The End of the Nightly Batch: How In-memory Computing and the Cloud Are Transforming Business Processes

Antoine Chambille[a] and Gaëlle Guimezanes[b]
Department of Research and Development, ActiveViam SAS, 46 rue de l'Arbre Sec, Paris, France

Keywords: Cloud, In-memory Computing.

Abstract: This paper illustrates our conviction that data processing relying on nightly batches is a thing of the past. It shows how a complex analytics platform for huge amounts of data can be created from scratch in a matter of minutes on the cloud, thus negating the need for dedicated on-premise machines that would do the job slower at a higher cost. We also present the CloudDBAppliance project, where we aim at building a Cloud database platform that will be available as a service, and will integrate the operational database with the tools for further data analysis, eliminating the hassle of exporting the data for processing.

1 INTRODUCTION

This is perhaps the closest thing to a tradition in the still-young field of IT. For decades, organizations have used nightly batches to integrate and compute all the changes registered during the day and deliver to business users their data in the morning.

It was the best and often only way to move data from operation systems onto analytical platforms for decision-making without affecting live users. It was also a practical way to complete complex calculations on fixed-size hardware.

However, this is not true anymore. With both in-memory computing and the Cloud having achieved full technological maturity, calculations that were previously too costly and had to be broken down into manageable data chunks can now be conducted in true real-time by in-memory solutions, while the Cloud model provides the flexibility to deal with business peaks, ensures a smooth user experience throughout and erases the need to write off costly hardware investments.

In this paper, we will benchmark ActivePivot (ActiveViam, 2019), a reference in-memory analytics database from ActiveViam, on the Cloud with some of the most resource-intensive, real-world business use cases today and see how this technology combination can enable organizations to drive their business in real-time, working with data that is either continuously updated or loaded and aggregated on-the-fly, and what benefits it brings to the business. Ultimately, we will see that nightly batches have become both unnecessary from a technical standpoint and not competitive cost-wise.

2 TESTING A FINANCE USE-CASE ON A CLUSTER ON THE CLOUD

We simulated a use-case from the finance industry based on ActiveViam's experience working with the largest banks in the world. Finance is one of the most demanding verticals for analytical capabilities. We loaded millions of transactions, prices and sensitivities recorded for hundreds of historical dates, amounting to 50 TB of data in memory across a 1,600-core cluster on Google Cloud. We wanted to see if a combination of Google Cloud hardware and services and ActivePivot in-memory technology could achieve high-enough performances to perform live analytics on such a volume of data during the course of a regular work day, without relegating any task to batch processing. We tested the scenario from start to finish, including setup, loading, queries and calculations.

[a] https://orcid.org/0000-0002-1344-4802
[b] https://orcid.org/ 0000-0002-6933-9171

Chambille, A. and Guimezanes, G.
The End of the Nightly Batch: How In-memory Computing and the Cloud Are Transforming Business Processes.
DOI: 10.5220/0007905606540659
In *Proceedings of the 9th International Conference on Cloud Computing and Services Science (CLOSER 2019)*, pages 654-659
ISBN: 978-989-758-365-0
Copyright © 2019 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

2.1 Setup and Loading

The exceptional storage and networking infrastructure used by public cloud providers make it possible to decouple storage from compute. They are fast enough to load large datasets in minutes and operate in-memory analytical platforms on demand.

At least that is the theory, because not every software platform can be operated on-demand. In-memory databases from Oracle, Microsoft and SAP for instance are now available in the Cloud, but still follow a monolithic architecture where storage is attached to and managed by the database, and that database is always on. To take full advantage of the cloud model and effectively eliminate the need for batch processing, another approach is needed that decouples storage from compute and allows customers to use memory resources on demand. In practice it means storing data in object storage and loading it on the fly at massive speeds in VMs, started on demand and disposed of when the analysis is done.

We leverage the Google Cloud Platform to do just that. We store the historical dataset in Google Cloud Storage, we start servers on the fly when a user needs them and shut them down when they are done. To mount the data in memory and deliver analytics in the hands of business users, we use the ActiveViam platform.

