

WHITE PAPER

# VOD 8.0 AND SOLID STATE DEVICE PERFORMANCE ANALYSIS

By Per Jochumsen, Senior Software Engineer

Sponsored by Versant Corporation



Versant Corporation U.S. Headquarters

255 Shoreline Dr. Suite 450, Redwood City, CA 94065

[www.versant.com](http://www.versant.com) | 650-232-2400

---

## INTRODUCTION

As Solid State Devices become more commonplace, it is interesting if and under what circumstances VOD can benefit from the high I/O performance of these devices.

We are especially interested in data that shows which parts/files of a database should be stored on an SSD to maximize performance gains: system volume, data volumes, logical log and/or physical log.

The tests for these performance analyses were performed using a high-end SSD (fusion-io's ioDrive 80 GB).

---

## SYSTEM CONFIGURATION

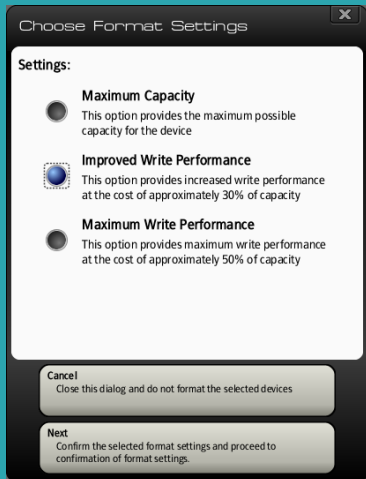
All tests are run on a Dell Precision T3500 workstation with Intel Xeon 4 core CPU, 6GB memory, SATA HDD 7.200 rpm (no RAID). The operating system is Windows XP x64 SP2.

The Versant Object Database system is VOD 8.0 GA (Backend Profile PROFILE.BE as documented in the Appendix).

The ioDrive was installed on a PCI-Express 16x slot and low level formatted using the shipped ioManager utility from fusion-io.

We used the setting "Improved Write Performance", accepting the reduced disk capacity.

The ioDrive was formatted as a NTFS volume with fixed sector size of 2k.



---

## Important Note for Windows

Mircosoft offers a feature, "write caching", to speed up storage devices. This has to be turned *off* , otherwise data loss and corruption is possible, which we definitely do not want for Enterprise Databases (a "flush to disk" command is not a physical flush when write cache is active).

The device driver of the ioDrive does not offer this feature at all, but for SATA devices and for some RAID controllers disabling the write caching feature should be ensured.

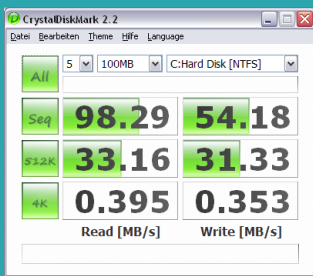
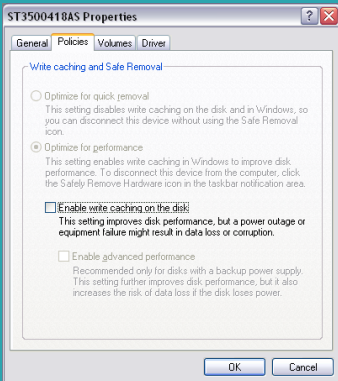
This is how you configure this feature:

- » **Start an Explorer**
- » **Choose the "Properties" dialog for the drive involved**
- » **Choose register card "Hardware"**
- » **Select the "Device" in the list box**
- » **Press "Properties" to open the dialog you see on the right.**

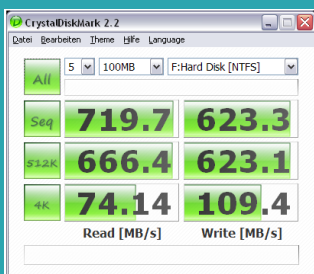
---

## Benchmark Performance

Crystal DiskMark Performance measurements gave us a first impression of the maximal I/O throughput for the ioDrive and the build in SATA HDD.



Hard Disk Drive



Io Drive SSD

---

## VOD8 Performance

We used a fixed set of database configurations for all performance tests:

**ALL ON HDD** All database files on the hard disk: sysvol, datavol, logical log and physical log as well as systrace, LOGICAL and all other files.

