



# FFmpeg-MR

*Distributed Video Transcoding using MapReduce*

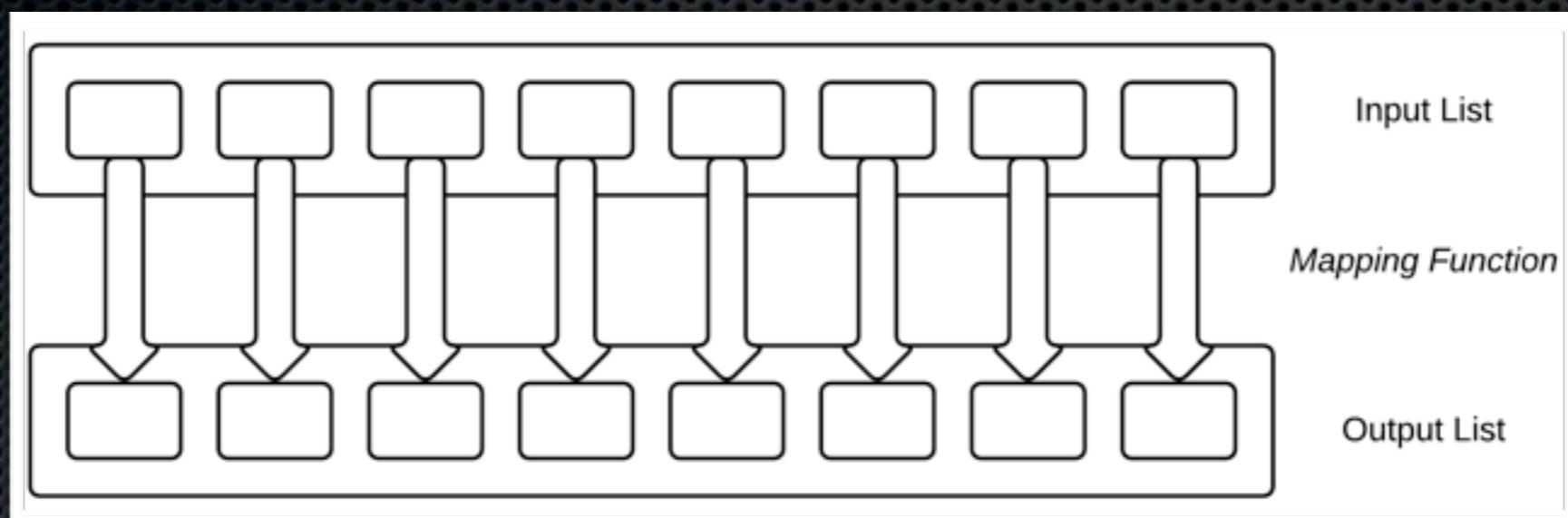
# Background: Video Transcoding Problem

(Compression)