In the following section we explain in practice how we did it for a 50TB dataset (500 historical days, 100GiB per day). In short, we managed to start a 1600-cores, 64TB RAM cluster from scratch, load the 50TB in-memory and be ready to serve real-time queries, all in less than 20 minutes...

2.1.1 Using Scripts to Set up a Whole Infrastructure: In-memory Computing without the Hassle

It can be difficult to take full advantage of in-memory computing. Managing tens of server nodes dynamically, optimizing data loading from cloud storage, efficient memory allocation of billions of records in terabytes of RAM or parallel computing on hundreds of core and NUMA architectures are difficult tasks. Fortunately, in a public cloud model the cloud and the software providers take care of all issues related to configuration and optimization, and client organization only need to start the servers to access almost immediately a fully-functioning analytics platform.

In the case of Google Cloud and ActiveViam, you can start and stop 100 servers if you want with a short script calling cloud APIs or by describing resource templates in tools such as the Google Deployment Manager. All servers start with a given image, in our case a Linux operating system configured with memory intensive settings.

Example of an allocation loop from the script used in the benchmark:

```
for ((i=0; i < 100; i++)) do

   NODE_NAME="$INST_NAME-$NODE_TYPE-$i"

   changeLine "TC_NODE_NUM" "$i"
"$STARTUP_SCRIPT_FILE"

   echo "=== Creating new $NODE_TYPE
node [$NODE_NAME]..."

   # Create node
   gcloud compute instances create
$NODE_NAME \
      --custom-cpu 32 --custom-memory
624 \
      --custom-extensions \
      --min-cpu-platform skylake \
      --zone $ZONE \
      --metadata-from-file startup-
script=$STARTUP_SCRIPT_FILE \
      --tags http-server,https-server \
      --image-project ubuntu-os-cloud \
      --image-family ubuntu-1604-lts
$MACHINE_SPECS \
      --async –quiet

done
```

Note that in this scenario, we take advantage of "custom" instances in the Google Cloud Platform. It allows us to tune the "memory-to-core" ratio, and allocate servers optimized for in-memory computing. In this case we allocate 100 servers with 32 vCPUs (16 cores, Intel Xeon Skylake) and 624 GiB of RAM (at the time of the experiment the maximum with custom instances).

To take advantage of such a cluster (1600 cores, 64TB RAM!) you need the latest technological advances: In-memory column stores, multidimensional cubes, advanced memory management that breaks the limits of garbage collection on the Java platform, work stealing thread pools to maximize the usage of the cores, MVCC to support real-time incremental updates… You need all that and more to deliver high definition analytics on the fly. But you don't need to learn how to do those things. It's integrated in the ActiveViam platform.

2.1.2 Data Loading

The next challenge is data loading. The dataset must

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

be transferred from cloud storage to the cluster within minutes to satisfy the on-demand constraint, which means we must move 50TB in minutes. This is where you can really leverage the world class infrastructure of a large public cloud provider.

We have 500 days of historical data, so we load 5 days per node (500GiB). The objective is that each one of the 100 nodes downloads its data at maximum speed, and that they all do it at the same time.

At the time of this test the maximum bandwidth of the network interface in a GCP instance was 16 Gbit/s. So, an instance had the theoretical capacity to transfer data at 2GiB per second. In this case the data is actually downloaded from Google Cloud Storage and the throughput of a data transfer is far less than that, being HTTP based. ActiveViam developed a special connector that opens tens of HTTP connections to blob storage and performs the transfer in parallel, transparently reconstructing blocks. With this trick, a node running the ActiveViam connector can saturate the 16 Gbit/s cap and actually download 500GiB in about 5 minutes.