**ALL ON IO** sysvol, datavol, logical log and physical log on the ioDrive  
all other files on the HDD

**LOGS ON IO** logical log and physical log on the ioDrive  
all other files on the HDD

**VOLS ON IO** sysvol and datavol on the ioDrive  
all other files on the HDD

**LLOG ON IO** logical log on the ioDrive  
all other files on the HDD

**PLOG ON IO** physical log on the ioDrive  
all other files on the HDD

---

## Simple Test

The first database test we ran is a simple test. It generates lots of objects in small transactions. One object per transaction. This keeps the logical log very busy as committing lots of small transactions needs I/O flushing.

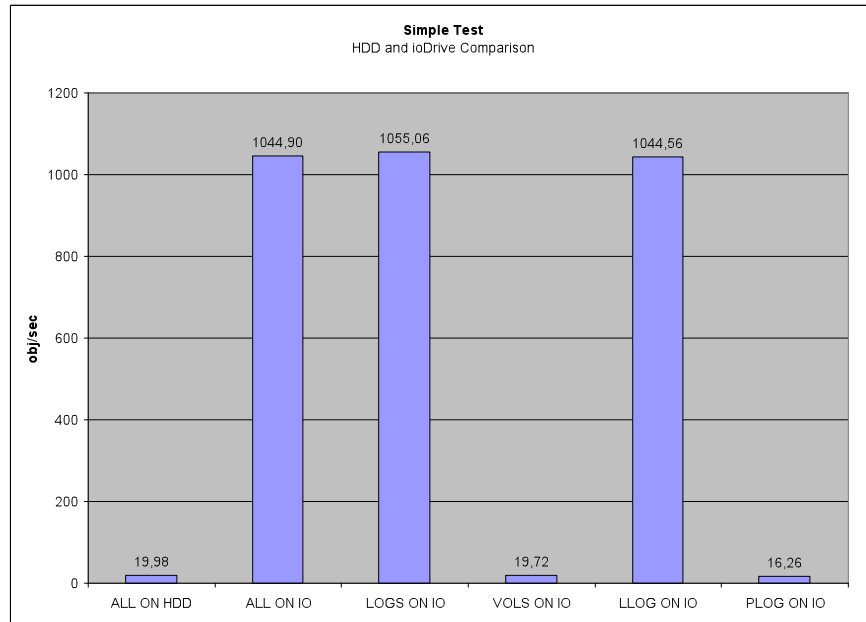
The physical log and the volumes are busy with I/O as well, but this happens asynchronous and the server buffer helps optimizing these I/O actions.

We measured the following: The ioDrive beats the HDD by a factor of 19 up to 52 when it comes to "objects created per second".

The factor depends on the size of each object and the total number of objects created.

This tells us: Yes, VOD is able the benefit from the great throughput of the ioDrive.

Here is one of the measurements:



It is not expected that we will reach the same performance factor in other, more complex tests were logical overhead, concurrency management and other aspects are of more importance.

For this simple test we have 98% of the over all runtime used up for commit and checkpoint actions executed on the server. This is a very heavy I/O load.

