End Point

22.9 Million Transactions Per Minute and Microsecond Latencies on a Single Commodity Server with Forsa

14 May 2019

Prepared for: Formulus Black

Prepared by: Josh Williams (jwilliams@endpoint.com)

Executive Summary

Formulus Black has developed a Linux-based software platform called Forsa that enables any application to run in-memory without modifications. Forsa does this by allowing applications to use memory (DRAM, NVDIMM, Intel® Optane™ Memory Mode, etc.) as storage instead of slower spinning disks or flash media. Forsa provides a KVM-based hypervisor and GUI + RESTful APIs for virtual compute and storage configuration and management and supports a "host mode" for Linux-based applications where virtualization is not required.

Formulus Black engaged End Point to benchmark the performance of Microsoft SQL Server running on a mid-range Intel server with Ubuntu Linux and Forsa installed. Test results include achieving over 22.9 million aggregate transactions per minute in "host mode" and over 5.1 million aggregate transactions per minute in a virtualized environment. Latency was an impressive 0.25ms, on average.

These results imply significant mixed read / write workload performance can be achieved even on a single mid-range Intel server by running applications in-memory with Forsa.

Configuration

Software

Ubuntu 18.04, Forsa 2.0, and Microsoft SQL Server 2019 were first installed on the test server. For SQL Server on Windows, MAXDOP was enabled and we followed the best practices as outlined at: https://blogs.msdn.microsoft.com/docast/2018/02/01/operating-system-best-practice-configurations-for-sql-server/

HammerDB was used as the benchmark tool. Version 3.1 runs the TPC-C benchmark used in this report: https://www.hammerdb.com/

HammerDB's parameters include both "Virtual User" count, the number of client threads created within the process to generate concurrent load, and "warehouses", a scaling factor for the size of the database generated. HammerDB's recommendation is to have no less than 250 warehouses per socket and this recommendation has been adhered to in all configurations.

System Used for the Benchmark

Dell PowerEdge R740xd with two Intel Silver 4114 CPUs, 768GB of memory (64GB x 12), two 800GB PCIe SSDs and a 200GB SSD Boot Drive. The SSD drives were not used and the databases were stored in memory on "LEMs" (Logical Extensions of Memory) created using Forsa (650GB out of 768GB was set aside for LEMs, leaving the system with 128GB of memory). Each Intel Silver 4114 CPU has 10 physical cores (20 total physical cores and 40 Hyper Threads).

Benchmark Configurations

Four different Microsoft SQL Server database configurations were tested:

1. Virtual Machine - Low memory
2. Virtual Machine - High memory
3. Bare Metal - Small Databases
4. Bare Metal - Large Database

Virtual Machine – Low Memory Details:

8vCPUs and 4GB of RAM were assigned to each virtual machine. The test varied the number of concurrent virtual machines, up to a total of four. Windows Server 2016 running Microsoft SQL Server was tested in parallel under load. Block storage was provisioned through Forsa by assigning LEMs to each virtual machine: 50GB for database and 20GB for transaction logs.

On the client side the “warehouses” value, a scaling factor within the HammerDB benchmark client, was set to 250 warehouses to mostly fill the data volume. The load was generated from virtual machines running on a secondary server: 8 client virtual machines each with 8vCPUs, 4GB of RAM, and 50 client threads (ie: 400 concurrent users) were set up to drive load to a total of four virtual machines running Microsoft SQL Server over a 1Gbps Ethernet LAN.

Virtual Machine – High Memory Details:

32vCPUs and 16GB of RAM were assigned to a single virtual machine , similarly running SQL Server 2019 on Windows Server 2016. Block storage was provisioned through Forsa by assigning LEMs to the virtual machine: 350GB for database and 100GB for transaction logs.

On the client side the warehouses value was set to 2000, again to mostly fill the data volume’s available space. The load was generated from virtual machines running on a secondary server: 4 client virtual machines each with 8vCPUs, 4GB of RAM, and 100 client threads (ie: 400 concurrent users) were set up to drive load to the virtual machine running Microsoft SQL Server over a 1Gbps Ethernet LAN.

Bare Metal – Small Databases Details:

Rather than run inside a VM, SQL Server 2019 was installed directly within the Ubuntu 18.04 host OS. The host’s 20 CPU cores (40 CPU threads) were all available, and SQL Server was limited to 16GB of RAM. Block storage LEMs were provisioned by Forsa: 450GB for databases and 100GB for transaction logs.