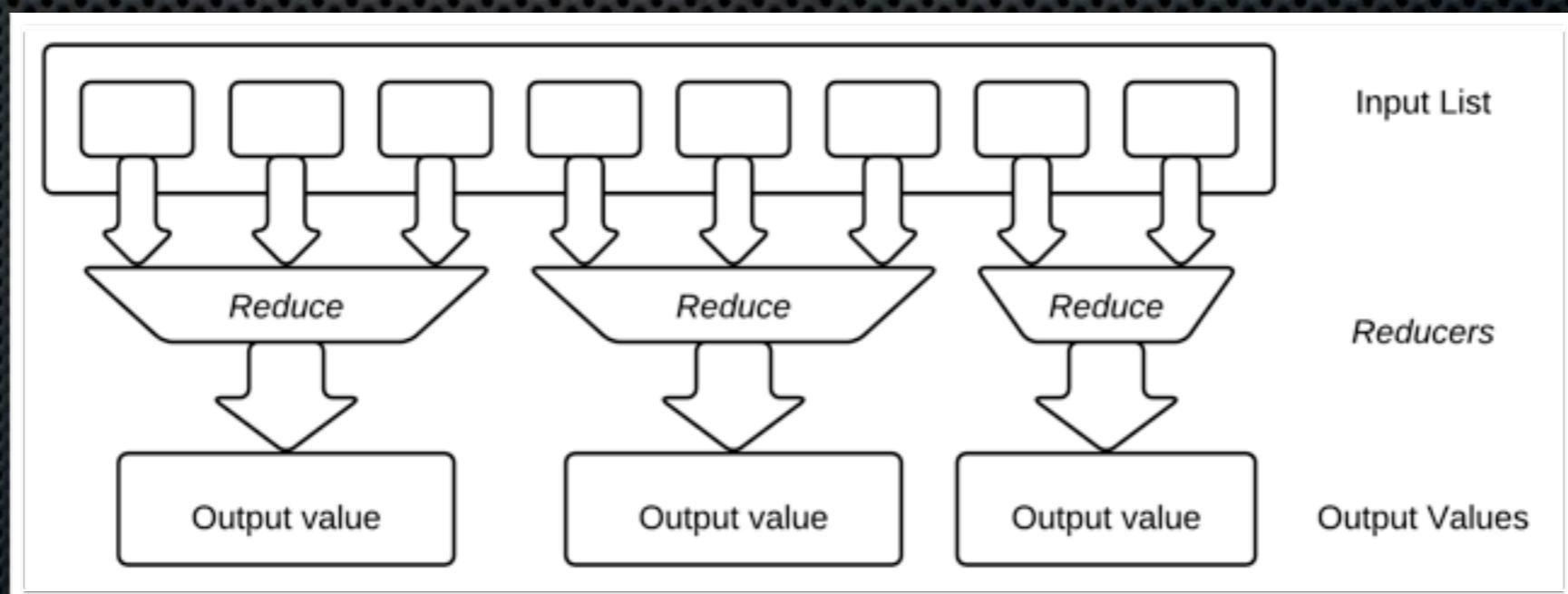
- What is it?
- Why do we need to do it?
- Tradeoffs:
  - *Performance*
  - *Compression/File Size*
  - *Quality*

# Background: MapReduce

Map



Reduce



# *Project: MapReduce Audio and Video Transcoding*

- Can the *MapReduce* paradigm be used for Video Transcoding to:
  - Scale well across machines?
  - Have comparable output quality to current sequential encoders?
  - Improve performance overall?

# Project Deliverables

- *MapReduce* Video Transcoding Solution
  - Runs on a cluster hosted in ‘the cloud’, scaling up to 20 nodes.
  - Supports multiple input formats, and a single output format
  - Java API for controlling submission of multiple transcodes from a single file with varying outputs, with a well defined file format to describe a ‘Transcode Job’
  - GUI for monitoring the progress of jobs

# Project Analysis

- **Performance Tests**

- Time taken as number of nodes increases on various different size files, compared to a single machine.

- **Quantitive Quality Test**

- PSNR (peak signal to noise ratio) compared to input, and a single machine output.
  - Two types of encoding AB (file size focus) and CRF (quality focus).

- **Qualitative User Study**

- Can users tell the difference between the distributed output, and one generated on a single machine?

