Demonstrating Near Real-Time Analytics with IBM DB2 Analytics Accelerator

Daniel Martin, Iliyana Ivanova, Raphael Mueller, Luis Eduardo Velez Montoya, Klaus Maruschka

IBM DB2 Analytics Accelerator Development
IBM Deutschland Research & Development GmbH
Schoenaicherstrasse 220
71032 Boeblingen, Germany
danmartin@de.ibm.com, iivanova@de.ibm.com, raphaelm@de.ibm.com, velez@de.ibm.com, marusch@de.ibm.com

Abstract: Version 3 of the IBM DB2 Analytics Accelerator (IDAA) takes a major step towards the vision of a universal relational DBMS that transparently processes both, OLTP and analytical-type queries in a single system. Based on heuristics in DB2 for z/OS, the DB2 optimizer decides if a query should be executed by "mainline" DB2 or if it is beneficial to forward it to the attached IBM DB2 Analytics Optimizer that operates on copies of the DB2 tables. The new “incremental update” functionality keeps these copy tables in sync by employing replication technology that monitors the DB2 transaction log and asynchronously applies the changes in micro-batches to IDAA. This enables near real-time analytics over online data, effectively marrying traditionally separated OLTP and data warehouse environments. With IDAA, reports can access data that is constantly refreshed in contrast to traditional warehouses that are updated on a daily or even weekly basis. Without any changes to the applications and without the need to introduce cross-system ETL flows, an existing OLTP environment can be used for reporting purposes as well. In this demo, we present a near real-time reporting application modeled on an industry benchmark (TPC-DS), but with a constantly changing set of tables with over 800 million rows that is running on DB2 for z/OS. In a browser-based user interface, demo attendants can influence the rate of changes to the tables and observe how the reporting queries are capturing new data as it is being modified by a separately running OLTP workload generator.

1 Introduction

IBM DB2 Analytics Accelerator (IDAA) version 3 is an evolution of IBM Smart Analytics Optimizer (ISAO) version 1[1SBS+11] which was using the BLINK in-memory query engine[RSQ+08, BBC+12]. Since version 2, the query engine and hardware used as the
basis for IDAA is Netezza 1000 (also known as "TwinFin")\(^3\). Figure 1 gives an overview of the involved components: IDAA is an appliance add-on to DB2 for z/OS running on an IBM zEnterprise 196 or EC12 mainframe, it comes in form of a purposely-built software and hardware combination that is attached to the mainframe via redundant 10GB fiber-channel connections to allow DB2 for z/OS to transparently offload scan-intensive queries. The acceleration factor compared to DB2 for z/OS standalone for such kind of queries can be up to 2000, because IDAA processes table scans on all of its disks in parallel, leveraging FPGAs to apply decompression, projection and restriction operations. Pushing these operations down to the level of the disk dramatically decreases the overall size of data required to be shipped over the internal network fabric and between the main parallel processing units (called SPU in Figure 1) that process joins, aggregations or complex expressions.

It is very important to note that nothing has to be changed on existing applications using DB2 for z/OS in order to get these benefits: they are connecting to DB2 for z/OS and are using the DB2 SQL syntax. In fact, there is no indication to an application if a query was processed by "mainline" DB2 or by IDAA. Based on how many fetch operations are estimated for a given query and how well existing indices match the predicates of that query, DB2 decides whether to process the query by itself or if it should be offloaded to IDAA. Deep integration into existing DB2 components ensures that only minimal training is required to operate IDAA with DB2 for z/OS as compared to DB2 for z/OS standalone. DB2 for z/OS is the owner of the data; data maintenance, backup and monitoring procedures do not change. Because IDAA operates on copies of the tables in DB2 for z/OS, whenever

\(^{3}\text{http://www-01.ibm.com/software/data/netezza/1000/}\)
there are changes to these tables, the changes need also to be reflected in IDAA. Currently, this is done by running DB2 UNLOAD utilities to either refresh an entire table in IDAA or by refreshing the changed partitions of that table.

The vision of IDAA is to allow reporting over "online" data, providing a solution that combines the features of traditionally separated OLTP and reporting systems that are connected by ETL jobs. For such scenarios, the granularity of the aforementioned refresh mechanisms is too coarse: if the total size of the changes is very low but the changes itself are spread over multiple partitions (or the tables itself are not partitioned), re-loading the majority of partitions or an entire table causes a lot of unnecessary work. Also, the practical minimum latency that can be achieved by running the refresh procedures is in the range of hours, it is impractical to run the UNLOAD-based refresh procedures every hour or even less than that. IDAA version 3 introduces a feature called "Incremental Update" that offers a mechanism to refresh the copy tables in IDAA by asynchronously monitoring the DB2 transaction log for changes. The changes are staged in memory and are transferred over the network to IDAA once their associated unit of work has been committed. For efficiency reasons, IDAA applies the changes it receives in "micro batches" that typically contain about 60 seconds of data from committed transactions sent from the log capture process on DB2 for z/OS. Multiple changes to a row within a micro batch are consolidated to a single change during the 60 seconds collection phase.

2 Demo Scenario

The demo is running on an IBM zEnterprise 196 mainframe running DB2 for z/OS which is connected to IDAA on a Netezza 1000-3 hardware. The IDAA hardware being used is a quarter-rack system that internally uses 24 1TB disks, 3 IBM HS-22 blades with 2 Intel Westmere 2.4Ghz 4-core CPUs and 3 FPGA daughter-boards on the Intel-blades with 2 4-core Xilinx FPGAs each. Communications in this FPGA-x86 cluster are coordinated using two IBM 3650 M3 hosts with 2 Intel Nehalem 2.4 Ghz CPUs each. The demo workload is based on TPC-DS\(^4\) with the fact table (STORE SALES) at a size of about 800,000,000 rows. Figure 2 depicts the demo scenario: a web interface allows users to run TPC-DS-based reporting queries and view the results as the queries are being processed. A second option allows users to start an OLTP workload generator that modifies the data by running high-frequent insert transactions on the tables. While the OLTP workload is running, users can again submit the reporting queries and observe that the reports are constantly changing as they are picking up the new data from the workload generator. The demo also provides status information about the total number changes that have been captured, the total number of changes that have been applied and an estimate of the current observable latency of the report, i.e. how far back in seconds is the current snapshot of the data on IDAA compared the state of the tables on DB2.

We will also provide a second demo running on this system that focuses on the capability of the DB2 for z/OS optimizer heuristic to distinguish between an analytical query that would

\(^4\)http://www.tpc.org/tpcds/default.asp
benefit when being offloaded to IDAA and transactional queries that run best on DB2 for z/OS. The workload generator in this demo simulates a real world business analytics workload with a mix of transactional and complex analytical queries (coming e.g. from IBM Cognos BI\(^5\)) that are triggered by a number of concurrent users. From the moment IDAA is enabled on the system, demo attendants will notice a drastic decrease in response time for the complex analytical queries and a major increase in throughput of the overall system.

References


\(^5\)http://www-01.ibm.com/software/analytics/cognos/