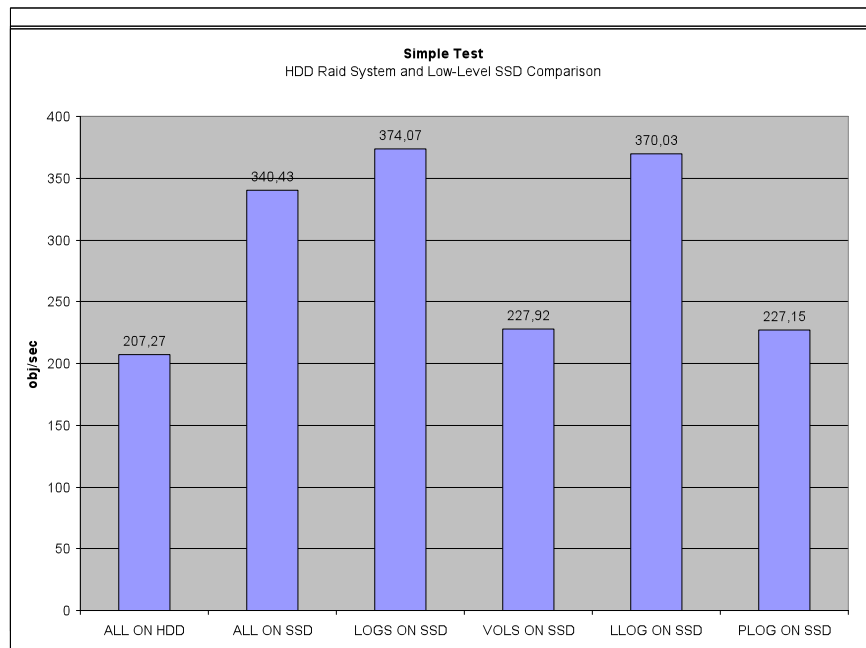
---

## Digression

We also ran the **Simple Test** with a medium priced SSD in another system environment:

- » **Solaris Intel server**
- » **SSD Intel X25E Extreme 64 GB**
- » **RAID-0 HDD**

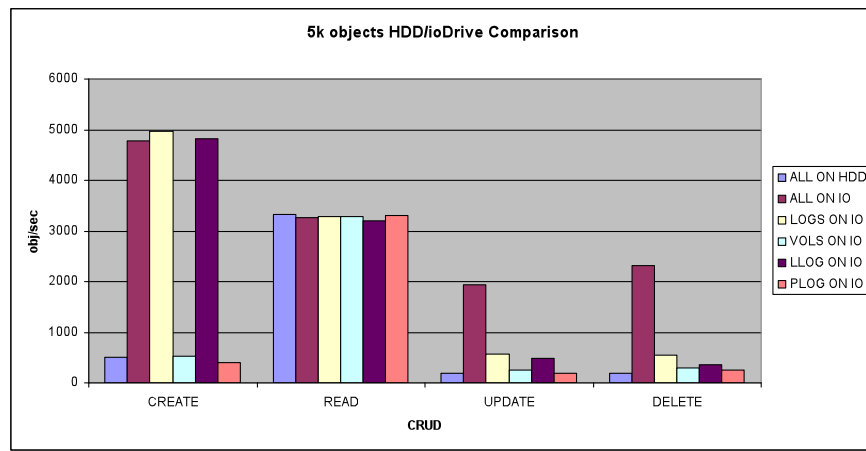
In this case we measured a ratio of only 2:1 of SSD compared to RAID-0 HDD.



# CRUD Test

## (Create-Read-Update-Delete)

The CRUD is one of our standard tests and often taken into account for internal performance tests. We did the HDD vs. ioDrive comparison with CRUD as well and here are the results.



As can be seen in the chart: Create, Update and Delete are much faster with the ioDrive, especially when at least the logical log is placed on the SSD (be careful when looking at the Read results, the VOD server cache does a very good job here).

For Update and Delete there is (as for Create) a lot of activity on the logical log. In addition, we also have a lot of activity on the data volume (which is very random). This explains why the configurations "LOGS ON IO" and "LLOG ON IO" are not as fast as in the Create scenario.

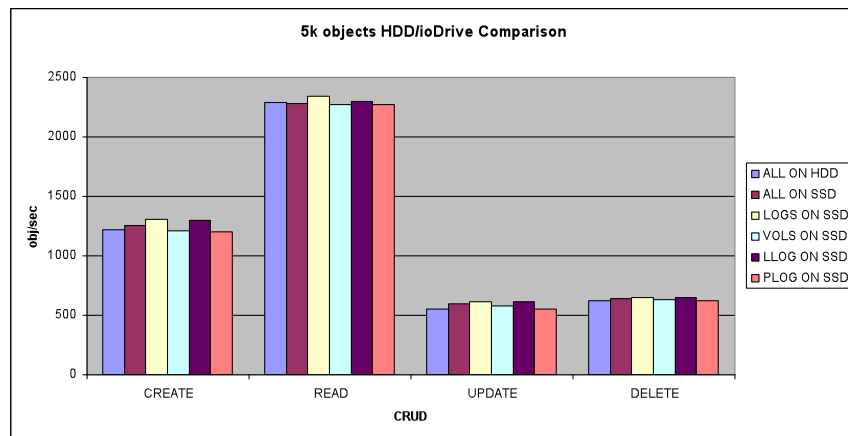
For these cases the configuration "LLOG AND VOLS ON IO" is faster. This is not in our chosen set for this document, but we ran a short test and it verifies: The "LLOG AND VOLS ON IO" configuration reaches 2/3 of the "ALL ON IO" numbers.