- **File Size/Compression Comparison Test**

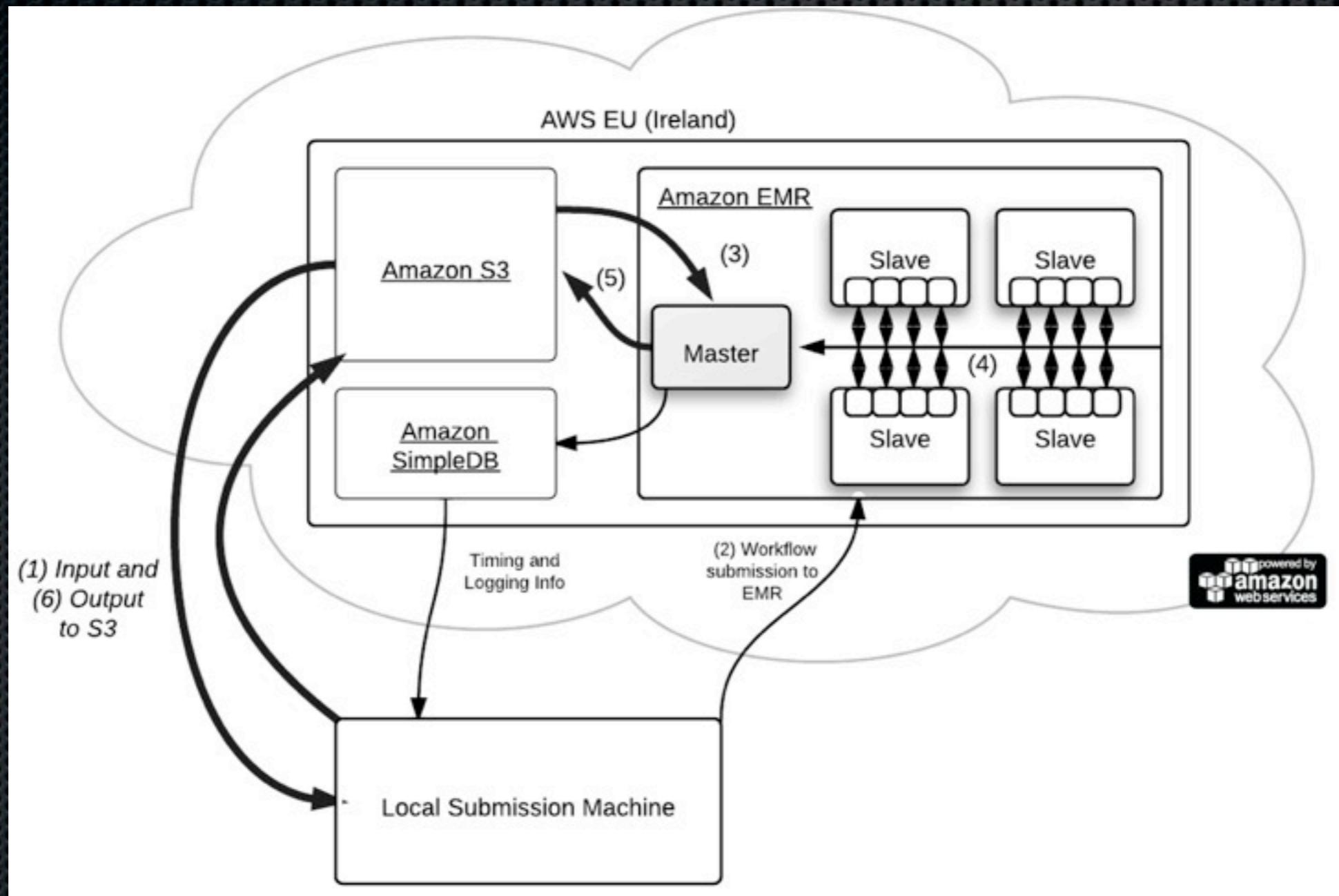
- Compared to input, and a single machine output.