The load was generated from virtual machines running on a secondary server: 8 client virtual machines each with 8vCPUs, 4GB of RAM, and 120 client threads (960 concurrent users) were set up to drive load concurrently to each of the 8 Microsoft SQL Server databases over a 1Gbps Ethernet LAN. The test varied the number of databases and client machines used, starting from

1 and ending at running on all 8 simultaneously. Each of the 8 databases was separately loaded with using 500 warehouses as the scaling factor.

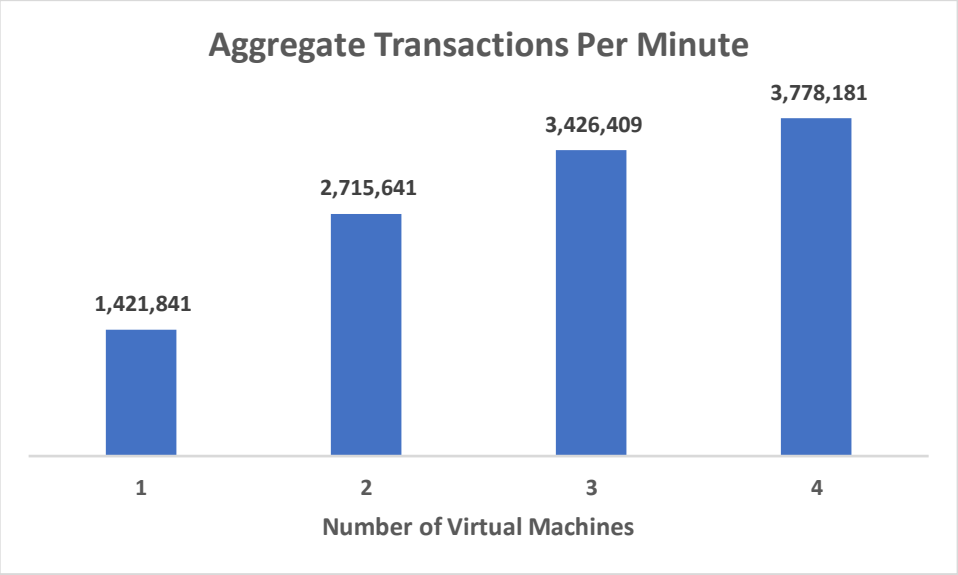
Bare Metal – Large Database Details:
The host's 20 CPU cores (40 CPU threads) were all available, and SQL Server was limited to 32GB of RAM. Block storage LEMs were provisioned by Forsa: 450GB for database and 100GB for transaction logs.

The HammerDB warehouses scaing factor was set to 4,000. The load was generated from one virtual machine running on a secondary server: 8vCPUs, 4GB of RAM, and 120 client threads (120 concurrent users) were set up to drive load to the large Microsoft SQL Server database over a 1Gbps Ethernet LAN.

