

Spark based Enterprise Quant Analytics

Enterprise Quant Computation

using

Apache Spark, Hadoop Kafka, together with JEE

Neeraj Singh (nk_singh@yahoo.com)

1.1 Overview & Problem Statement

In the investment banking business, a wide variety of financial products are traded by banks to its customers wherein banks need to apply its strategy of financial engineering and reduce the risks to acquire the most optimal portfolio position.

These trading systems capture the trade executed by front-office while the back-office verifies this trade. In the middle-office, risk and strategy processing are carried out on portfolio comprising of live and historical trades to provide the recommendation for optimal portfolio. The risk and strategic analysis includes Cash-Balance, Liquidity, P&L, VaR, Portfolio Position, Risk Limits, Hedging, Mark-to-Market, Gap Analysis, etc.

To compute the risk and strategic analysis, essentially millions of trades are priced with different yield-curves, sensitivities and risk factors are calculated in order to generate a most optimal portfolio which yields maximum profit while keeping the cost & risk under limit.

To do this processing, generally it takes 8 hours in night batch processing on Solaris based Sun Prime-power 1500, comprising of 5 high-end machines. With Spark based technologies, the time taken could be reduced to 1 hour. Moreover, with in-memory cluster based computation of Spark, we can generate cache in memory after 1 hour of processing to compute the task in real-time and show it on blotter, so that investment banks are enabled to take optimal portfolio position in real-time.

1.2 Project

Build a small prototype using Spark based technologies for strategy and risk management system in domain of investment banking, so that risk and strategy related analysis could be done in real-time in order to be aligned with the optimal portfolio position.

1.3 Process-Flow & Technologies for my Certification Project

Generally trading is done on standardized trading systems like Summit, Murex, Calypso, Sophis, Openlink etc.

My trading system is Summit which acts as OLTP, wherein a client acting as streamer is attached to distribution server so that any new or updated trades are directly forwarded to Kafka messaging system.

Most important point is here the messaging system should support the reliability & integrity through guaranteed delivery, high throughput and distributed messaging capabilities, so I have selected Kafka.

All the new trades are considered as events; therefore on arrival of new trades from OLTP through streamer client, Kafka sends the notification to subscribers. Spark streaming module acts as

subscriber which start processing on notification of new event, essentially new trade. At Spark streaming, portfolio is re-calculated and RDD is updated to be persisted in memory.

Once the portfolio is in RDD, further processing is carried out through financial & mathematical libraries together with Spark ML Lib. Analytical module processes the risk and strategic analysis of portfolio. Some of the analyses are sequential processed only for new trades while non-incremental tasks are processed for whole portfolio.

These results are persisted incrementally to Hbase; thereby a blotter built with Obba (similar to JFreeChart) displays the result with respect to risk and analysis besides providing updates about most optimal portfolio in real time.

+++++