CLABBY ANALYTICS

Advisory

The ETL Problem Solved: The Compelling Financial Case for Running Analytics on the Mainframe

Executive Summary

With the introduction of analytics products tuned for the mainframe, and with improvements that have been made to z System architecture (manifest with the introduction of IBM's z13/z13s), IBM z Systems have become formidable transaction and analytics servers. It is now time to make mainframe architecture the enterprise centerpiece for analytics/Big Data processing.

Given that z Systems can now efficiently process analytics workloads side-by-side with transactions, the well-established practice of extracting, transforming and loading (ETLing) data to other servers needs to be abandoned. This practice is time consuming, inefficient, expensive and fraught with risk. It makes far more sense to process data on servers that already own that data.

Enterprise information technology (IT) executives need to formulate a cross-platform Big Data processing strategy that eliminates data movement. Stopping the ETL practice is the first step. And adopting a strategy that allows data from multiple sources to be analyzed without moving that data should be the second step. The evolving open source Apache Spark standard federates data from multiple sources.

Background

For decades, mainframes have been identified with high-speed, high-volume, secure transaction processing. To process analytics workloads, enterprises have traditionally extracted, transformed and loaded their data to other types of servers to process data-intensive workloads.

Over the past nine years, however, IBM has strengthened its z Systems analytics portfolio – as well as the z System platform. In addition to running the standard IBM transactional portfolio, several highly-tuned products have turned z Systems into formidable analytics processors:

- In 2011, IBM introduced a tightly coupled analytics appliance known as the IBM DB2 Application Accelerator (IDAA) that could perform complex analytics on mainframe data at high speed. This appliance has been extremely well received, particularly in financial and banking market segments, and new, more powerful versions have been introduced every year since.
- In early 2015, IBM announced a completely redesigned mainframe – the z13 – with much more memory, faster processing and additional systems software to accelerate analytics processing. A smaller version of this architecture, the z13s, was delivered a year later.
- In mid-2015, IBM partnered with Zementis to enable complex predictive analytics algorithms to be executed within the scope of an in-flight transaction without putting SLAs at risk.

With a broad/deep software portfolio, with the ability to accelerate complex analytics processing and perform predictive analytics within the scope of a transaction, and with improvements made to z Systems, mainframes can outperform traditional distributed systems and data warehouse/business intelligence clusters in terms of time-to-solution. For this reason, enterprise IT executives need to consider running analytics on z Systems.

In addition to building a broad/deep analytics portfolio and making important mainframe server improvements, IBM, in mid-2015, announced a major commitment to Apache Spark architecture (investing $300 million in coders, technology and education to promote Spark). Open source Apache Spark is a real-time data analysis environment that can process data 100x faster than Hadoop MapReduce in memory (or 10x faster on disk); it is highly flexible from a programming perspective (applications can be written in Java, Scala, Python and R); it enables structured and unstructured data to be combined (libraries include SQL, DataFrames and MLib for machine learning); and Spark can run on Hadoop, Mesos or standalone – and can access diverse sources of data including HDFS, Cassandra, HBase and S3.

Most recently, in March 2016 IBM made available the z/OS Platform for Apache Spark - an Apache Spark distribution of the open source, in-memory processing engine designed for big data combined with the industry's only mainframe-resident Spark data abstraction solution, providing universal, optimized data access to a broad set of structured and unstructured data sources through Spark APIs.

From a mainframe perspective, perhaps the most important thing about Spark is that it does not rely on a specific underlying data store (unlike Hadoop, which can only operate on data stored in an HDFS). This means that, with IBM-provided data access optimization technology, the list of Spark-enabled data sources is extended to the likes of DB2, IMS, VSAM, Adabas, and many other native mainframe file types.

Apache Spark enables mainframe data to be more easily analyzed in-place – as well as federate this analysis with data from other sources (such as social network environments) such that that data can be analyzed at the mainframe level. Spark makes it easier for data scientists to blend and analyze data from a variety of sources without having to move any of that data. Given the virtual elimination of the latency, cost and risk involved with copying data across platforms, enterprises need to consider building their Big Data processing strategies on Spark.

