**Map reduce - February 18**

[**VIEW RECORDING - 86 mins (No highlights)**](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc)

**Meeting Purpose**

[Review MapReduce paper and introduce Apache Spark concepts](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=0.0)

**Key Takeaways**

* [MapReduce enables distributed processing of large datasets across clusters](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=20.0)
* [Apache Spark improves on MapReduce by utilizing in-memory processing for faster performance](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=46.0)
* [Spark introduces RDDs (Resilient Distributed Datasets) as a key abstraction for distributed data and computation](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=132.0)
* [Spark provides a richer set of transformations and actions beyond just map and reduce](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=765.0)

**Topics**

**MapReduce Overview**

* [Developed by Google in 2004 to process large datasets](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=68.0)
* [Key features: big data processing, parallelization, fault tolerance, data distribution, load balancing](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=97.0)
* [Programming model: map (k1,v1) -> list(k2,v2), reduce (k2,list(v2)) -> list(v2)](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=139.0)
* [Execution steps: input splitting, master/worker assignment, map execution, intermediate results, sorting, reduce execution, output](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=309.0)

**MapReduce Fault Tolerance**

* [Master node monitors worker status and reassigns failed tasks](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=650.0)
* [Backup tasks launched for slow-running "straggler" nodes](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=800.0)
* [Locality optimization to minimize data transfer across network](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=865.0)

**Motivations for Apache Spark**

* [MapReduce involves frequent disk I/O, which is slow](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=1174.0)
* [Need for interactive queries and iterative processing (e.g. machine learning)](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=1258.0)
* [Decreasing memory costs enable in-memory data storage](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=1714.0)

**Apache Spark Concepts**

* [Resilient Distributed Datasets (RDDs) as key abstraction](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=2256.0)
* [Transformations (e.g. map, filter, join) and actions (e.g. reduce, count, collect)](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=2308.0)
* [In-memory processing for faster performance](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=2830.0)
* [Automatic fault tolerance and straggler handling](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=2870.0)
* [Rich set of operations beyond just map and reduce](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=2920.0)

**Spark Programming Model**

* [Data-centric rather than compute-centric](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=4704.0)
* [Operations run on distributed data, hiding complexity](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=4792.0)
* [Simplified programming abstraction for distributed computing](https://fathom.video/share/yX-ALXqz2nyRiG9EXj8izMU-zYv6uNKc?tab=summary&timestamp=4916.0)

**Next Steps**

* Download and install Apache Spark
* Write report on installation process and initial learnings
* Review Spark programming concepts and RDD abstraction