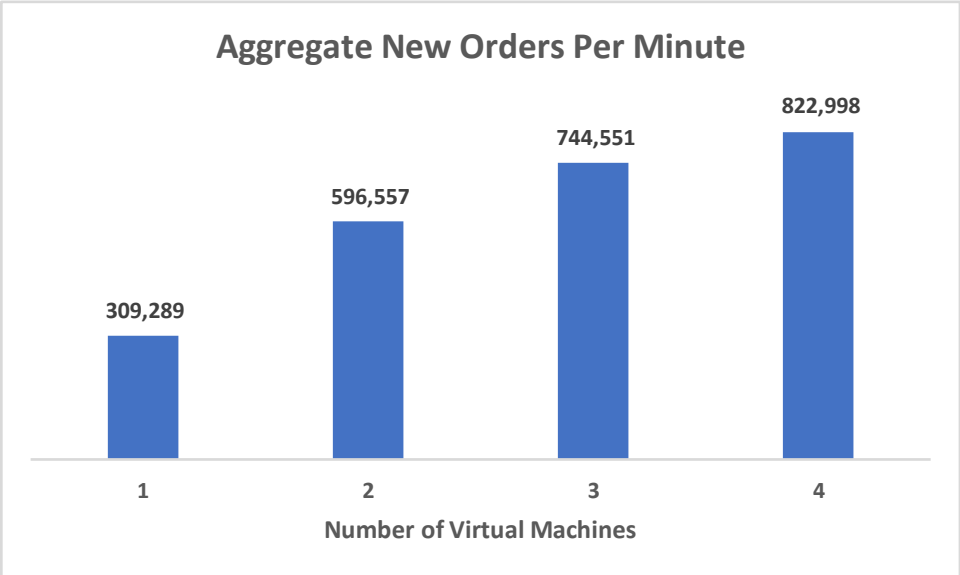
Results

Virtual Machine Low Memory Configuration-

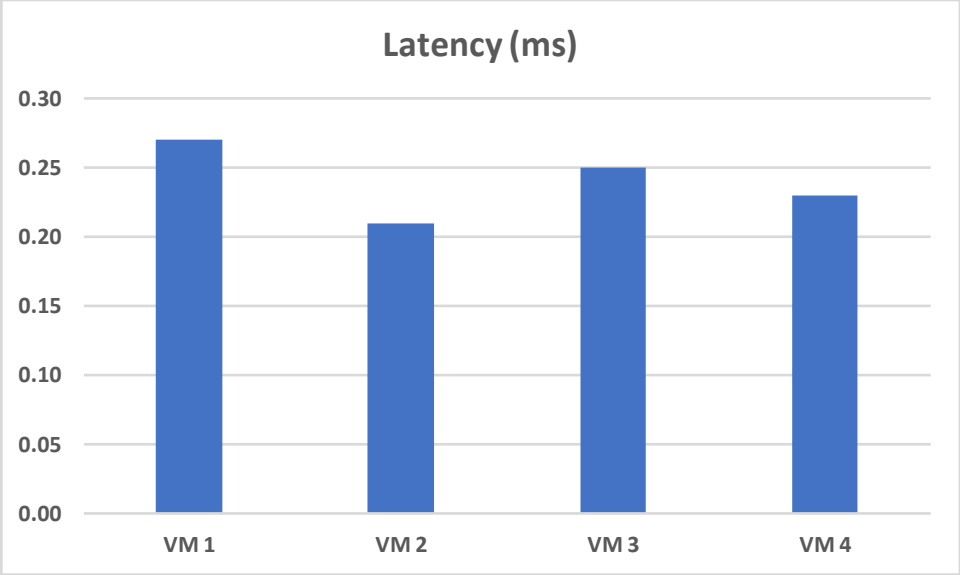
Transactions Per Minute



New Orders Per Minute

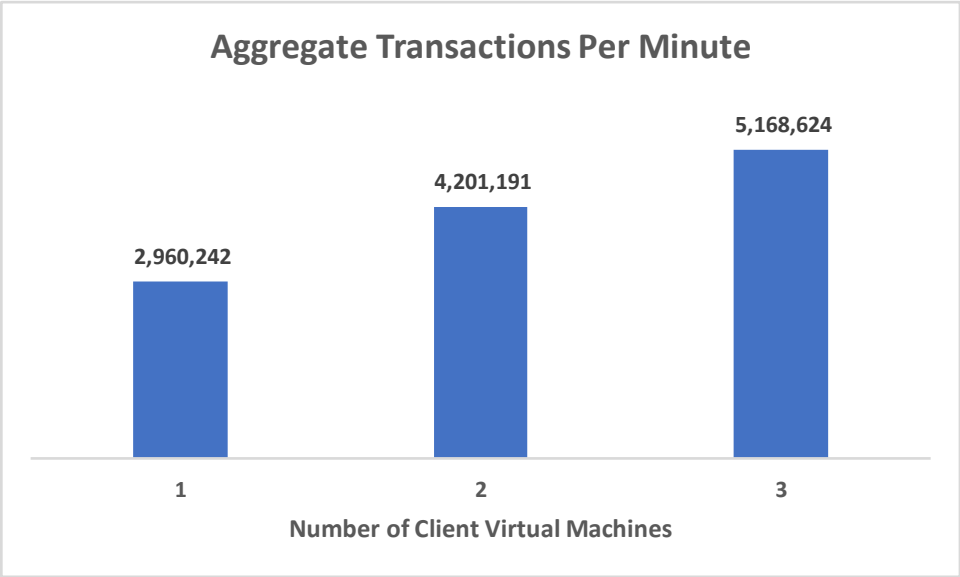


Virtual Machine Low Memory Configuration - Latency

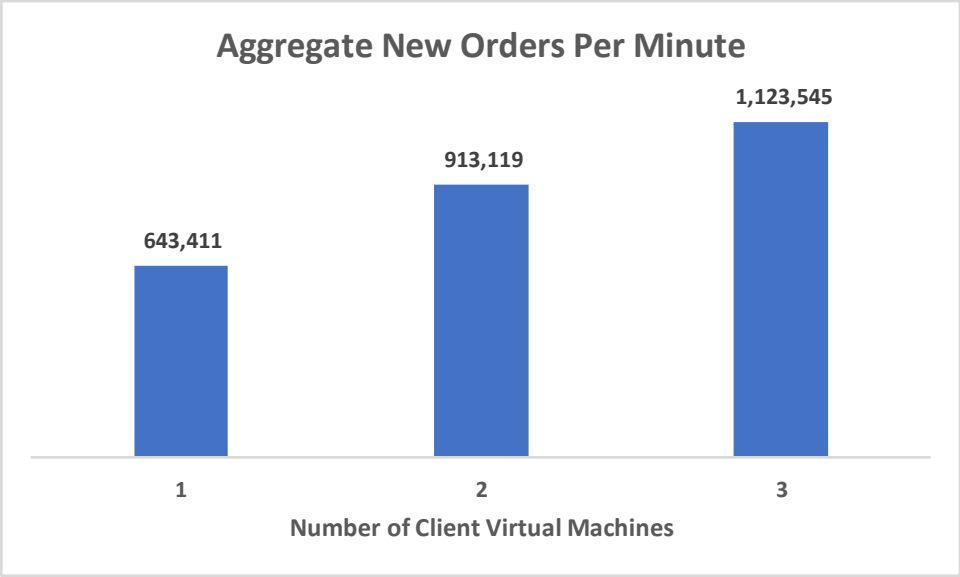


