

Распределённые объектные технологии: **Модель MapReduce**

Д. А. Усталов

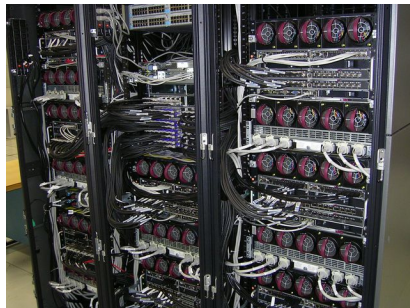
УрФУ и ИММ УрО РАН

12 апреля 2016 г.

- В чём разница между *распределёнными* и *параллельными* вычислениями?
- Разница состоит в масштабе: для распределённых вычислений нужно хотя бы два компьютера.

Распределённые вычисления

- Задача разбивается на «независимые» блоки.
- Реализуется ПО для их вычисления и согласования.
- Система управления задачами запускает их на кластере.



- **Разделяемая память (OpenMP)**
 - Потоки обращаются к одному и тому же блоку памяти.
 - Требуется захват управления и блокировки.
- **Передача сообщений (MPI)**
 - Процессы обмениваются сообщениями.
 - Требуются механизмы обмена сообщениями.

Сеть — медленный элемент вычислительной системы.

<http://www.cs.cornell.edu/projects/ladis2009/talks/dean-keynote-ladis2009.pdf>

- Как избежать сетевых задержек?
- Минимизировать количество этапов коммуникации.
- Обработать данные по месту их расположения.

Перед компанией Google стояла задача построения поискового индекса и ранжирования веб-страниц со всего Интернета.

- Большие объёмы данных.
- Неудобство программирования.
- Поломки оборудования.

<http://www.datacenterknowledge.com/archives/2008/05/30/failure-rates-in-google-data-centers/>

- Работа с парами «ключ-значение».
- Обработка данных в два этапа: *map* и *reduce*.
- Требуется специальная инфраструктура.

Функции *map* и *reduce*

$$\text{map}(\mathbf{k}_1, \mathbf{v}_1) \rightarrow \text{list}(\mathbf{k}_2, \mathbf{v}_2)$$

$$(d_1, \text{Hello}) \rightarrow [(\text{Hello}, 1)]$$

$$(d_2, \text{Hello}) \rightarrow [(\text{Hello}, 1)]$$

$$(d_2, \text{there}) \rightarrow [(\text{there}, 1)]$$

$$\text{reduce}(\mathbf{k}_2, \text{list}(\mathbf{v}_2)) \rightarrow \text{list}(\mathbf{v}_3)$$

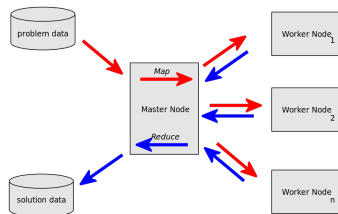
$$[(\text{Hello}, 1)$$

$$(\text{Hello}, 1) \rightarrow [(\text{Hello}, 2), (\text{there}, 1)]$$

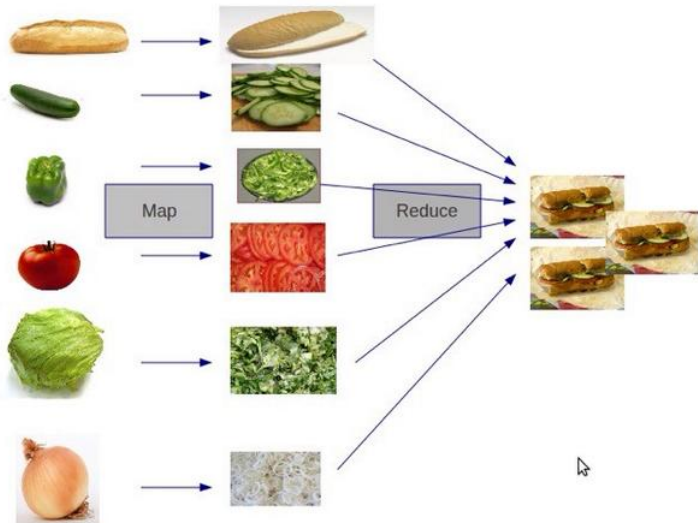
$$(\text{there}, 1)]$$

Этапы работы MapReduce

- Предварительная подготовка для Map.
- Выполнение заданной функции Map.
- Распределение результатов Map по процессорам Reduce.
- Выполнение заданной функции Reduce.
- Запись финального результата.

