

# ConduSiv's V-locity VM Accelerates SQL Server 2012 by 128% on Virtual Machines without Additional Hardware

Optimizing I/O for Increased Throughput and Reduced Latency

## openBench Labs



## Executive Overview

### WHY READ THIS DOCUMENT?

For this briefing, openBench Labs tested the ability of V-locity® VM to optimize I/O in a SQL Server environment. Using SQL Server 2012, we tested a mix of a high volume of single-row

SQL online transaction processing (OLTP) updates and information retrieval queries, combined with report queries that involved complex database joins. All transactions were generated by daemon processes utilizing random number generators with typing and thinking time intervals before and after queries based on a Poisson distribution with a mean of 25 ms.

In a series of tests simulating external customers and internal sales people creating and placing orders online, openBench Labs measured the ability of IntelliMemory™ technology to offload I/O on read operations through dynamic caching, boosting throughput by providing faster access to data. We also examined the ability of IntelliWrite® to prevent unnecessary split I/Os, using its intelligence to extend current database files and create new log files as single contiguous collections of logical blocks.

In testing SQL Server query processing, openBench Labs found that on a vSphere 5.1 VM running SQL Server 2012, IntelliMemory caching dramatically reduced the average wait times for both SQL Server data locks and latches during query stress tests. By lowering I/O latencies, V-locity VM also provided faster access to data, as measured by a lower average response time, boosted the rate of transactions per second (TPS) that could be sustained, and enabled SQL Server to support more users.

*“By running V-locity VM on virtual machines supporting SQL Server, V-locity VM improved response time by 128%, which in turn boosted average throughput by 50%.”*

### SNAPSHOT OF FINDINGS

- 1) For high-end enterprise OLTP applications, V-locity VM enabled SQL Server to support twice as many SQL client daemon processes while maintaining an average response time under 100 ms. V-locity VM improved response time by 128%, which in turn boosted average throughput by 50%.
- 2) With V-locity VM, we were able to scale to 100 SQL client daemons and sustain a transaction rate of 884 TPS—more than twice the maximum rate we were able to sustain without V-locity VM.
- 3) V-locity VM with IntelliWrite prevented 98% of log file fragments from being created during a typical stress test, contributing to the increase in transaction processing rate and improved response time.



# Executive Briefing: Condusiv's V-locity VM Accelerates SQL Server 2012 by 128% on Virtual Machines without Additional Hardware

Jack Fegreus  
June 10, 2013

## *Optimizing I/O for Increased Throughput and Reduced Latency*

With the release of SQL Server® 2012, Microsoft® is positioning SQL Server as a database for use with large structured and unstructured—dubbed “big data”—applications. Whether with traditional data warehouse scenarios or as a link between unstructured data platforms such as Hadoop, sites with large databases with large numbers of rows are now a key target, which makes SQL Server 2012 a good way for CIOs at data centers that leverage Microsoft technology to initiate big data pilot projects.

### UNDER TEST: VM I/O ACCELERATION Condusiv Technologies' V-locity VM

- |    |   |
|----|---|
| 1) | <b>V-locity's read optimization (IntelliMemory™) technology</b> reduces disk I/O requests by predictively caching active data within available server memory to increase local IOPS performance and lower overhead on shared storage devices by reducing physical read I/O requests on storage devices. V-locity does not cause resource contention on VMs by throttling usage dynamically if an application needs more memory. |
| 2) | <b>V-locity's write optimization (IntelliWrite®) technology</b> provides continuous I/O write optimization using dynamic intelligence when creating or extending files to eliminate split I/Os for greater sequential throughput performance on writes and subsequent reads.  |
| 3) | <b>By optimizing I/O processing</b> , more VMs can run on a host as each VM will generate less I/O traffic to an underlying datastore. Using IntelliWrite and IntelliMemory, V-locity VM boosted transaction throughput on SQL Server 2012 by an average of 50%.  |
| 4) | <b>V-locity's advanced I/O optimization technology</b> is compatible with all advanced storage features, such as replication, de-duplication, thin provisioning, and snapshots.   |

To provide a test case of a large mission-critical application, openBench labs created a 30GB instance of the ANSI SQL Standard Scalable and Portable AS3AP benchmark database and populated it with 32.5 million rows of data.

For IT operations, optimizing I/O of transaction processing applications is complicated by the difference in the rate that CPU and memory performance have advanced versus the rate that disk access times have improved. Attempts to deal with these issues often create CPU bottlenecks as processing stalls waiting for the delivery of data.

