DataWalk Performance Test: Multi-Node Scaling

Executive Summary

In a performance test where the Weakly Connected Components (WCC) graph algorithm was run against data from the Bitcoin blockchain, the DataWalk software platform demonstrated linear scaling through six compute servers, for both data indexing, and for execution of the WCC graph algorithm. The test also reflected that DataWalk can effectively support large volumes of data.

Introduction

DataWalk is designed to effectively support integration and analysis of vast amounts of data. One enabler of this is a scale-out architecture, whereby additional performance can be gained by adding compute servers to a DataWalk cluster.

In this paper we review results from performance testing conducted by DataWalk to evaluate the performance benefits (scalability) of adding additional compute nodes to DataWalk when running the Weakly Connected Components (WCC) graph algorithm against data from the Bitcoin blockchain.

Performance Test and Results

The WCC algorithm is one of the essential tools for pre-processing graphs, providing an effective means of identifying key patterns and structures within large graph data. The algorithm allows for the automatic discovery of the network segments that are interconnected within their own community, yet disconnected from other communities.

Extracting the connected components has several applications in further analytics. These include:

- Providing a comprehensive overview of the network structure by analyzing the distribution of component sizes and examining the proportion of small and large communities
- Serving as an intermediate step for executing other algorithms on the detected components, e.g. label propagation
- Enabling a deeper understanding of individual communities and their properties. This understanding can have practical applications, such as detecting fraud in financial networks or entity resolution by analyzing various community statistics, such as the number of rings, diameter, and maximum ring perimeter.
- Utilizing component statistics or community labels (e.g. “Is the customer part of a community containing fraudulent transactions?”) as graph features for machine learning models or scoring rules

The test data was Bitcoin transactions from the years 2009 through 2021, and this data was represented in graph form with just over three billion nodes and edges (specifically 3.1B). Tests were performed using standard AWS EC2 instances using r6i.4xlarge nodes, each with 128 GB RAM and 16 3.5GHz vCPUs, running DataWalk version 4.1.

Measurements were first done for data indexing time, with results shown in Table 1 below.

Number of compute servers	Indexing Time for 3.1B graph nodes + edges	Equivalent Data Indexing Rate (Objects Per Second)
1	91 minutes	567K
3	38 minutes	1.361M
6	22.5 minutes	2.295M

Table 1. DataWalk indexing time for 3.1B nodes + edges

The above data indicates that the data indexing rate essentially increased linearly through six compute nodes, as shown in Figure 1 below.