How the Mainframe Became a Powerful Analytics Processing Environment

Circa 2010, IBM started to focus on improving mainframe data-intensive computing functions. With the acquisition of Netezza (a maker of a complex query appliance), IBM developed a tightly coupled version of this appliance designed to make it possible to rapidly process complex queries. This mainframe-tuned appliance is marketed as IBM's DB2 Analytics Accelerator or IDAA.

What is important to note about IDAA is that it transparently and seamlessly uses mainframe data to deliver complex query results expeditiously. With IDAA, data can now be managed, controlled and secured within the mainframe environment – without having to ETL that data to other servers.

The next big step involved tuning and packaging analytics offerings on the mainframe (the z12 at that time). IDAA and the new mainframe architecture provided proof that mainframes could be used to process data analytics workloads. But further work was needed expand the amount of data that could be placed into memory in order to improve mainframe analytics processing performance.

In January 2015, IBM introduced the z13 – a mainframe environment initially focused on processing in-transaction analytics, but that now can process a broad range of analytics workloads. With more than three times more memory than the previous generation EC12, with significantly more cache, with greater parallelism and with a faster I/O subsystem – the z13 was now better suited to process a wide variety of analytics workloads on the mainframe.

> ***With the introduction of the z13, IBM mainframes became better suited than ever before to process analytics workloads – eliminating the need to purchase additional servers/software and the need to ETL data to those platforms.***

A Closer Look at the z13 and IDAA

As we indicated in this report, we observed five changes that made the new generation (z13) mainframe an outstanding analytics processor. These included:

1. Dynamic multi-threading;
2. Large memory pools;
3. Accelerated analytics processing;
4. Extended computational performance; and,
5. Data compression acceleration.

These enhancements – and the benefits that they deliver – are illustrated in Figure 1.

Figure 1 – Analytics Architectural Improvements

Large Memory Pools	Dynamic Multi-threading	Accelerated Analytics Processing	Data Compression Acceleration
Access and analyze large datasets in real time instantly	Boost performance for Linux, Java, and zIIP workloads	Optimization of complex, numerically-intensive analytics queries	Capture new opportunities due to lower cost of keeping data online
Up to 10TB of data to deliver up to 50% reduction in response time	24 to 1 consolidation ratio from x86 to zNext for up to 70% lower TCA	Significant throughput and response time improvement for analytics workloads	Reduce storage cost for sequential data by up to 75%

Source: IBM Corporation – January, 2015

Changes at the Processor Level

For decades, the z processor has been a single threaded, stacking CPU – a design that is particularly effective and efficient when it comes to processing transactions. But now, with processor improvements such as the ability to handle more threads and with single instruction, and with multiple data set (SIMD) improvements, this processor is now better suited to handle compute-intensive workloads. With new dynamic multi-threading capabilities, the z processor can now execute multiple threads simultaneously (as opposed to its single thread orientation in previous models) – enabling the new z Systems to boost performance when handling Linux, Java and zIIP (z Integrated Information Processor – a specialized Java processing environment) workloads. And, because of this threads/performance boost, the new z Systems are better positioned to handle x86 server consolidation (with a 24 to 1 consolidation ratio).

The ability to exploit SIMD (single instruction, multiple data set) vector processing is also important in the new z Systems architecture. SIMD enables z Systems to claim best of breed single thread performance, and best cache to thread ration in the industry.

Changes in System Design – Large Memory Pools

To handle multiple queries, while also processing transactions and other workloads, z Systems needed system design improvements – particularly in the amount of memory supported, the amount of cache available, and in the speed of the input/output (I/O) subsystem that feeds data to memory from storage. z Systems got a huge memory boost when the z12 was introduced four years ago (with up to 3TB of main memory). *The new z Systems, the has more than tripled the amount of memory (to 10TB) – and offers greatly increased cache, and support for even faster I/O speeds*. Further, the speed at which data can be delivered has also increased.

z Systems can now access and analyze large data sets in real-time. With improvements in available memory, cache and I/O speed the new z Systems can legitimately be called a unified, integrated transaction/analytics processing environment.