With I/O access time highly dependent on disk drive mechanics, IT frequently turns to costly hardware solutions involving

solid-state drives (SSDs). Such hardware solutions, however, suffer from dependence on a hardware platform, under-utilization of resources, lack of flexibility with changing workloads, and high costs. Further complicating these issues, many CIOs are running database-driven applications in a VMware® virtual infrastructure (VI).

In this analysis, openBench Labs examines the ability of Condusiv Technologies' V-locity VM to maximize I/O performance for a VM running Windows Server 2012 and SQL Server 2012 in a VMware vSphere™ 5.1 environment.



Running V-locity on a Windows VM, our SQL load generator initiated transaction loads for groups of up to 100 daemon processes of varying size, each generating multiple SQL Server client requests. By optimizing I/O at the Windows VM OS layer, V-locity optimized the processing of database transactions sent from another VM application.

*“Using sophisticated pattern analysis to determine which blocks will be placed in cache, IntelliMemory offloads a significant portion of I/O read operations from a VM’s logical disks.”*

## REMOVING BARRIERS TO I/O PERFORMANCE

To optimize I/O in a VI, V-locity VM has been designed to resolve two key issues:

- 1) V-locity eliminates nearly all unnecessary I/O operations at the source when writing a file, which in turn eliminates all unnecessary I/O operations on subsequent reads.
- 2) V-locity caches frequently-accessed data to keep read requests from traveling the full distance to storage and back.

## INTELLIWRITE TECHNOLOGY

V-locity VM solves the important issue of unnecessary I/O generation with IntelliWrite® technology. IntelliWrite prevents the Windows OS from storing files as disjointed block sets in its logical block space representation of a logical storage volume.

In a virtualized environment, the problem of superfluous I/O is compounded as multiple VMs share the same storage resource, resulting in highly random I/O behavior pushed down to the disk subsystem. Generation of unnecessary I/Os not only slows the speed of a single virtual machine, it slows other VMs on the same host, as well as any VMs sharing the same storage.

To solve this, IntelliWrite adds intelligence to the way that the Windows OS preallocates file space in order to continuously restructure writes in a coherent manner, store files as contiguous sets of blocks, and prevent performance penalties. Moreover, when a file is accessed and modified in the future, IntelliWrite will automatically restructure that file for optimal I/O performance.

## INTELLIMEMORY TECHNOLOGY

Optimizing writes and eliminating unnecessary I/O operations does not resolve all important data access issues, especially when reading data. To provide the full spectrum of I/O optimization, V-locity VM implements IntelliMemory, a highly efficient data caching solution that leverages available server memory to provide faster access to data and dramatically improved throughput. Using sophisticated pattern analysis to determine which blocks will be placed in cache, IntelliMemory offloads a significant portion of I/O read operations from a VM’s logical disks. More importantly, by offloading physical I/O, V-locity VM helps improve performance on any VMs sharing the same storage resources via a SAN or NAS.



## THE TEST

To assess the ability of V-locity VM to optimize I/O in a VMware vSphere VI, we configured a test scenario using three VM servers. Within this scenario, we set up SQL Server 2012 on a VM running Windows Server 2012 to set up a database-driven production environment characterized by simultaneous complex batch reports and OLTP queries.

The following is an overview of the test environment and process:

- 1) VM running Windows Server with six CPUs, 8GB RAM, and an independent logical disk for SQL Server 2012.
- 2) An ANSI AS<sup>3</sup>AP database configured with 20 million data entries, which translated into a 15GB database.
- 3) We configured SQL Server to use a maximum of 5GB of server memory.
- 4) We set up two VMs running Windows Server 2008 R2 and a SQL load generator to simulate a mixed load of batch reports and interactive OLTP queries.
- 5) We leveraged the AS<sup>3</sup>AP benchmark's random number generator to create distinct single-row queries for each daemon OLTP user and each batch process by inserting random data into fixed templates.
- 6) All OLTP queries included an interval of “typing time” before each transaction and an interval of “think time” after each transaction based on a Poisson distribution with a mean of 25 ms.
- 7) We terminated all test runs when the average response time of OLTP queries exceeded 100 ms—a heuristic based on user perception studies about when systems are considered to be reacting instantaneously.

## MEASURING RESULTS

Our goal was to assess the ability of V-locity VM to foster an environment in which SQL Server 2012 could reach maximum performance and scalability. To measure the impact of V-locity VM, we created a scenario that involved a fixed baseline level of background batch production processing and varying levels of OLTP activity.

The base level of production report processing was anchored by a fixed load of distinct subtotal- and total-report queries that involved database joins of tables that contained 20 million rows of data. In addition, we launched OLTP client daemons to simulate external customers or internal sales people placing online orders.