# Demo

*Single Machine vs ‘The Cloud’*

*Sequential vs Parallel MapReduce*

# Cloud Architecture



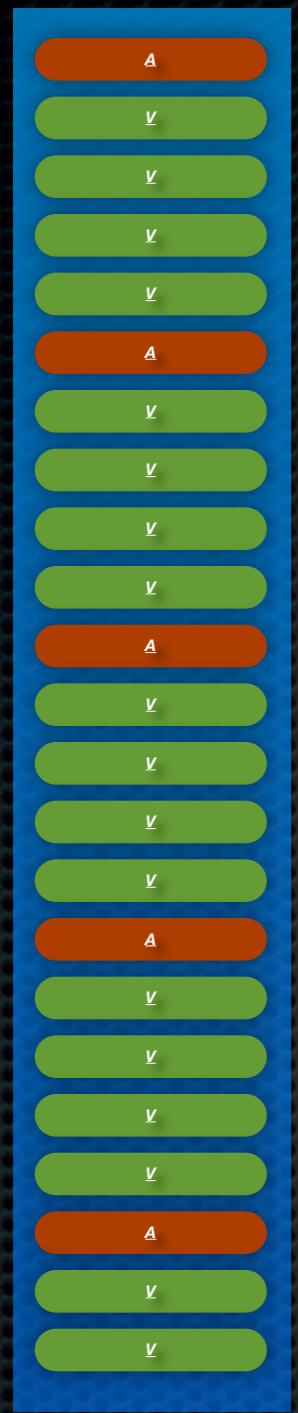
*Input*

Demux

Map

Reduce

Merge



Start -> End of File

*Partition*

*Input*

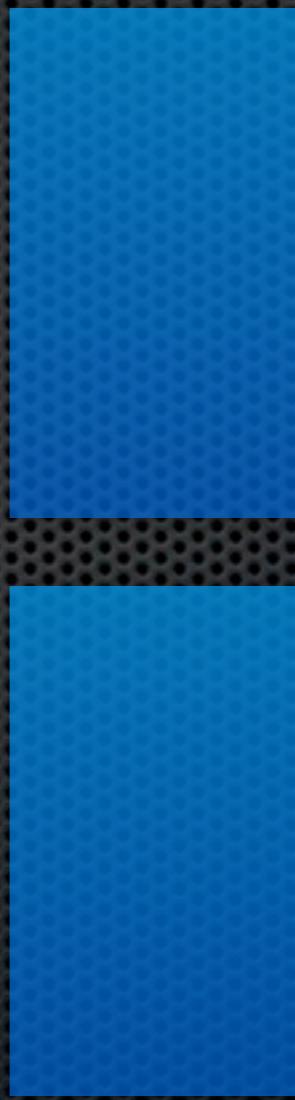
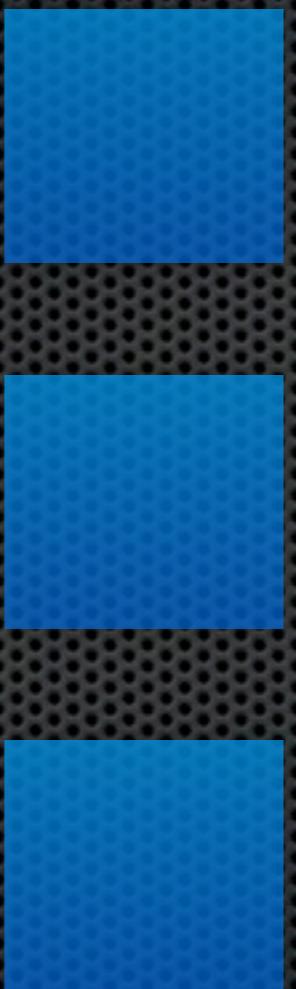
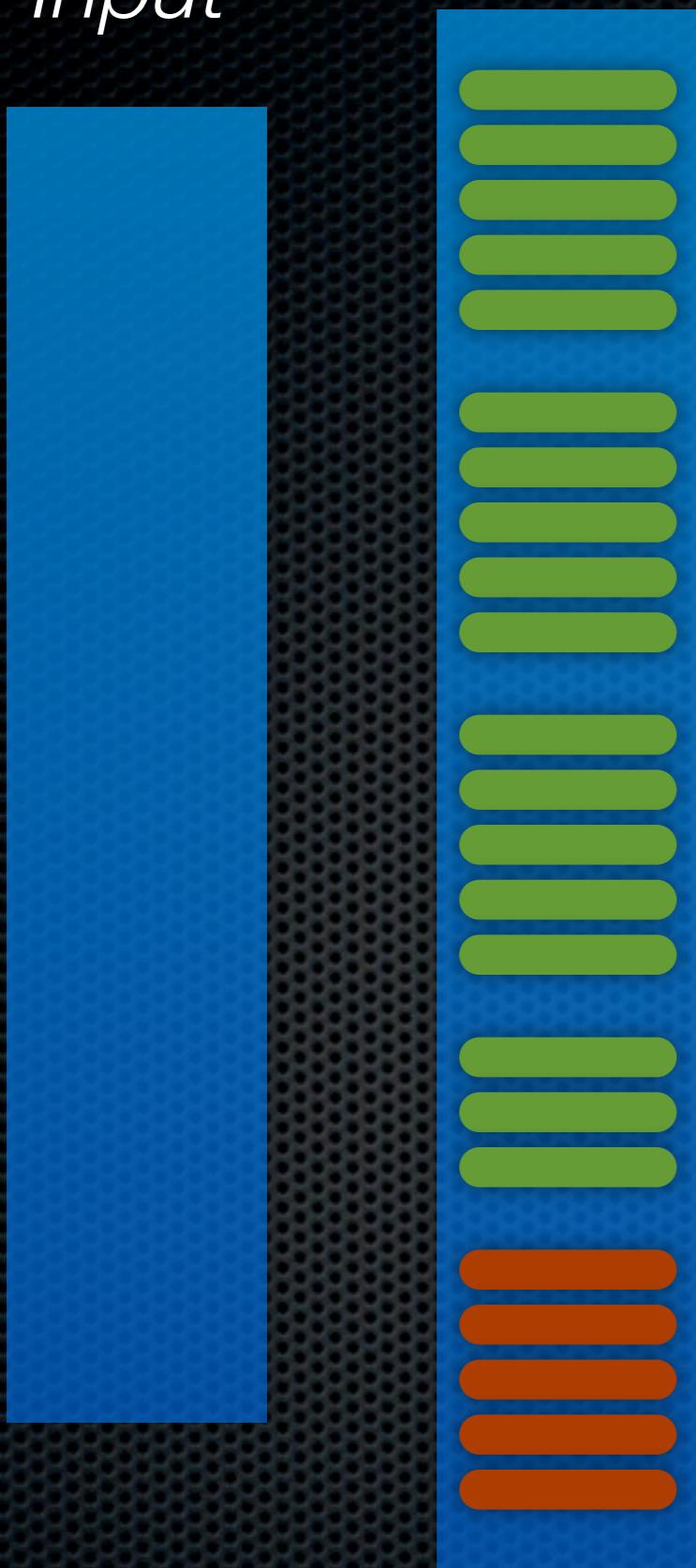
Demux