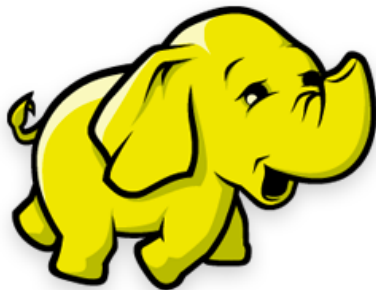


Понятная аналогия



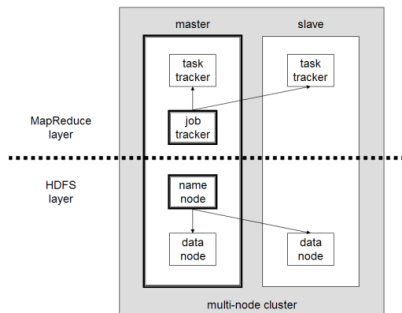
<https://twitter.com/tgrall/status/520810627867348992>

- Apache Nutch — открытый поисковой движок на основе Apache Lucene.
- Возникла необходимость использовать модель MapReduce для аналогичных задач.
- Название в честь игрушечного слоника.



Apache Hadoop

- Фреймворк для построения распределённых приложений.
- Включает ядро, MapReduce и HDFS.
- Написан на Java, но можно использовать любой другой язык.



Пример работы

```
$ hdfs dfs -put file.txt
$ hdfs dfs -ls -R
Found 3 items
drwx----- - hdfs supergroup .Trash
drwx----- - hdfs supergroup .staging
-rw-r--r-- 3 hdfs supergroup file.txt
$ hdfs dfs -cat file.txt
privet))
```

Пример работы

```
$ hadoop jar app.jar args...
```

```
Starting Job
```

```
15/03/11 23:47:52 INFO client.RMProxy: Connecting to Resource
```

```
15/03/11 23:47:53 INFO input.FileInputFormat: Total input pat
```

```
15/03/11 23:47:53 INFO mapreduce.JobSubmitter: number of spli
```

```
15/03/11 23:47:54 INFO mapreduce.JobSubmitter: Submitting to
```

```
...
```

Вычисление числа π методом квази-Монте-Карло.

- `QuasiMonteCarlo.java`
- `hadoop jar hadoop-mapreduce-examples.jar \`
`pi 16 1000`

Демонстрация работы II

Вычисление количества слов в корпусе текстов.

- WordCount.java

- ```
hadoop jar hadoop-mapreduce-examples.jar \
 wordcount \
 input \
 output
```

“Hadoop is a distributed system for counting words.”



Минимальный пример работы Hadoop Streaming.

```
■ hadoop jar hadoop-streaming.jar \
 -input input/ \
 -output output/ \
 -mapper /bin/cat \
 -reducer /usr/bin/wc
```

- После каждого этапа вычисления осуществляется запись в файловую систему.
- Модель MapReduce накладывает некоторые ограничения на спектр решаемых задач.
- Развита экосистема: машинное обучение (Mahout), обработка графов (Giraph), и т. д.
- Ускорение при помощи вычислений в памяти (Ignite).
- Развёртывание и администрирование Hadoop — нетривиальная и *очень* трудозатратная операция.

Извлечь  $n$ -граммы из корпуса текстов и для каждой триграммы вычислить значение  $p(w_3|w_1, w_2) = \frac{f(w_1, w_2, w_3)}{f(w_1, w_2)}$ , где  $f(\cdot)$  — количество появлений  $n$ -граммы в корпусе.

- Использовать набор данных `rus_web_2002_300K-text` из [Leipzig Corpora Collection](#).
- Известно, что  $f(w_1, w_2, w_3) \leq f(w_1, w_2)$ .
- Итоговый результат представить в виде  $(w_1, w_2, w_3, f(w_1, w_2, w_3), f(w_1, w_2), p(w_3|w_1, w_2))$ .
- Следует помнить, что Hadoop сортирует ключи между этапами Map и Reduce.

Спасибо за внимание!

## Дмитрий Усталов

 <https://linkedin.com/in/ustalov>

 <http://kvkt.urfuclub.ru/courses/dot/>

 <https://telegram.me/doturfu>

 [dmitry.ustalov@urfu.ru](mailto:dmitry.ustalov@urfu.ru)