Accelerated Analytics Processing

In 2012 we wrote a report that described some of the progress that we were seeing in z System design as it pertained to analytics processing. In that report we stated that "IBM introduced IEEE binary floating point facilities at the end of the 1990s. The early 2000s brought 64-bit computing and superscalar parallelism to the mainframe (superscalar architecture implements a form of parallelism called "instruction level parallelism" ─ allowing a single processor to process work at a rate faster than its clock rate). Also, clock speeds have continually increased. Further, we noted that IBM had added "out-of-order execution," and has substantially improved floating point performance (mainframe floating point now rivals reduced instruction set processors). And, with the introduction of the EC12, IBM added significantly more on-chip cache. All of these improvements contribute strongly to positioning the IBM mainframe as an excellent processor for compute-intensive [numerically-intensive] SIMD vector processing."

The new z13 builds on all of this activity by optimizing the processing of complex, numerically intensive analytics queries – greatly improving throughput and response times for analytics queries.

Data Compression/Acceleration

The z13 also feature strong data compression/acceleration enhancements. With compression features, storage costs for sequential data can be reduced by up to 75%. As for acceleration, IBM's IDAA accelerator acts as an extension of DB2 for z/OS. It features the use of field programmable gate arrays (FPGAs) that speeds communications between the mainframe and the accelerator, as well as speeds the processing of data. As for the data, it is snapshotted to the accelerator and then constantly refreshed such that the data that is being analyzed is constantly kept current.

With All of These Improvements – Why ETL?

There is a common belief in the IT marketplace that the cost to transfer data between disparate systems (extract it, transform it, and load it) is inconsequential. After all – the logic goes – it can't cost that much to send a bunch of bits over a network to other systems, can it? As we show in this report, however, there are tremendous costs involved in moving data off of one platform to another – including costs related to additional hardware and software, as well as expensive labor-related costs. Further, the ETL process introduces security risks as data-in-motion can be tapped – creating security challenges for enterprises that ETL their data.

In October, 2013, *Clabby Analytics* wrote a report entitled "The ETL Problem" in which we described the cost penalties for ETLing data. These penalties can be found in three areas:

1. The additional servers, storage and networking equipment needed to support file transfers and process data;
2. The labor hours involved in managing file transfers and associated data; and,
3. The costs pertaining to wasted system cycles (system misutilization).

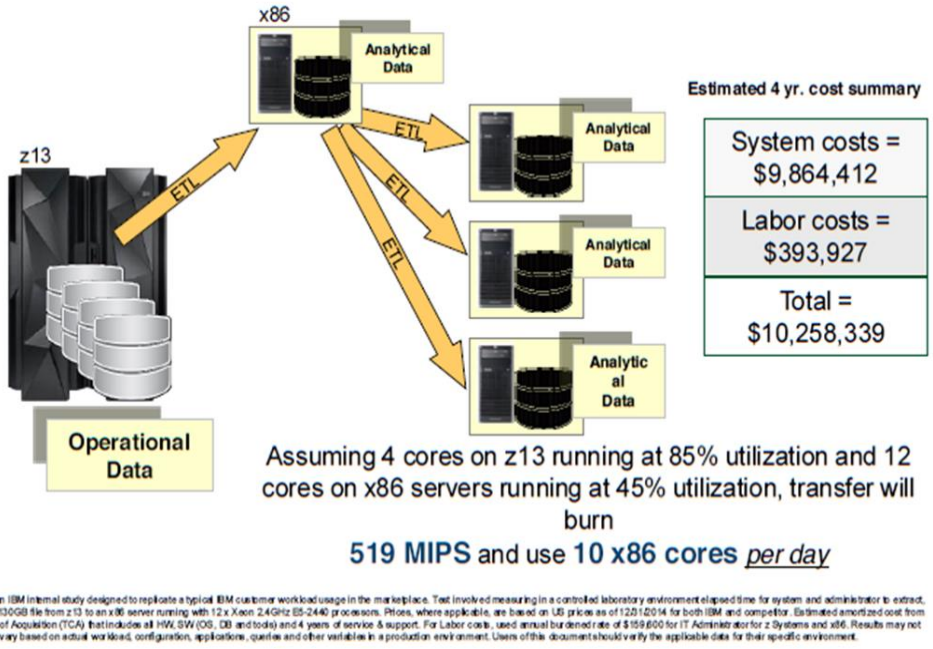
How much money are we talking about? Depending on the amount of data being transferred (and duplicated), the additional systems and labor costs can run into the millions of dollars – a huge expense just to move data from one place to another and manage it.

