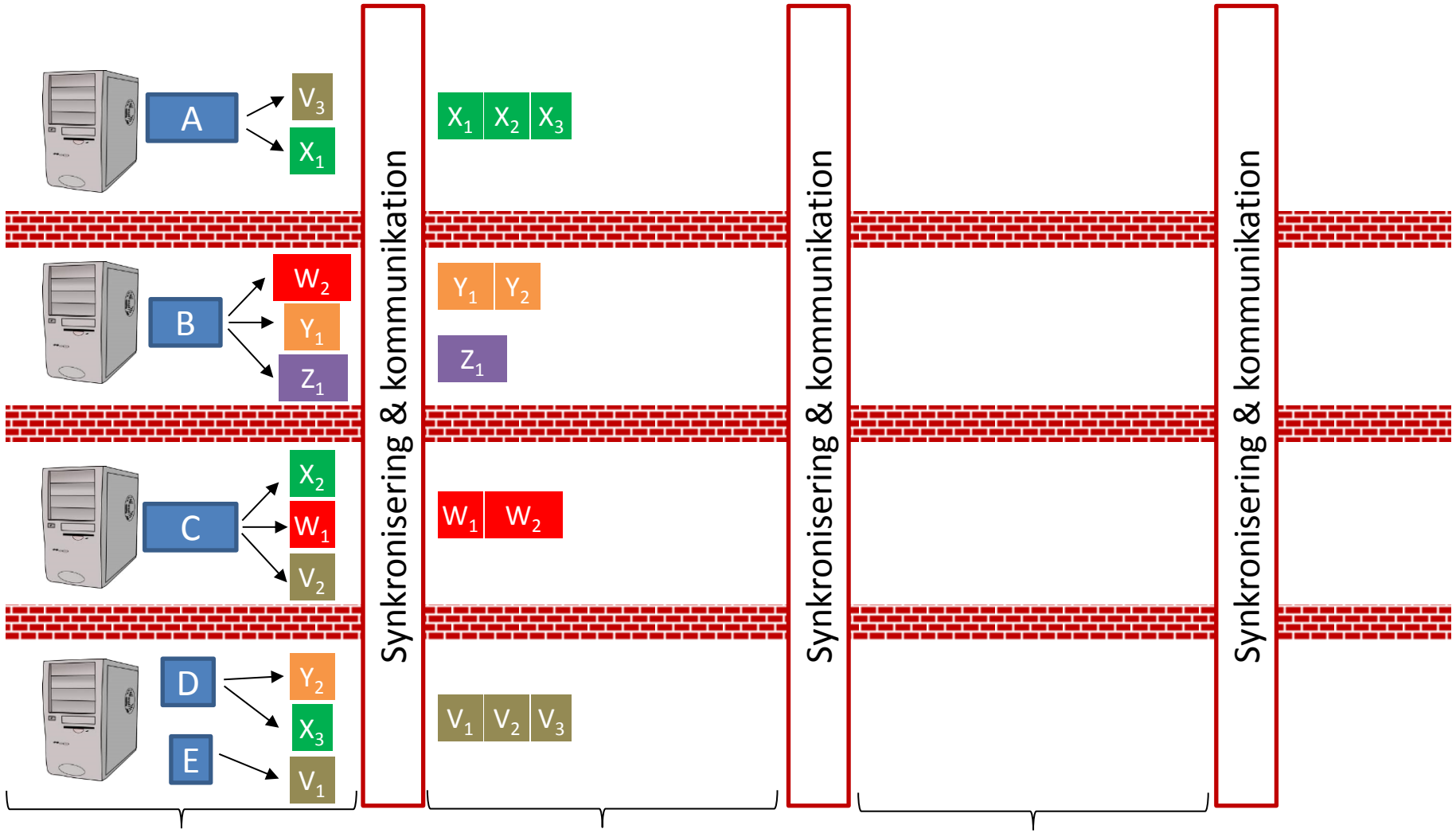




- **Praksis**
  - meget svært at programmere
  - korrekthed, concurrency problemer, …
  - maskiner arbejder med forskellige hastigheder, strejker, bryder sammen

Ønskes: Simpel men slagkraftig parallel model

Afgrænsning af Kommunikation



Parallelle Programmer

- Traditionelt specielt udviklede programmer for hvert problem man ønsker at løse
  - mange ikke-trivielle detaljer i parallel programmer
  - fejl-tolerance (1000'er af maskiner fejler regelmæssigt)
  - fordeling af opgaver blandt maskiner
  - balancering af arbejdet blandt maskiner
- **MapReduce** interfacet ( , 2003)
  - håndter ovenstående automatisk
  - meget begrænsede kommunikation mellem maskiner
  - algoritmen udføres i Map og Reduce faser

MapReduce

- Brugeren skal definere to funktioner

map v1 → List[(k,v2)]

reduce (k, List[v2]) → List[v3]

- Eksempel: Antal forekomster af ord i en tekstsamling

map
- (”www.foo.com”, ”der var en gang en...”) → [(”der”, ”1”), (”var”, ”1”), (”en”, ”1”), (”gang”, ”1”), (”en”, ”1”)...]
- (”www.bar.com”, ”en lang gang...”) → [(”en”, ”1”), (”lang”, ”1”), (”gang”, ”1”), ...]