Virtual Machine High Memory Configuration

Transactions Per Minute



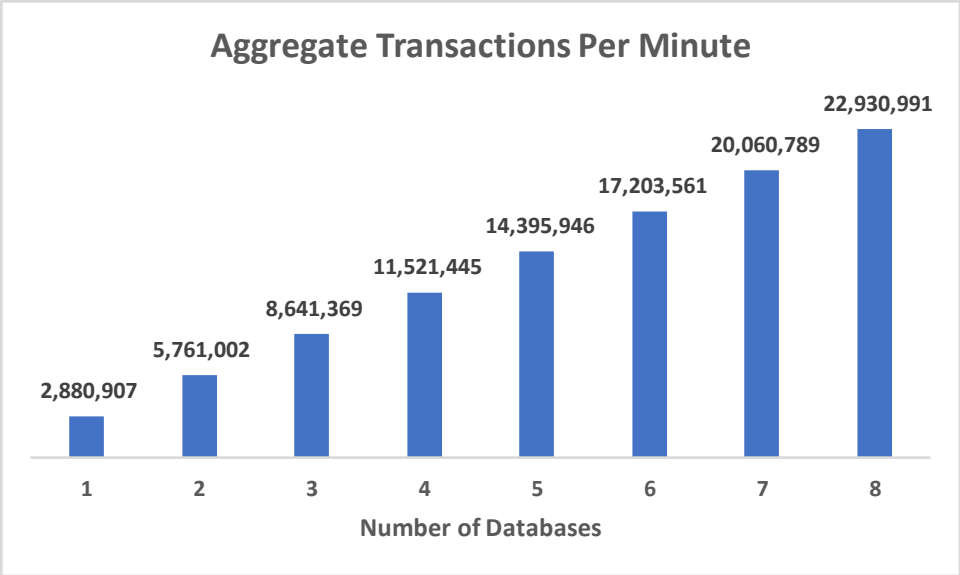
New Orders Per Minute



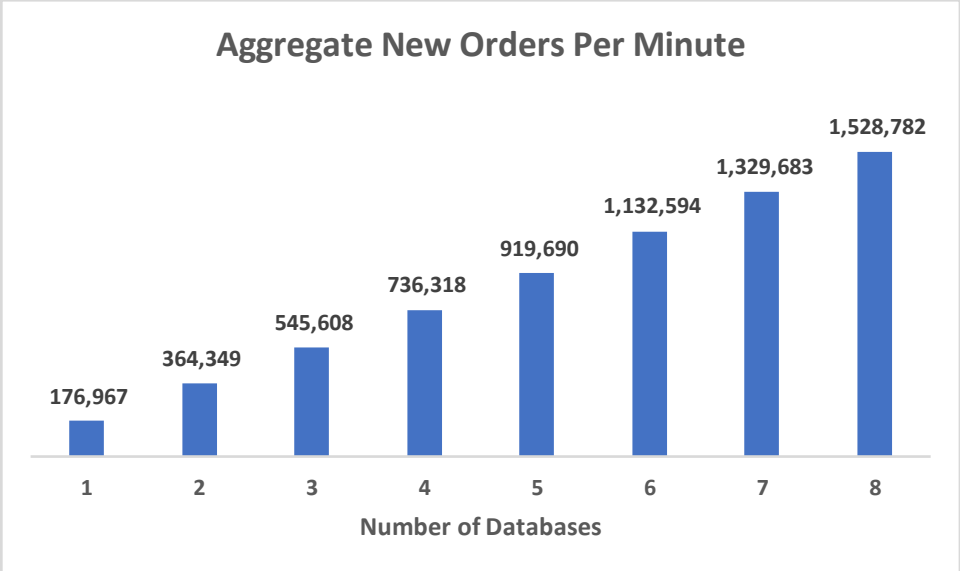
While not charted, average latency was approximately 0.25ms in these test configurations, mirroring previous results.