This ETL cost calculator can be used to estimate current ETL costs. For most enterprises, millions of dollars in equipment, software and management costs can be saved by ceasing the practice of ETLing data.

In our original ETL study we found that the cost of copying 1TB of data per day over a four year period was approximately $8 million. These costs included charges for additional hardware and software, as well as steep administrative costs for managing the movement and duplication of data. Updated data (shown in figures 2 and 3 – next page) now pegs the cost for moving the same amount of data at over $10 million. Wasted MIPS

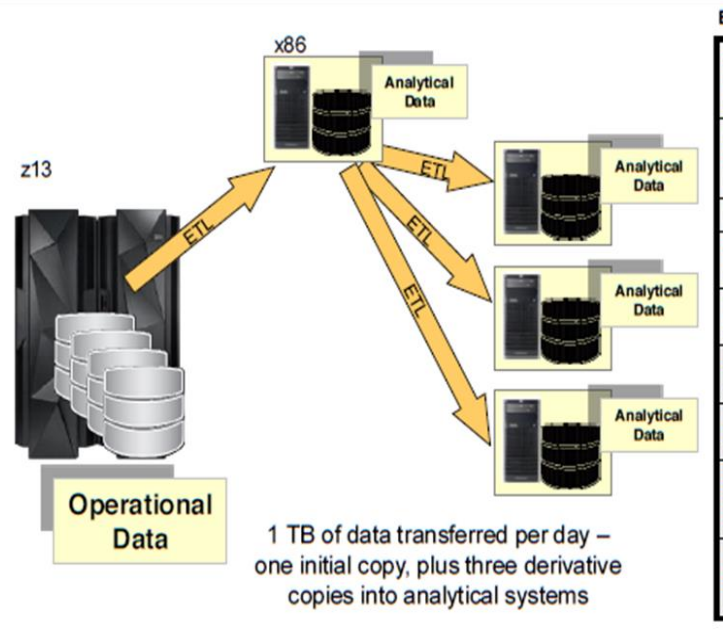
on the mainframe (an expense for moving the data), additional distributed systems and storage costs, additional software and increased labor costs (at almost $400,000) all contribute to the rising costs for ETL. And, we expect these costs to continue to rise in the future.

Figure 2: Costs Associated with ETLing Data to Other Platforms



This is based on an IBM internal study designed to replicate a typical IBM customer workload usage in the marketplace. Test involved measuring in a controlled laboratory environment elapsed time for system and administrator to extract, send and receive 130GB file from z13 to an x86 server running with 12 x Xeon 2.4GHz E5-2440 processors. Prices, where applicable, are based on US prices as of 12/31/2014 for both IBM and competitor. Estimated amortized cost from 4 Year Total Cost of Acquisition (TCA) that includes all HW, SW (OS, DB and tools) and 4 years of service & support. For Labor costs, used annual burdened rate of $159,600 for IT Administrator for z Systems and x86. Results may not be typical and will vary based on actual workload, configuration, applications, queries and other variables in a production environment. Users of this document should verify the applicable data for their specific environment.

Figure 3 – A Closer Look at ETL Systems, process and Administrative Costs



Estimated 4 yr. cost summary

System z Extract and Send	$1,593,056
Distributed Receive and Load	$7,924,021
Network	$58,285
System z Storage	$49,330
Distributed Storage	$238,720
System z Admin	$22,207
Distributed Admin	$143,080
System z Storage Admin	$20,800
Distributed Storage Admin	$207,840

This is based on an IBM sponsored and internal study designed to replicate a typical IBM customer workload usage in the marketplace. Test involved measuring in a controlled laboratory environment elapsed time for system and administrator to extract, send and receive 130GB file from z13 to an x86 server running with 12 x Xeon 2.4GHz E5-2440 processors. Prices, where applicable, are based on published US list prices as of 12/31/2014 for both IBM and competitor. Estimated amortized cost from 4 Year Total Cost of Acquisition (TCA) that includes all HW, SW (OS, DB and tools) and 4 years of service & support. For Labor costs, used annual burdened rate of $159,600 for IT Administrator for z Systems and x86. Results may not be typical and will vary based on actual workload, configuration, applications, queries and other variables in a production environment. Users of this document should verify the applicable data for their specific environment.