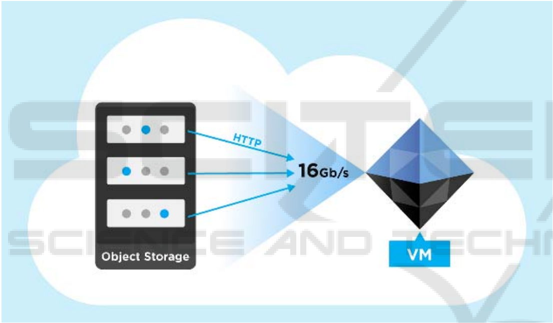

Figure 1: Opening several connections to the storage for each VM.

Of course, when 100 instances download their 500GiB concurrently and at full speed, Cloud storage becomes the next bottleneck. This is where cloud infrastructures are really put to the test, and during our benchmark we observed impressive scalability from Google Cloud Storage. The 50TB dataset was entirely transferred from storage to the 100 nodes cluster in less than 10mn, as shown on Figure 1. That’s a monumental 90GiB/s effective throughput between storage and compute, reached with possibly the simplest and least expensive storage solution in the cloud platform.

So, it takes a few minutes to start the GCP instances, 10 minutes to transfer the data, and another few minutes for the ActiveViam platform to get its in-memory data structures ready. All together the entire cluster is ready in about 20 minutes.

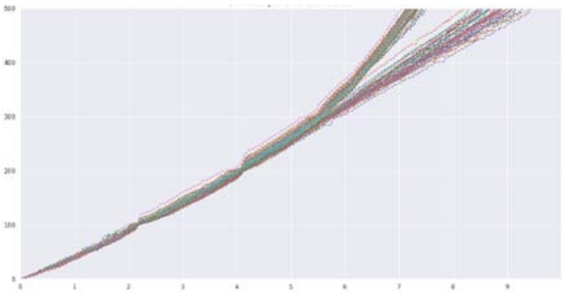

Figure 2: Downloading 50TB with 100 instances. Amount of data downloaded per instance in GiB (Y axis) when time passes (time in minutes on X axis).

It is fast enough to be started up at the beginning of the work day, or at any time at the request of business, without having to schedule a batch.

2.2 Benchmarking Queries for Interactive Analysis

Finally, we looked at the performance of interactive queries on this on-demand cluster. We performed two different queries and check if they follow Gustafson’s Law (Gustafson, 1988): queries on twice the dataset should take the same time if you also double the CPU power and RAM. Query 1 performs aggregations on all the records of the dataset, Query 2 performs aggregations on a filtered selection of records (about 10%).

We scaled up the cluster step by step, each time doubling its size, from 1 VM to 100 VMs. Below you can see that the scale-up is good: the query times are close to constant. In the end, we only observe a 10% difference between a query running on a single VM and a query running on a 100-VMs cluster. Query times, as shown on Figure 3, remain well under half-a-second, which is fast enough to provide end users with a comfortable, responsive experience for analysis.

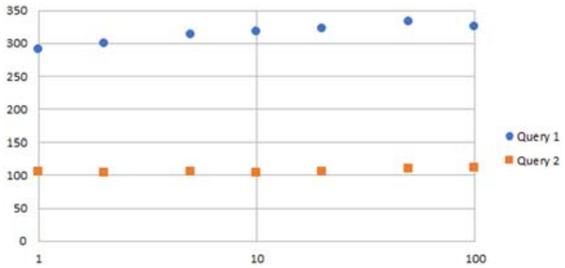

Figure 3: Scaling of ActiveViam queries. Time taken for a query in ms (Y axis) per number of nodes (X axis).

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

For this test, we deliberately simulated a use case that is as challenging as the most demanding real-world business, or even more. The lesson here is that if it can free itself from batch processing, there isn't really any business case in the real world that cannot.

3 ONE INSTANCE ONLY: AN ALTERNATIVE TO BATCH PROCESSING FOR SMALLER APPLICATIONS

The previous use-case that we just presented is extreme, but most of the business applications that currently use batch processing are in fact of a much smaller scale. Some of the use-cases that we encounter at ActiveViam are listed in Table 1. As we can see, those use cases don't need to be run on a huge cluster and can be run on a single machine, even if that machine needs to be one with large memory.