Our client daemon processes initiated a mix of single-row queries that involved either information retrieval or data updates. Client daemons were launched in groups of five in intervals of five minutes. We terminated any test when the average response time for all OLTP queries being generated by client daemons exceeded 100ms.

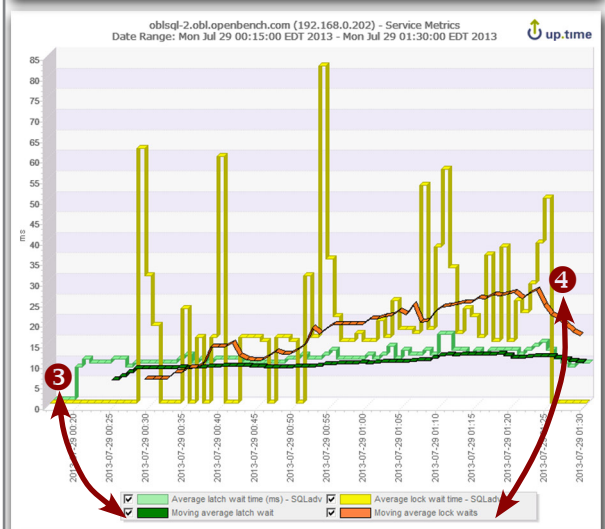
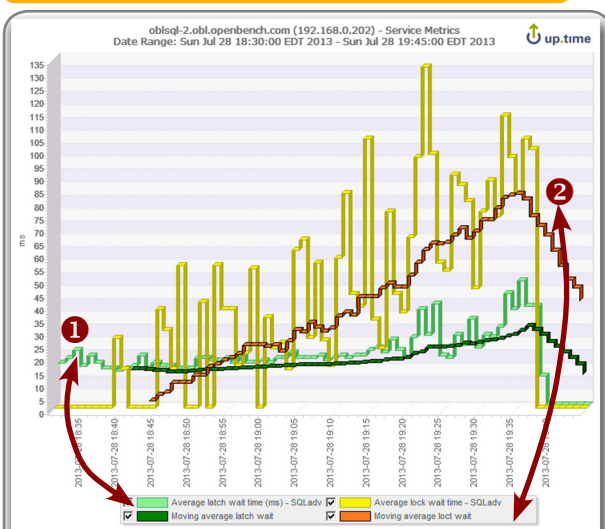
To start our analysis of V-locity VM, we ran SQL stress tests on our ANSI AS<sup>3</sup>AP database with the default SQL Server 2012 memory configuration. The default memory configuration allows SQL Server to utilize all the memory provisioned in a server; however, in all of our tests, SQL Server 2012 conservatively claimed a narrow range of memory that ranged from 3GB to 4GB.

Given the memory utilization pattern measured in our initial tests, we reset the maximum memory available to SQL Server from its default value to 5GB. After making that change, overall TP throughput for SQL Server improved about 5% in our test workloads, as SQL Server was immediately placed in an optimal environment, in which to run the transactions associated with our application.

*“Using its RAM cache, V-locity VM reduced disk IOPS by caching 67% of the application database queries and the tempdb system database reads not cached by SQL*