Map

Reduce

*Merge*

*Partition*



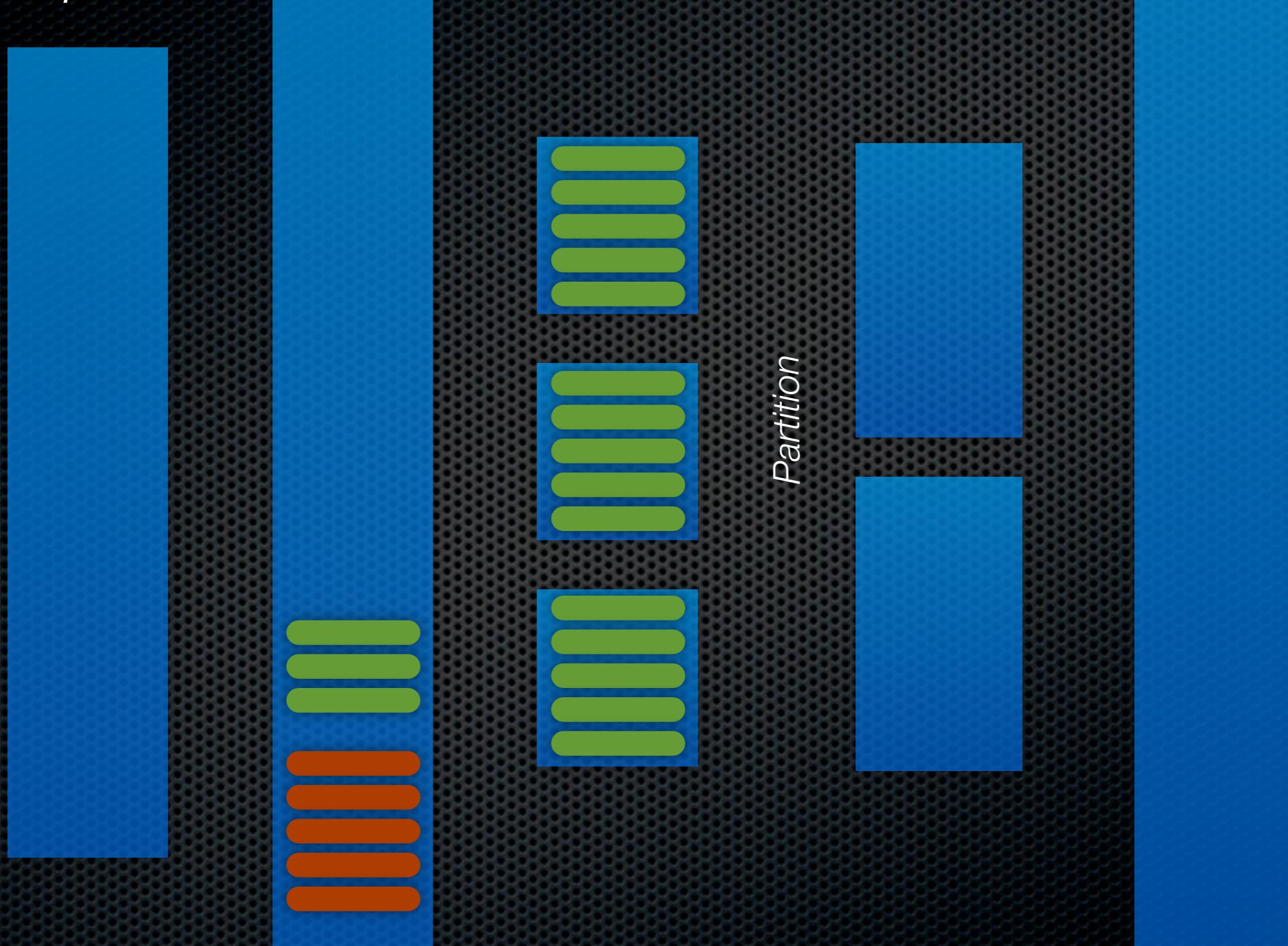
*Input*

Demux

Map

Reduce

*Merge*



*Input*

Demux

Map

Reduce

*Merge*

*Partition*



*Input*

Demux

Map

Reduce

Merge

*Partition*



*Input*

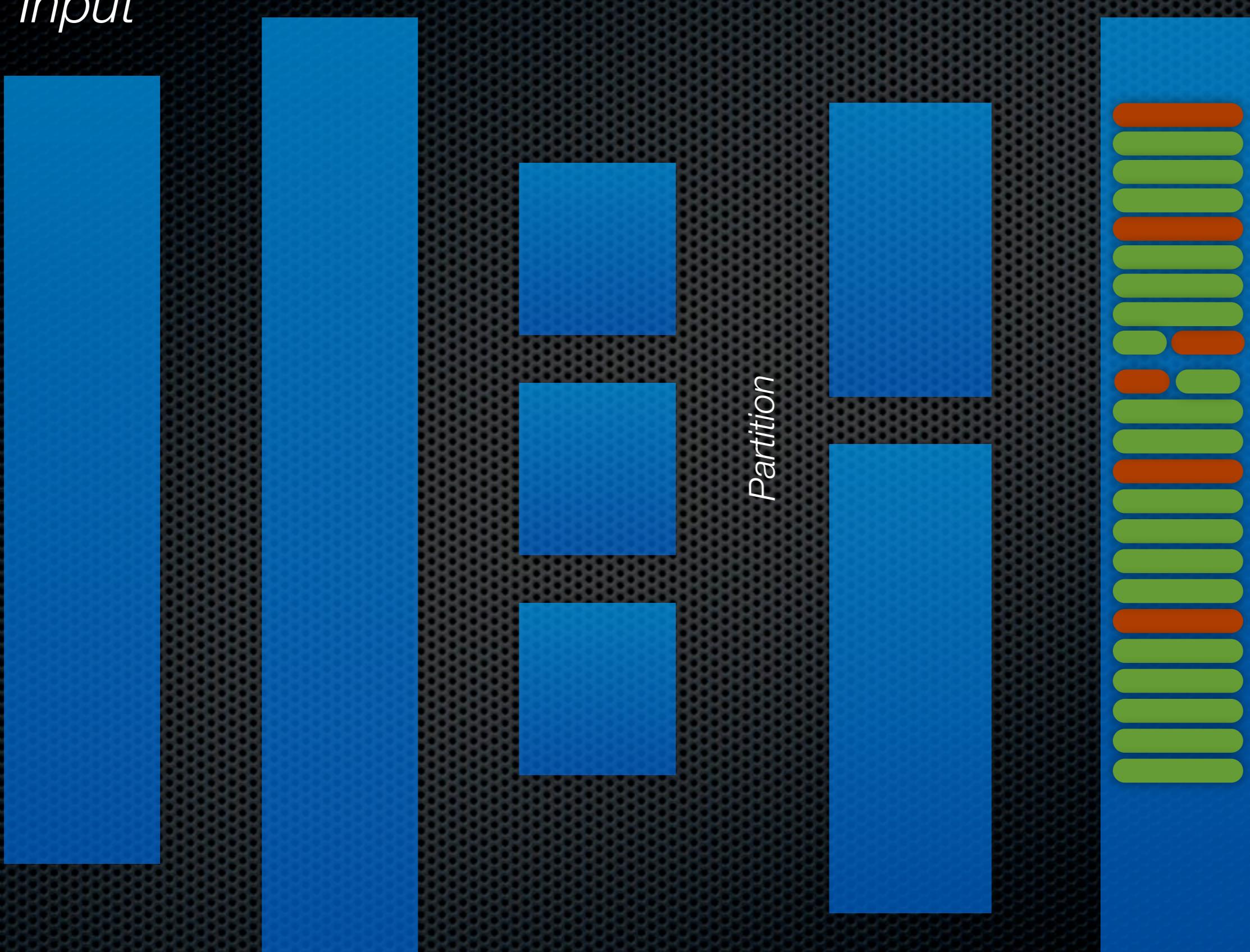
Demux

Map

Reduce

*Merge*

*Partition*



*Turn off the cluster...*