Bare Metal Small Databases Configuration

Transactions Per Minute

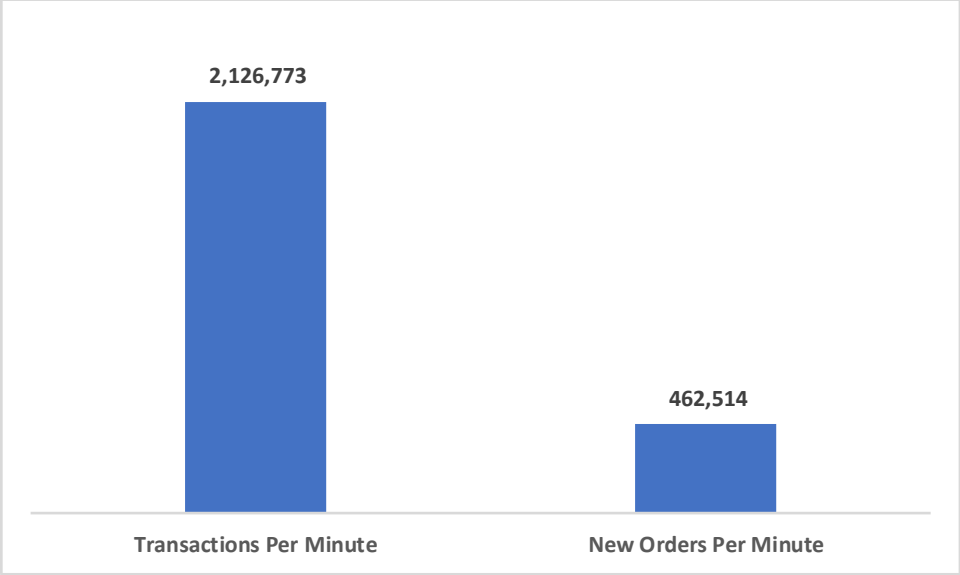


New Orders Per Minute



While not charted, average latency was approximately 0.25ms in these test configurations, mirroring previous results.

Bare Metal Large Database Configuration - TPM and NOPM



While not charted, average latency was approximately 0.25ms in this test configuration, mirroring previous results.

Conclusion

Compared to publicly available performance reports like an October 2018 report* by Principled Technologies that also tested a Dell PowerEdge R740xd running Microsoft SQL Server, the TPM and latency results achieved on Forsa are about 129x* better (22.9 million TPM vs. 177 thousand TPM). While the Principled Technologies report used the DVD Store 2 benchmark and this report used HammerDB TPC-C benchmark, the key difference is of course attributed to running the database on SATA SSD's vs. in memory on LEMs with Forsa.

With over 22.9 million TPM in "host mode" and over 5.1 million TPM and microsecond latency even in a virtualized environment the benchmark results imply significant performance levels can be achieved on a single mid-range Intel server when running applications in memory with Forsa. Because Forsa allows all data to persist in-memory all the time, applications, especially databases, can take advantage of the massively parallel processing capabilities of the memory channel and deliver levels of performance that cannot be achieved when data is stored in traditional mediums like SSDs, HDDs or over a SAN. Additionally, virtualized environments can take advantage of the increased performance to reduce the amount of CPU and RAM assigned to each VM. This, in turn, can allow more SQL Server VMs to be supported on a single host, potentially reducing the license footprint (with associated cost savings) for the organization.

* https://www.principledtechnologies.com/Dell/PowerEdge_R740xd_Microsoft_SQL_2017_RHEL_transactional_database_1018_v2.pdf

Contacts

Jing Xie: jing@formulusblack.com
Josh Williams: jwilliams@endpoint.com