## THE RESULTS: A CLOSER LOOK AT SQL SERVER

### V-LOCITY VM SQL SERVER PERFORMANCE



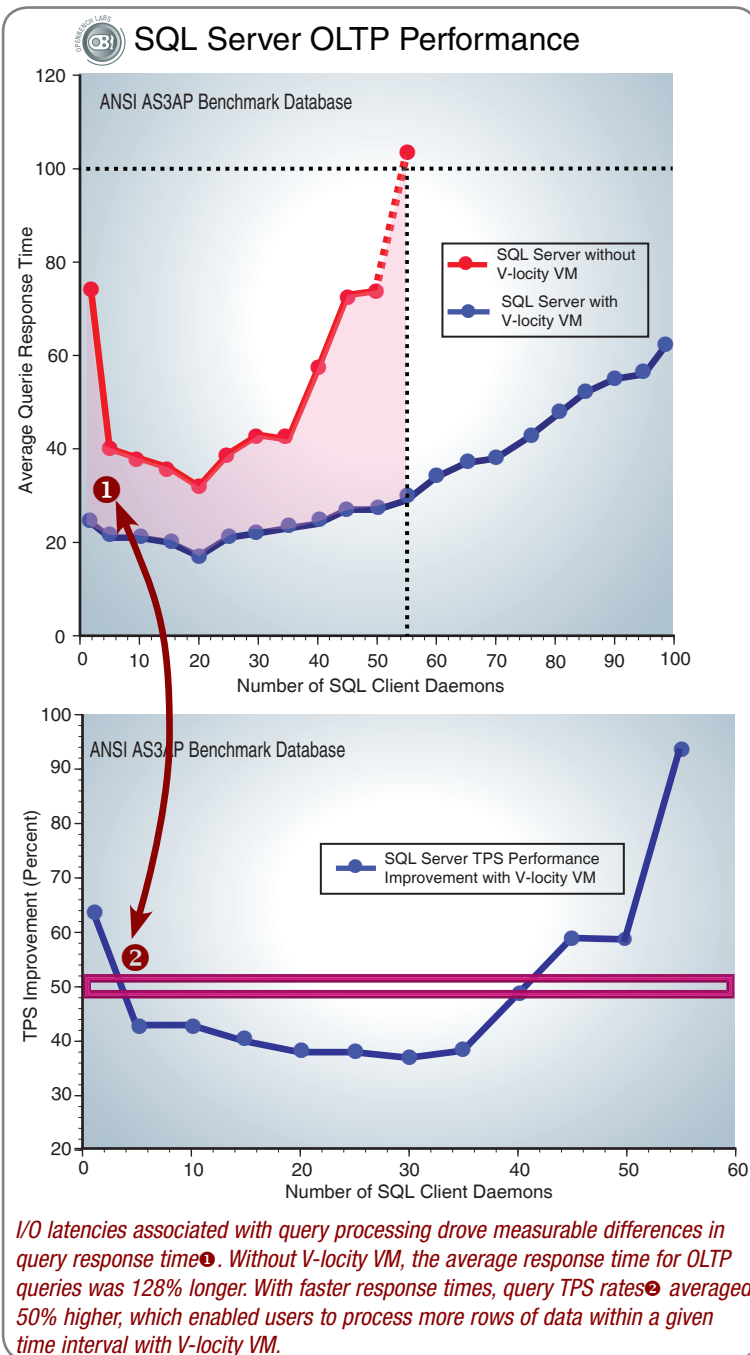
Without V-locity, average wait times for SQL Server latches<sup>1</sup> and locks<sup>2</sup> increased faster as we added daemons, impacting query response time. With V-locity, wait times for latches<sup>3</sup> and locks<sup>4</sup> decreased, dramatically improving query response time.

When we ran our SQL Server tests with V-locity VM in our modified environment, IntelliMemory was able to claim 2GB of memory for caching on the start of a test. Using IntelliMemory and its RAM cache, V-locity VM reduced disk IOPS on the VM by caching 67% of the application database queries and the tempdb system database reads not cached by SQL Server. More importantly, as V-locity VM lowered the total number of physical IOPS over the server's disk subsystem, it mitigated multiple data access latency issues and thereby improved the TPS rates of SQL Server queries.

In particular, V-locity VM caching dramatically reduced the average wait times for both SQL Server data locks and latches during query stress tests. SQL Server uses these mechanisms to ensure data integrity by synchronizing data access for multiple queries attempting to modify data in one or more rows of one or more database tables. Without V-locity VM, average wait times for latches and locks were three times longer, which directly impacted average query response time

Database locks are typically used when data is being accessed on disks, while latches are used when SQL Server moves data from disk to its page cache. In both cases, any other disk requests, which may involve other SQL Server data or general system I/O, will openly compete for disk resources and extend lock and latch wait times. More importantly, as the I/O latency associated with those key internal SQL Server I/O mechanisms increases, it creates enough I/O overhead to raise the average response time, lower the TPS rate, and lower the work completed for user queries.

## OLTP PERFORMANCE IMPROVEMENT



As we increased the number of client daemons generating OLTP queries without V-locity VM, I/O latency associated with database latch and lock wait times rose dramatically. In particular, as the number of SQL client daemons increased from 1 to 55, the moving average of latch wait times increased from 20 to 30 ms.

Even more dramatic, lock wait time increased by a factor of four, as the moving average increased from an initial level of 20 ms to 80 ms. More importantly, with the average wait time for disk locks taking 80 ms, the average response time for OLTP queries ballooned to more than 100ms. As a result, we ended OLTP query testing at 55 daemon clients without V-locity VM running.

When V-locity VM ran in conjunction with SQL Server, the moving averages for latch and lock wait times were dramatically shorter. What's more, increasing the number of SQL client daemons had much less of an effect. With V-locity VM, average wait times for latches remained essentially constant at 10 ms, while the moving average for lock wait times increased from 10 to 25 ms.

