Comparison of distributed frameworks

Louis Jachiet
What is Big Data?
"Big data" is a field that treats ways to analyze, systematically extract information from, or otherwise deal with data sets that are too large or complex to be dealt with by traditional data-processing application software.

– Wikipedia @ 09/09/2019
The three Vs

Figure 1: source Wikipedia
Big Data depends on the CONTEXT.
Controversy of Big Data

• All data is BIG now
• Hype to sell Hadoop based systems
• Ethical concerns about accessibility
• Limited access to Big Data creates new digital divides
• Statistical Significance:

  When the number of variables grows, the number of fake correlations also grows Leinweber: S&P 500 stock index correlated with butter production in Bangladesh
The six Vs?

- Volume
- Variety
- Velocity
- Value
- Variability
- Veracity
Figure 2: source EMC Digital Universe
Digital Universe

**Figure 3:** source EMC Digital Universe
Figure 4: source EMC Digital Universe
The big data batch family
Hadoop

- Started in 2006
- For very large data
- And for disk-heavy computations
Spark

- Started in 2014
- Fast data processing
- For large data
- Memory-heavy
Flink

- Started in 2011
- Fast data processing with optimizations
- For large data
- bulk-batch and iterative processing (dataflow)
The streaming family
Spark

- Processes data in micro-batches
- High-level
- High throughput-high latency
Flink

- Natively dataflow
- High-level
- Complex windowing
- High throughput-low latency
Storm

- User creates a "topology" with spouts and bolts
- Handles message distribution/tolerance
Kafka

- Publish-subscribe model
- Distributed log
Samza

- Built on top of Kafka
- Handles high-level operations
<table>
<thead>
<tr>
<th>Model</th>
<th>API</th>
<th>Resiliency</th>
<th>State</th>
<th>Window</th>
<th>Guarantee</th>
<th>Latency</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>Native</td>
<td>Native</td>
<td>Batch</td>
<td>Native</td>
<td>Native</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>API</td>
<td>Compositional</td>
<td>Compositional</td>
<td>Declarative</td>
<td></td>
<td>Declarative</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resiliency</td>
<td>ACK</td>
<td>Log</td>
<td>RDD</td>
<td></td>
<td>Checkpoint</td>
<td></td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>with trident</td>
<td>Stateful</td>
<td>DStream</td>
<td></td>
<td>Stateful</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Window</td>
<td>No</td>
<td>No</td>
<td>Time</td>
<td></td>
<td>Policy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guarantee</td>
<td>1+</td>
<td>1+</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Latency</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
<td></td>
<td>Medium</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Throughput</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
<td></td>
<td>High</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Others
• “Meta” language for optimization
• Supports streams and batch
TensorFlow

- Stateful dataflow graphs
- Compatible with various libraries (e.g. keras for machine learning)
- Deploy on single machine / GPU / multiple CPU or GPU