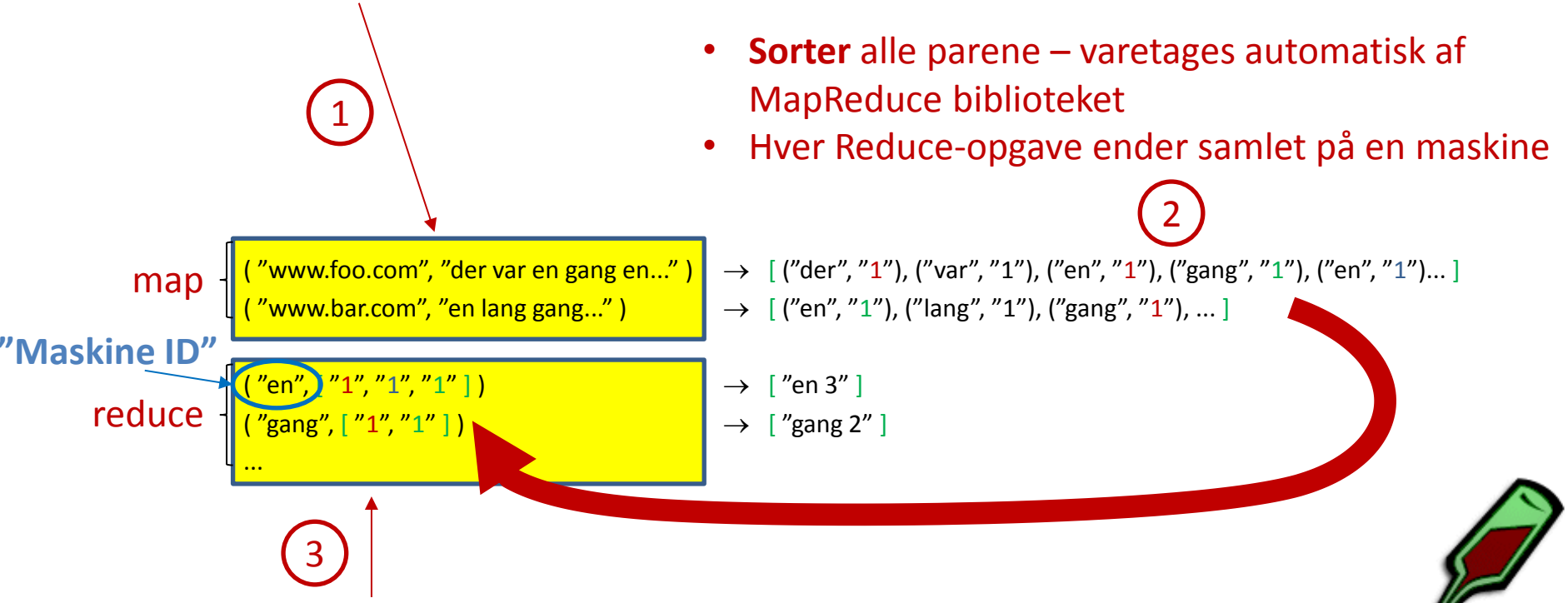
reduce
- (”en”, [”1”, ”1”, ”1”]) → [”en 3”]
- (”gang”, [”1”, ”1”]) → [”gang 2”]
- ...

- Output fra en map-reduce kan være input til den næste map-reduce

MapReduce

Master = en maskine der
1. planlægger og fordeler opgaverne
2. genstarter hængene opgaver på andre maskiner

- Map-opgaverne ligger spredt ud på maskinerne (i et GFS hvor data typisk er replikleret 3 gange)
- Hver maskine udfører et antal uafhængige map-opgaver



- **Sorter** alle parene – varetages automatisk af MapReduce biblioteket
- Hver Reduce-opgave ender samlet på en maskine

- En reduce-opgave løses sekventielt på én maskine (mulig FLASKEHALS)
- Hver maskine udfører et antal uafhængige reduce-opgaver
- Output er en delliste af det samlede output

Antagelse Antal maskiner $\leq n^{1-\varepsilon}$, hver maskine hukommelse $\leq n^{1-\varepsilon}$