The strong 300% internal SQL Server I/O latency advantage that we measured with respect to latch and lock waits, manifested itself as a 54% advantage in the average query response time during our OLTP tests. In terms of the work being processed by queries, metrics such as improved query TPS rate and the number of

database rows processed during a test iteration, reflected a 50% improvement in work performed by user queries with V-locity VM.

Average query response time was at a minimum for tests with and without V-locity VM, when there were 20 SQL client daemons generating queries. At this point, average query response time was 17 ms with V-locity and 32 ms without V-locity. With the



average query response time 1.8 times longer without V-locity compared to queries processed with V-locity, transaction processing rates also differed significantly: 20 client daemons sustained 292 TPS with V-locity and only 211 TPS without V-locity.

Table 1: V-locity VM with SQL Server 2012 Test Results

V-locity VM Performance: SQL Server Stress						
1 to 100 Daemon Loads Sampled in 5-Daemon Increments for a 30GB Database						
I/O Activity	TPS Rate With V-locity VM	TPS Rate Without V-locity VM	TPS Rate Improvement With V-locity VM	Response Time With V-locity VM	Response Time Without V-locity VM	Response Time Differential Without V-locity VM
1 daemon	13	8	64%	25 ms.	74 ms.	196%
5 daemons	70	49	43%	21 ms.	40 ms.	90%
10 daemons	133	93	43%	21 ms.	38 ms.	81%
15 daemons	221	158	40%	20 ms.	36 ms.	80%
20 daemons	292	211	38%	17 ms.	32 ms.	88%
25 daemons	359	261	38%	21 ms.	39 ms.	86%
30 daemons	427	312	37%	22 ms.	43 ms.	95%
35 daemons	481	347	39%	23 ms.	42 ms.	83%
40 daemons	538	364	48%	24 ms.	57 ms.	138%
45 daemons	576	363	59%	27 ms.	73 ms.	170%
50 daemons	646	406	59%	27 ms.	74 ms.	174%
55 daemons	693	358	94%	29 ms.	103 ms.	255%
60 daemons	735			34 ms.		
65 daemons	759			37 ms.		
70 daemons	778			38 ms.		
75 daemons	797			42 ms.		
80 daemons	812			47 ms.		
85 daemons	818			52 ms.		
90 daemons	841			55 ms.		
95 daemons	880			56 ms.		
100 daemons	884			64 ms.		
V-locity VM Database I/O Caching (IntelliMemory)						
Disk Reads	5,086,564 - 33%	N.A.				
Cache Reads	3,639,095 - 67%	N.A.				
I/O Latency	3.3 ms	7.5 ms.				
V-locity VM Database Log File Fragment Prevention (IntelliWrite)						
File Fragments	1,303 Prevented with V-locity	1,251 Fragmented without V-locity				



When we reached 50 SQL client daemons, average response time was 27 ms with V-locity and 74 ms without V-locity, and the transaction processing rate for OLTP queries was 578 TPS and 406 TPS respectively.

*“Using V-locity VM, IT can lower operating expenses by adding more user processes to a VM application and increasing the number of VMs running on a host—without increasing overhead costs through additional storage hardware.”*

Without V-locity VM, 50 SQL client daemons was the highest number of clients that our test configuration could support without exceeding 100 ms as the average OLTP query response time.

At that point, adding more daemons without V-locity VM caused the average transaction response time to balloon further and the TPS rate to plummet. In contrast, we were able to scale to 100 SQL client daemons with V-locity VM.

## **BOTTOM LINE**

By running V-locity VM on virtual machines supporting SQL Server, internal SQL Server operations dependent upon data access will exhibit lower I/O latency and perform more efficiently. More importantly, IT can provide Line of Business users with significantly improved application performance.

Moreover, using V-locity VM, IT can lower operating expenses by adding more user processes to a VM application and increasing the number of VMs running on a host—without increasing overhead costs through additional storage hardware. As a result, IT has a powerful tool to maximize the ROI associated with any VI application initiative driven by SQL Server.

Westborough, Mass.-based openBench Labs was founded in 2005 by Dr. Jack Fegreus. openBench Labs is a trusted IT industry source, providing hands-on evaluation and certification of Information Technology products and services. openBench Labs enjoys a unique position in the information technology sector. As the premier independent test lab and provider of third-party validation services, OBL has worked with virtually every major vendor and evaluated the most important products and technologies to appear over the past decade.

© 2013 OpenBench. All Rights Reserved. V-locity, IntelliWrite, InvisiTasking and IntelliMemory are trademarks or registered trademarks of Conduvis Technologies Corporation. All other trademarks are the property of their respective owners.