In addition to cost savings, enterprises that don't ETL their data can experience better control of data (because process breakdowns when moving data can lead to data becoming corrupted– leading to "multiple versions of the truth"). And, more importantly, by not moving data to other systems, enterprises can get analytics results more quickly, improving Quality-of-Service by speeding the time it takes to obtain query results.

Advice: Analyze Data Where It Lives

In our first ETL report, we suggested that the first question an IT executive should ask when it comes to analyzing data is: "where's the data located?" The reason we believe that this is so important is that moving data from one platform to another is both costly (see figures 2 and 3) and risky.

> ***Figures 2 and 3 show how expensive it is to move data from mainframes to distributed systems. But what if the data is owned by distributed POWER Systems? Or, what if the data is behind distributed x86 servers? Our same advice applies in each case: avoid ETLing that data if the systems that own the data are capable of running analytics on that data.***

We continue to hold to this position, and firmly believe that enterprises that follow this guidance will be able to "workload optimize" their computing environments – driving the cost to compute down to its absolute minimum while improving service levels.

Another Efficient Approach to Analyzing Data Across Platforms: Apache Spark

By some estimates, mainframes own 70% of enterprise mission critical data. As a consequence, this *Advisory* has focused on why it makes sense to process mainframe-owned data at its source – on IBM z Systems. But what about data that is not under direct mainframe control. How can that data be blended with z data and analyzed efficiently?

As stated previously, IBM has made a heavy financial commitment to Apache Spark. *This open source environment represents a means to federate analysis across multiple types of platforms, including mainframes, RISC servers and x86 environments.* Spark offers a unified programming model with rich libraries, allows for machine learning, and can work with many language types (including Java, Python and others). Using Spark, open source virtual libraries can trade transaction information, can access data environments that are not SQL, can combine enterprise data with social network data – and much more. Spark also preserves data science syntax, which makes programming simpler for data scientists.

> ***Spark on z Systems is available on Linux via the Apache Spark open source distribution. On z/OS, the z/OS Platform for Apache Spark product bundles this Spark support with optimized data access services that provide rapid access to data regardless of location, format or interface – without having to ETL data – enabling the analysis of multiple data types and speeding the time-to-solution.***

Summary Observations

In 2010, IBM started making great strides in changing the very nature of its mainframe architecture. Until 2010, mainframes had largely been used as powerful, high-volume transaction processing systems. But, with the availability of IDAA, the mainframe entered

a new era in computing – an era where mainframe data could be efficiently analyzed without leaving the control of the mainframe. IBM's IDAA architecture is tightly coupled with the mainframe – enabling the efficient processing of complex analytics workloads.

As IBM continued to tune the mainframe, the next generation mainframe (the z13) was being developed – a server environment that offered faster processors, that handled more threads than the previous generation, that offered access to three times as much memory – and that contained data handling improvements in related systems software. With the introduction of the z13, IBM turned the corner – making z Systems not only the market's most powerful transaction processor, but also a powerful analytics engine.

IBM's partnership with Zementis enabled predictive models to be executed directly within the scope of an in-flight transaction, opening up new opportunities to both reduce bottom-line expense (e.g. better control fraud, waste, abuse and financial crimes) as well as grow top-line revenue with up-sell/cross-sell suggestions calculated on each and every transaction.

As IBM has embraced open source Apache Spark, Spark has been enabled on the mainframe along with new optimized data access technologies that unlock the ability for this data to be analyzed in-place, greatly simplifying data preparation and data programming for data scientists. And since not all data resides at the mainframe level, Spark's ability to federate analysis across platforms levels the data playing field when it comes to moving data – Spark eliminates the need to extract, transform and load data on various platforms.

In days gone by, the mainframe was essentially quarantined, locked into a position as a transaction processing engine. But with very significant improvements to mainframe hardware, with improvements to mainframe systems software, and with additional support coming from the independent software vendor (ISV) community – the mainframe has now become a powerful transaction and analytics serving environment that can deliver results faster than waiting for external servers. Accordingly, the need to move data away from the mainframe to other servers has dissipated. The ETL problem has now been solved.

IT executives looking for a more efficient, cost effective way to process their growing analytics and Big Data workloads now need to look no further than their venerable, secure, efficient mainframe environment.

Clabby Analytics
http://www.clabbyanalytics.com
Telephone: 001 (207) 239-1211