Table 1: Use-case and memory usage.

Use cases	Typical memory usage
Dynamic pricing for online retailers, real-time portfolio monitoring for one trading desk	128 GB
Global Finished Vehicle Logistics management, multichannel and geolocalized retail pricing	512 GB
Supply chain control tower for a global retailer, market risk Value at Risk and Expected Shortfall (FRTB regulation)	1 TB
Trade level xVA analytics for investment banks	4 TB

3.1 An Affordable Alternative on Public Clouds

Dedicated in-memory computing servers on premise have experienced a slow progression over the last 10 years. Such machines remain expensive and their specific configuration means they cannot be repurposed as easily as more conventional hardware. They represent a significant investment that, like all hardware purchases, will depreciate over time and eventually will have to be sold or written off.

On the other hand, in-memory computing is now completely mainstream and standardized on all the main public cloud platforms. Public clouds have turned a terabyte of memory into a commodity and the major cloud providers offer a similar range of compute instances optimized for in-memory computing, going from a few GB to several terabytes all available on a "pay-as-you-use" basis. Such memory sizes as those presented in Table 1 are readily available on public clouds nowadays.

This is true for conventional hardware configurations as well, but using the Cloud for in-memory applications is especially attractive from a budget standpoint considering the fairly high purchasing price of the specialized hardware needed for this usage.

Price has become standardized, and there is no premium on large memory servers like there used to be. Basically, a TB of memory is now a commodity that can be purchased for around $7 per hour from any major public cloud provider as shown in Table 2. Please note that these prices were snapshotted at a given time, for example sake only, since prices on public clouds evolve continuously.

Table 2: High-memory instances from major public Clouds.

	GCP	Azure	AWS
128 GB	n1-highmem-16 9.1$ /TB/hr	E16 v3 8.5$ /TB/hr	r5.4xlarge 7.8$ /TB/hr
512 GB	n1-highmem-64 9.11$ /TB/hr	E64 v3 9.3$ /TB/hr	r4.16xlarge 8.7$ /TB/hr
1 TB	n1-ultramem-40 6.5$ /TB/hr	M64s 7$ /TB/hr	x1.16xlarge 7$ /TB/hr
4 TB	n1-ultramem-160 6.8$ /TB/hr	M128ms 7$ /TB/hr	x1e.32xlarge 7$ /TB/hr

One dilemma when starting an in-memory computing project is how to choose the right server for your project, and there is a temptation to over-provision, especially if you anticipate growth. With the cloud, you use exactly what you need now. If the business grows, you can switch to a larger resource almost seamlessly - as long as the software is designed to cope.

We have shown that it is possible to load data from the Cloud storage and launch ActivePivot instances very quickly to answer to a business query that, in previous architectures, would have to wait for the next nightly batch. However, this design assumes that the business applications producing the data consumed by ActivePivot have actually pushed their data in real-time to the Cloud storage. In the next section we will discuss the CloudDBAppliance project, in which we aim to build a platform that will be able to contain both the operational database and the analytical one, both in-memory, allowing for the shortest delays between a change in the operational data and its availability for analytical queries.

3.2 CloudDBAppliance: Operational Database and Analytics, All in One

The CloudDBAppliance project is funded by European Union's Horizon 2020 research and innovation programme. It aims at producing a Cloud

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Database Appliance for providing a Database as a Service able to match the predictable performance, robustness and trustworthiness of on-premise architectures. This project will deliver a cloud database appliance featuring both the next-generation hardware for in-memory computing and the necessary software for processing high update workloads on an operational database, performing fast analytics and applying data mining / machine learning techniques on a data stream or on a data lake, the robustness being protected by redundancy and failover mechanisms.