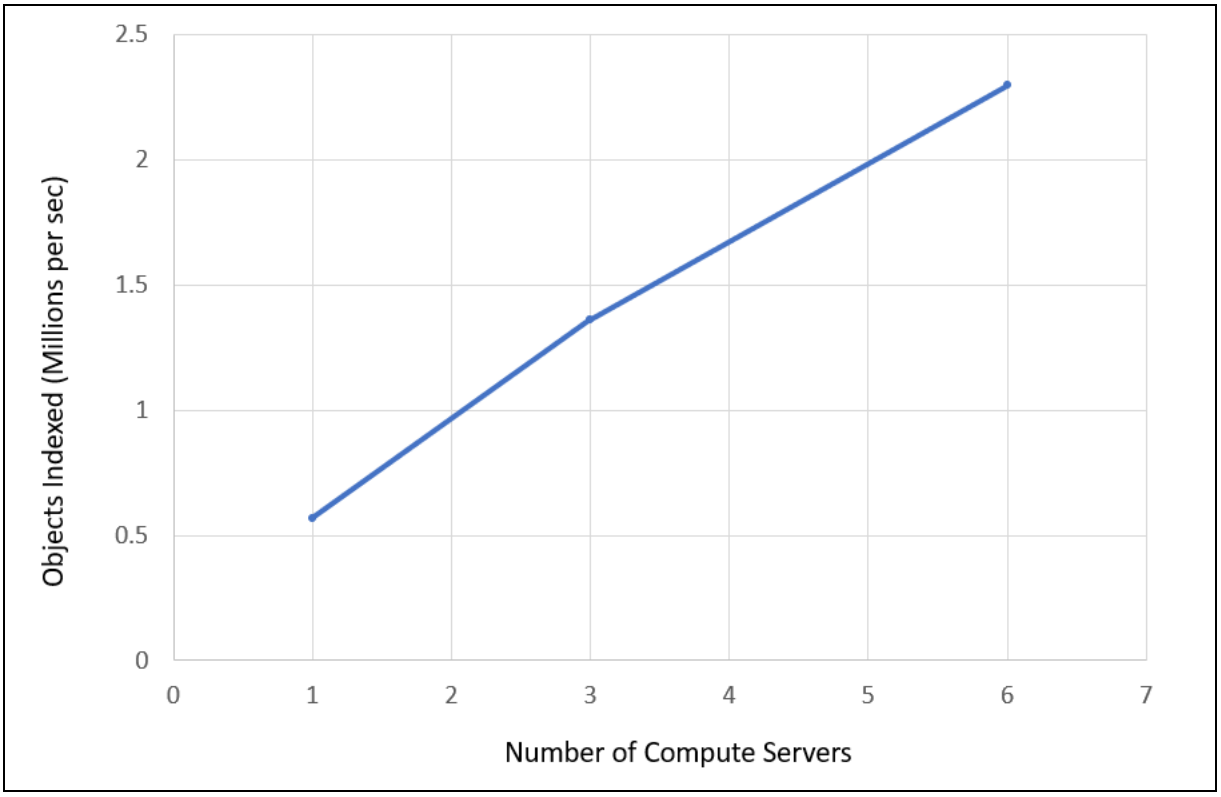

Figure 1. Data indexing rate through six compute nodes

Execution time of the WCC algorithm was also measured with DataWalk as the number of compute servers increased. Results are presented in Table 2 below.

Number of Compute Servers	WCC Execution Time for 3.1B nodes + edges	Equivalent objects executed per second
1	10 minutes	5.2M
3	4.7 minutes	11M
6	2.8 minutes	18.6M

Table 2. WCC execution time for 3.1B nodes + edges

The above data reflects that the rate of execution again increased essentially linearly, as the number of compute nodes was increased from one to six. See Figure 2 below.

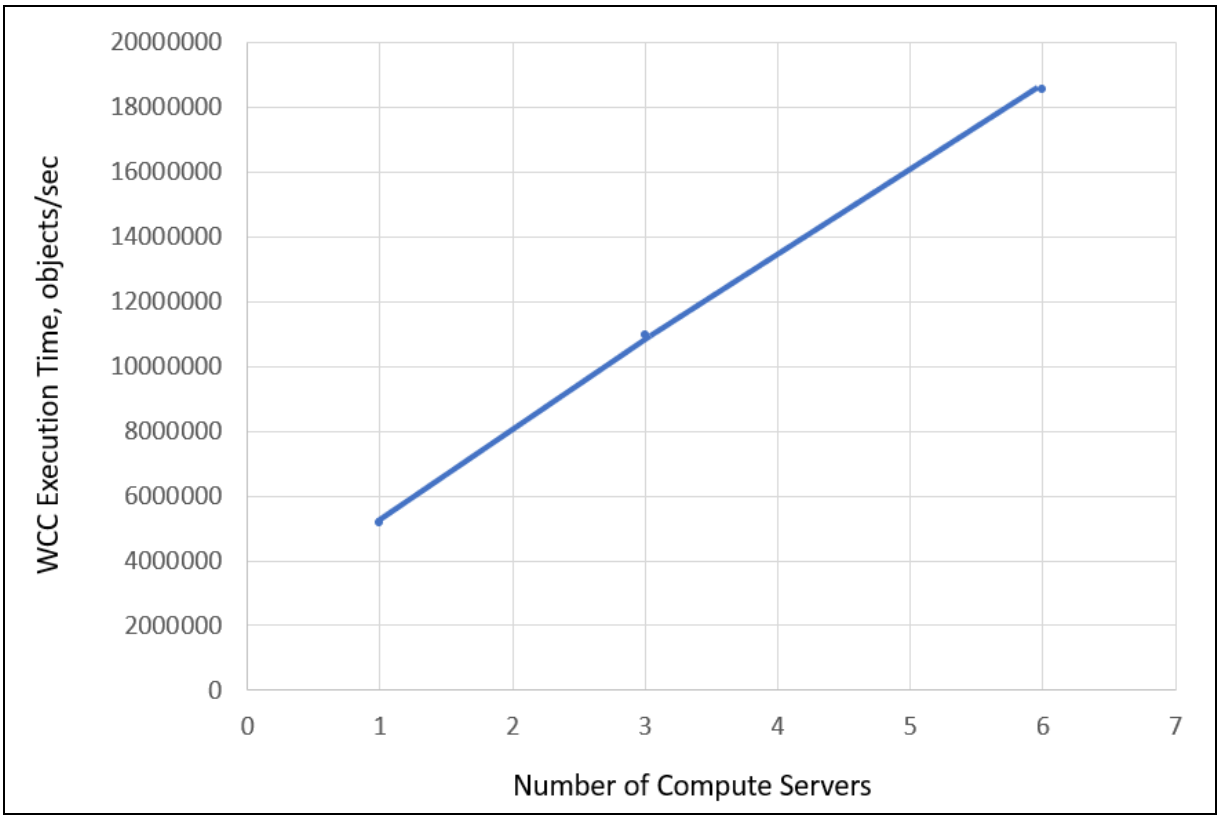

Figure 2. WCC execution rate through six compute nodes

Conclusion

These tests show that DataWalk performance essentially scales linearly through six compute servers, for both data indexing, and for execution of the WCC graph algorithm.

These tests also reflect that DataWalk can effectively support large volumes of data. Using standard AWS hardware options with six compute servers, DataWalk can index 3.1B nodes + edges in as little as 22.5 minutes, and can execute the WCC graph algorithm across these 3.1B nodes + edges in as little as 2.8 minutes.