FFmpeg-MR

Distributed Video Transcoding using MapReduce

Background: Video Transcoding Problem

(Compression)

- What is it?
- Why do we need to do it?
  - *Performance*
  - *Compression/File Size*
  - *Quality*
- Why is it more important now than it ever has been?

Background: MapReduce

- What?!
  - **Map** and **Reduce**
  - Influence from Functional Programming (although not quite the same!)
    - Google popularised it in 2004 for querying **large data sets** on **clusters** of computers
- Other uses?
  - Video Transcoding?
- The ‘Cloud’

Project:
MapReduce based Audio and Video Transcoding

- Feasibility study
  - Can we use MapReduce for more than just large data processing?
  - Is video transcoding a good candidate for MapReduce?
    - Is the performance increase close to being linear w.r.t the number of nodes?
    - Is the problem easily split up into units of work?
    - Can the ***Reduce*** function be used effectively?

High-level solution walkthrough

Input Demux **Map** **Reduce** *Merge*

Sort

Input

Demux

Map

Sort

Reduce

Merge

Input

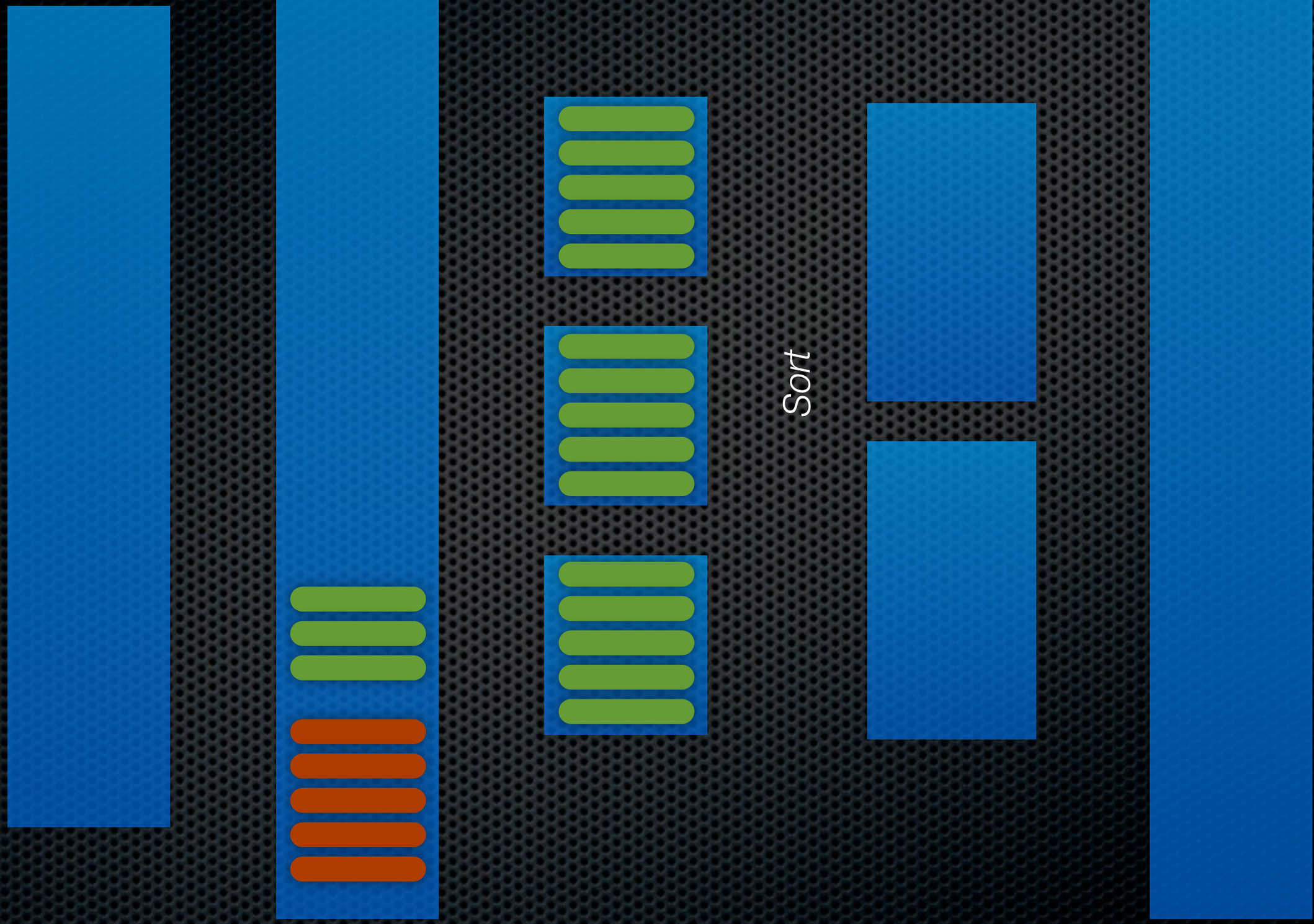
Demux

Map

Sort

Reduce

Merge



Input

Demux

Map

Reduce

Merge

Sort

Input

Demux

Map

Reduce

Merge

Sort

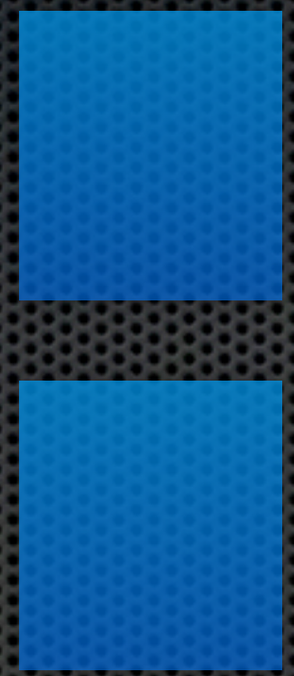
Input

Demux

Map

Reduce

Merge



So

Low Level Implementation Summary

- MapReduce is only a ***programming paradigm***.

- Apache **Hadoop** = open source implementation of MapReduce.
  - Manages the assignment of the *Map*, *Sort* and *Reduce* tasks.
  - Manages the replication and distribution of the data (if you want it to).

Low Level Implementation Summary

- FFmpeg libav* libraries are used for working with the video files (hence the project name).
  - Good performance, codec support and quality.
- But:
  - FFmpeg = **C**.
  - Hadoop = **Java**.
- So there is a layer of custom JNI (Java Native Interface) in between that deals with this conversion. *This does have some performance trade offs.*

Project Analysis

- **Performance Tests**
  - Time taken as number of nodes increases on various different size files, compared to on single machine FFmpeg run.
- **File Size/Compression Comparison Test**
  - Compared to input, and a control FFmpeg run.
- **Objective Quality Tests**
  - PSNR (peak signal to noise ratio) compared to input, and a control FFmpeg run.
  - SSIM (structural similarity index) compared to input, and a control FFmpeg run.
- **Subjective User Study Quality Tests**
  - Can users tell the difference between the distributed output, and one generated by FFmpeg on a single machine?

Progress and Conclusion

- Demux and (Map) implementation are feature complete.
- Reduce ongoing.
- (Testing on Amazon EMR (Elastic MapReduce) works well.)

- Solution looks to be technically feasible.
- Analysis will reveal if performance and compression is good in the coming weeks.

Questions?

http://www.flickr.com/photos/ryanhadley/3369519392/

A take-off of xkcd's Compiling (http://xkcd.com/303).