Skalierbare Finanzanalysen unter Verwendung von Big Data mit Spark

Studiengang: Master of Science in Engineering | Vertiefung: Informations- und Kommunikationstechnologien
Betreuer: Prof. Dr. Kurt Stockinger ZHAW, Prof. Dr. Urs Sauter BFH
Experte: Dr. Jörg Behrens (Fintegral Consulting AG)

Die Subprime- und die Euro-Krise haben gezeigt, dass die Risikoanalyse in vielen Fällen der Komplexität der einzuschätzenden Vorgänge nicht gerecht werden konnte. Die Auswirkungen von fehlerhaften Risikoanalysen waren weltweit und über mehrere Jahre spürbar. Eine mögliche Ursache für diese Fehler sind fehlende Standards für Risikobewertungen.

Ausgangslage

Das Projekt ACTUS stellt einen Standard zur Verfügung, um Finanzinstrumente abzubilden. Aus diesen standardisierten Daten (Contracts) können Ereignisse (Events) wie Ratenzahlungen, Rückzahlungen etc. als Datensatz erstellt werden, die unabhängig vom Finanzinstrument immer gleich aufgebaut sind. Entsprechend zuverlässig lässt sich aus ihnen unter Berücksichtigung verschiedener Szenarien eine Risikoanalyse erstellen. In vorgängigen Arbeiten wurde die Berechnung dieser Events bereits auf skalierbaren Systemen implementiert, ohne dabei allerdings eine Performance zu erreichen, die es einem grossen Institut ermöglichen würde, innerhalb einer sinnvollen Frist eine Risikoanalyse zu erstellen.

Zielsetzung

Da die ACTUS-Algorithmen als Java-Bibliotheken zur Verfügung stehen und Spark ebenfalls Java-Programme ausführen kann, wird als These angenommen, dass die Berechnungszeit um die Events zu erstellen, verbessert werden kann, indem die Implementierung von ACTUS in Spark umgesetzt wird. Ziel dieser Arbeit ist es deshalb, herauszufinden, wie viele Verträge pro Sekunde in einem verteilten System (Cluster) verarbeitet werden können, um die Events für eine Risikoanalyse zu erstellen und diese dann auch durchzuführen.

Methoden und Resultate

Unter Berücksichtigung vorliegender Arbeiten werden vier Methoden ausgearbeitet und auf einem Cluster mit acht Knoten à 2 GB RAM und 2 CPUs implementiert. Die Methoden werden in verschiedenen Experimenten getestet, um festzustellen, ob die Ziele erreicht werden können. In der ersten Methode wird gezeigt, dass sich Spark als Cluster eignet, um Daten mit Java zu verarbeiten, und in den Experimenten kann dargestellt werden, dass eine Skalierung über das Cluster stattfindet. In der zweiten Methode wird gezeigt, dass sich die ACTUS-Algorithmen mit Java in Spark implementieren lassen, und in den Experimenten kann dargelegt werden, dass auch hier eine Skalierung stattfindet. In der dritten Methode wird dokumentiert, dass auch mehrere Risikoszenarien verarbeitet werden können. Dafür werden drei Versionen erstellt, die die Daten vor der Verarbeitung auf unterschiedliche Art filtern sollen. Alle drei Versionen werden zuerst mit 100 Risikoszenarien und 10 bis 100 000 Contracts und anschliessend mit 100 Contracts und 10 bis 100 000 Risikoszenarien verarbeitet. In den Experimenten kann kein unterschiedlicher Performanceeinfluss dieser drei Versionen festgestellt werden, dafür ist die Skalierung sowohl über die Contracts als auch über die Risikoszenarien identisch. In der vierten und letzten Methode werden die erstellten Events als Basis für eine Risikoanalyse gebraucht. Dafür werden der Nominal Value, der Mark-to-Market Value und die Funding Liquidity verwendet. In den Experimenten kann ebenfalls eine Skalierung über das Cluster festgestellt werden.

Fazit und Ausblick

Obwohl gute Performancewerte erreicht werden, gibt es bei einer Vergrösserung des Clusters von 4 auf 8 Knoten stets einen Performanceeinbruch. Um dieses Verhalten zu untersuchen, werden spezifische Performancetests durchgeführt, und es stellt sich heraus, dass das gewählte Speicherformat aufgrund seiner Komprimierung sehr CPU-lastig ist, was einen Bottleneck erschafft. In der Schlussfolgerung der Arbeit wird erörtert, dass auf dem Cluster bis zu 15 900 Contracts pro Sekunde verarbeitet werden können, was beinahe 20-mal schneller ist als in ersten Versuchen aus vorgängigen Arbeiten. Im Ausblick wird beschrieben, wie man die Performance durch eine Anpassung der Hardware und die Änderung des Zielformats eventuell noch weiter steigern könnte.

Daniel Bruttel
daniel@bruttel.com