Hadoop : WordCount

Java kode fra tutorialen på hadoop.apache.org

```
public class WordCount {
  public static class Map extends MapReduceBase
  implements Mapper<LongWritable, Text, Text, IntWritable> {
    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();
    public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output,
                    Reporter reporter) throws IOException {
      String line = value.toString();
      StringTokenizer tokenizer = new StringTokenizer(line);
      while (tokenizer.hasMoreTokens()) {
        word.set(tokenizer.nextToken());
        output.collect(word, one);
      }
    }
  }

  public static class Reduce extends MapReduceBase
  implements Reducer<Text, IntWritable, Text, IntWritable> {
    public void reduce(Text key, Iterator<IntWritable> values,
                OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
      int sum = 0;
      while (values.hasNext()) {
        sum += values.next().get();
      }
      output.collect(key, new IntWritable(sum));
    }
  }
  ...
}
```

En søgemaskines dele

Indsamling af data

- Webcrawling (gennemløb af internet)

Indeksering data

- Parsning af dokumenter
- Leksikon: indeks (ordbog) over alle ord mødt
- Inverteret fil: for alle ord i leksikon, angiv i hvilke dokumenter de findes

Søgning i data

- Find alle dokumenter med søgeordene
- Rank dokumenterne

Inverteret fil

Input: List[(URL, tekst)]

Output: List[(Ord, URL'er)]

Map

(URL, tekst) $\rightarrow$ [(ord_1, URL),..., (ord_k, URL)]

Reduce

(ord, [URL_1,..., URL_m]) $\rightarrow$ [(ord, "URL_1,...,URL_m")]

Indgrad af alle siderne

Input: List[(i, j)]

Output: List[$(i, \text{indgrad}(i))$]

Map

$(i, j) \rightarrow [\ (j, 1)\ ]$

Reduce

$(i, \underbrace{[\ 1, 1, \ldots, 1\ ]}_{k = \text{indgrad}(i)}) \rightarrow [\ (i, k)\ ]$

Sum

Input: $[\ x_1,\ldots,x_n\ ]$

Ouput: $[\ \mathrm{sum}(x_1,\ldots,x_n)\ ]$

1

Map

$x_i \rightarrow [\ (\mathrm{random}(1..R), x_i)\ ]$

Reduce

$(r, [\ x_{i_1},\ldots,x_{i_k}\ ]) \rightarrow [\ x_{i_1}+\cdots+x_{i_k}\ ]$

$R \approx$ antal maskiner

2

Map

$x_i \rightarrow [\ (0, x_i)\ ]$

Reduce

$(0, [\ x_1,\ldots,x_n\ ]) \rightarrow [\ x_1+\cdots+x_n\ ]$

Øvelser - PageRank

Input: List[(i, j)]

Ouput: List[$(i, p_i^{(s)})$]

$$p_i^{(s)} = 0.85 \cdot \sum_{j:j\to i} \frac{p_j^{(s-1)}}{\text{udgrad}(j)} + 0.15 \cdot \frac{1}{n}$$

$$p_1^{(0)} = 1.0 \qquad p_2^{(0)} = \cdots = p_n^{(0)} = 0.0$$

PageRank

$$p_1^{(0)} = 1.0 \qquad p_2^{(0)} = \cdots = p_n^{(0)} = 0.0 \qquad p_i^{(s)} = 0.85 \cdot \sum_{j\,:\,j \to i} \frac{p_j^{(s-1)}}{\text{udgrad}(j)} + 0.15 \cdot \frac{1}{n}$$

$[\ (i_1, j_1), (i_2, j_2), \ldots\ ]$

① → $[\ (i_1, j_1, p_{i_1}^{(0)}), (i_2, j_2, p_{i_2}^{(0)}), \ldots\ ]$

② → $[\ (i_1, j_1, p_{i_1}^{(0)}, \text{udgrad}(i_1)), (i_2, j_2, p_{i_2}^{(0)}, \text{udgrad}(i_2)), \ldots\ ]$

③ → $[\ (i_1, j_1, p_{i_1}^{(0)}, \text{udgrad}(i_1), n), (i_2, j_2, p_{i_2}^{(0)}, \text{udgrad}(i_2), n), \ldots\ ]$

s {

④ → $[\ (i_1, j_1, p_{i_1}^{(1)}, \text{udgrad}(i_1), n), (i_2, j_2, p_{i_2}^{(1)}, \text{udgrad}(i_2), n), \ldots\ ]$

④ → $[\ (i_1, j_1, p_{i_1}^{(2)}, \text{udgrad}(i_1), n), (i_2, j_2, p_{i_2}^{(2)}, \text{udgrad}(i_2), n), \ldots\ ]$

...

④ → $[\ (i_1, j_1, p_{i_1}^{(s)}, \text{udgrad}(i_1), n), (i_2, j_2, p_{i_2}^{(s)}, \text{udgrad}(i_2), n), \ldots\ ]$

⑤ → $[\ (i_1, p_{i_1}^{(s)}), (i_2, p_{i_2}^{(s)}), \ldots\ ]$

Hint ①, ② og ⑤ er nemmest