We have tested the ActivePivot analytical database on the hardware built during this ongoing project. The platform we tested is a Bull Sequana S800 with 8 processors Intel(R) Xeon(R) Platinum 8158 CPU @ 3.00GHz, (which represents 96 cores), each constituting a NUMA node with 512 GB of RAM (4 TB in total). For our experiment, we have used the well-known TPC-H benchmark's dbgen tool (TPC, 2019) to generate data at two different scales: we used a factor of 100, then 500, to generate respectively 107 GB then 542 GB of data. Those data quantities are roughly representative of the two smaller use case sizes cited in Table 1, for which the platform would have enough memory to co-host the operational database and the analytical database. The data (as generated in CSV format) was stored on SSD disk on the machine. We evaluated the total startup time, from launching the command line that starts the analytical database until the analytical database is able to answer queries on the full dataset. This includes the file reading, parsing to transform the CSV into the actual data types, adding the data to the datastore, and publishing to the analytical cube.

On the 107 GB dataload, the total startup time was 6 minutes and 31s, which represents an average throughput of 2,209,997 records published per second on that initial load. On the factor 542 GB dataload, the total startup time was 29 minutes and 9 seconds, with an average throughput of 2,006,060 records published per second. The total startup time might seem a bit long considering everything is local to the machine with no need to download the data through network. However, the loading performance is hindered by the TPC-H setting in which all data resides in a single CSV file: even if the parsing and publishing of records can happen in parallel, the actual file reading still has to be done sequentially.

We are currently implementing a connector between the LeanXcale database, the operational database that was selected for the CloudDBAppliance project (CloudDBAppliance, 2018), and the ActivePivot analytical database. We expect throughputs to be much higher, since the data will be pushed directly from the in-memory database, so:

- The data will be read from RAM, not SSD disk;
- the data will already be in a binary format, no need for parsing the CSV into actual data types;
- the data will already be broken into individual records, removing the constraint of having to read one single CSV file sequentially.

We can envision several scenarii for the best usage of the platform's capacities. In one scenario, the operational database is always on, but the rest of the machine resources are shared between different kinds of applications: the ActivePivot analytical database can be started on-demand to compute complex analytical queries, and be shut down after usage to leave resources to other applications such as machine learning Spark jobs performed on the datalake. In that scenario, the main difference from using a datawarehouse built from nightly batches is that the data will be fresher, since the application can be loaded on-demand from the latest version available in the operational database. In another scenario, the ActivePivot is always on, receiving data updates continuously from the operational database. This way analytical queries can be performed at any time without any loading phase and reflect real-time changes in the operational data. In both scenarii, the whole machine is always up because the operational database has to be always available. The main advantage of being in the Cloud rather than using an on-premise machine resides in the ability to easily switch from one configuration to another when the needs evolve, instead of having to buy a new server. The LeanXcale operational database indeed comes with data migration capacities that allow to switch the hosting machine without interruption of service, and restarting an ActivePivot on the new machine has been shown to be a less than half an hour affair.

4 CONCLUSION

While it's not feasible to try out every use case, we believe we demonstrated through our tests that one of the most demanding business cases can be addressed fully without the need for a recurring nightly batch. If financial risk analytics can be performed without scheduled batches through a combination of cloud infrastructure and in-memory computing, we are convinced that there aren't many real-world business scenarios that cannot be addressed in the same way.

Furthermore, while it serves its purpose for decades, nightly batch processing now appears not only a second-rate option at best, but simply wasteful compared to the alternative. It is wasteful in time, in capital and in power. Fortunately, the transition to a better model is now easier than ever for organizations.

We are currently participating in a research project that aims at building a cloud platform on which an operational database is integrated with analytical capabilities, allowing for real-time data processing without hassle.

ACKNOWLEDGEMENTS

The CloudDBAppliance project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 732051.

REFERENCES

ActiveViam, 2019. *ActivePivot product page*. https://activeviam.com/en/products/activepivot-in-memory-analytical-database.

The CloudBDAppliance Consortium, 2018. *Deliverable D2.3: Final version of the In-Memory Many-Core Operational Database.*

Gustafson, J., 1988. Reevaluating Amdahl's Law. In *Communications of the ACM*. Volume 31 Issue 5.

TPC, 2019. *TPC-H Homepage*. http://www.tpc.org/tpch/.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS