# Intro to batch processing

## STREAMING CONCEPTS

**Mike Metzger**
Data Engineer

# What is batch processing?

- Processing data in **groups**
- Runs **from start** of process **to finish**
  - No data added in between
- Typically run as result of
  - an **interval**
  - starting **event**
- Processed in a certain **size** (batch size)
- An instance of a batch process is often referred to as a **job**

# Common batch processing scenarios

- **Reading** files or parts of files (text, mp3, etc)
- **Sending** / **receiving** email
- **Printing**

# Why batch?

- **Simple**
- Generally **consistent**
- Multiple ways to improve **performance**

# Let's practice!

STREAMING CONCEPTS

# Scaling batch processing

## STREAMING CONCEPTS

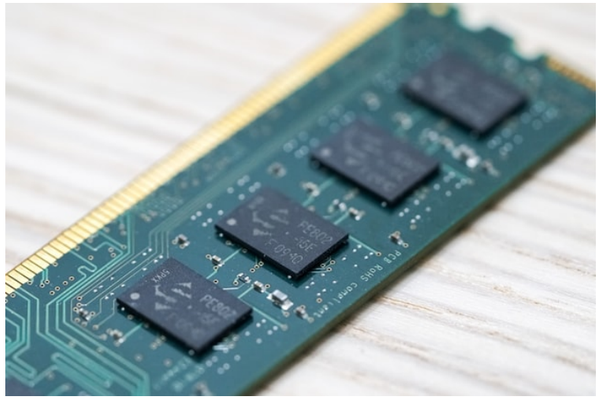
**Mike Metzger**
Data Engineer

# What is scaling?

- Improving **performance**
  - Processing **more quickly**
    - *Less time to process the same amount of data*
  - Processing **more data**
    - *More data processed in the same amount of time*

# Vertical scaling

- **Better computing**
  - Faster CPU
  - Faster IO
  - More memory
- Typically the **easiest** kind of scaling
  - Least complexity
  - Rarely requires changing underlying programs / algorithms



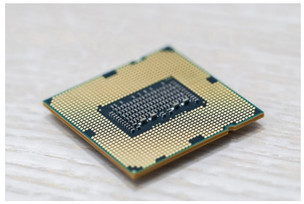
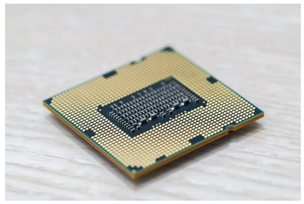
[1] Images courtesy https://unsplash.com/@jeremy0

# Vertical scaling cons

- Inherently **limited**
- Can be **expensive** / low ROI
- Industry improvements are **not guaranteed**

# Horizontal scaling

- **Splitting** a task into multiple parts
  - More computers
  - Could also be more CPUs
- Best done on tasks that are **"embarrassingly parallel"**
  - Tasks that can be easily divided among workers
- Can be very **cost effective**
- Can have **near-linear performance improvements** for certain types of processes

# Horizontal scaling cons

- **Complexity**
  - Requires a processing framework (like Apache Spark or Dask)
  - Requires more extensive networking
- **Ongoing management**
- Can be **expensive** depending on requirements
- **"Non-parallel"** tasks

# Let's practice!

STREAMING CONCEPTS

# Batch issues

STREAMING CONCEPTS

**Mike Metzger**
Data Engineer

# Delays

- Time until **data is ready** to process
  - *Is all data available?*
- Time until **process begins**
  - *When does the next interval start?*
- Time to **process data**
  - *How long until completion?*
- Time until processed data is **available for use**
  - *How long until users can use the data?*

# Example #1

**Waiting on the source data**

- Machines sending log files at times of **low utilization**
- Works ok during **normal** utilization
- **High utilization** would **limit** ability to send logs, potentially hiding issues.

# Example #2

**Waiting on the process**

- **100GB** log files **per day**
- Currently takes **23 hrs** to process
- Approximately 4.4GB/hr
- Grows at 5% per month
- Next month would be **105GB** and take **~24 hrs**
- Following month would be **~110GB** and take **~25** hrs
- **Takes longer than a day to process one day's worth of data!**

# Example #3

**Waiting on the data to be available**

- How long until analytics are **available**?
- Sales report must **wait** for all information to generate
- Sum of delays is **minimum time** to generate new report
  - Amount of time to **collect / prepare** data: **1 day**
  - Time required to **process** data: **7 hrs**
  - Time to **update** systems: **5 hrs**
  - Time to **generate** report: **2 min**
- Total time for each report: **1.5 days**

# Let's practice!

STREAMING CONCEPTS