## Digression

We also ran the **CRUD Test** on the Solaris System as we did for the **Simple Test** (see digression Page P006 - there you find more details for the system environment).

The goal was to compare the performance of the high end ioDrive with the Intel X25E Extreme SSD.

The performance gains in comparison to the HDD are relatively low on the system with the Intel X25E Extreme SSD (4%-11%). Again the highest rates we gain are for the configurations with the logical log on the SSD.



But comparing the absolute numbers we see:

The hard disk performance is much better with RAID-0 HDD system in the Solaris machine (page P008 - with Intel X25E Extreme SSD). It is faster by factor 2,5 in comparison to the single SATA HDD in the Windows system (page P007 - with the ioDrive).

So we have a much faster hard disk configuration on this Solaris machine.

On the other hand the SSD in the Solaris system is not as fast as the ioDrive.

This is easy to see if you look at the configurations "ALL ON HDD" and "ALL ON IO/SSD" only.

The performance gains of the Intel X25E Extreme SSD compared with the RAID-0 HDD system is minimal.

## PolePosition

Open Source Database  
Benchmark

<http://www.polepos.org>

---

## PolePosition

Pole position is an independent Open Source Database Benchmark (<http://www.polepos.org>). We executed the Test Circuits "Melbourne" and "Bahrain."

Remember: The simple test (Page P004) uses 98% for commit and checkpoint actions and these are the actions generating the I/O flushing. Of course there are I/O operations during the remaining runtime (mostly read actions) but these can often be optimized by buffering data.

---

### Melbourne Circuit

The charts illustrate:

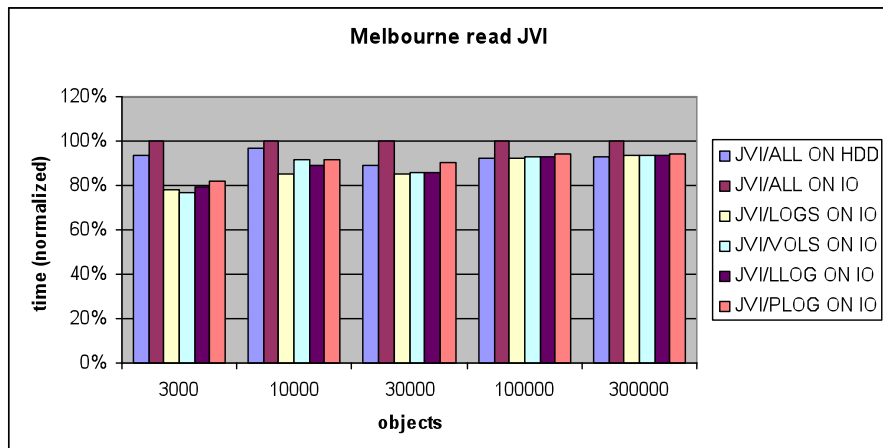
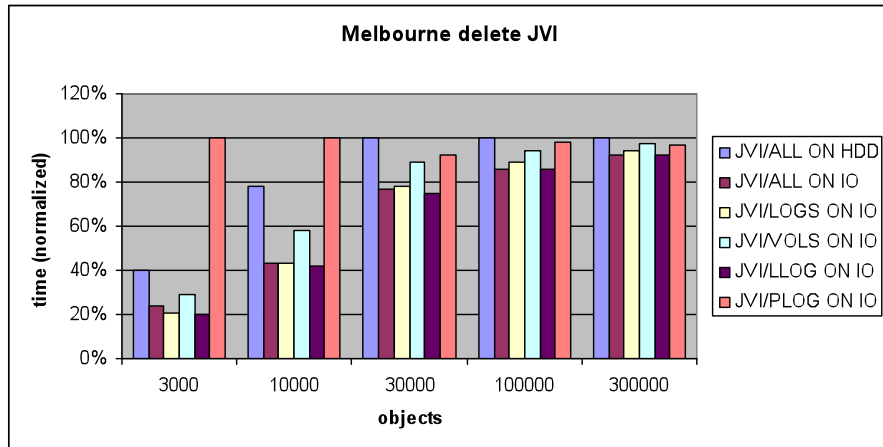
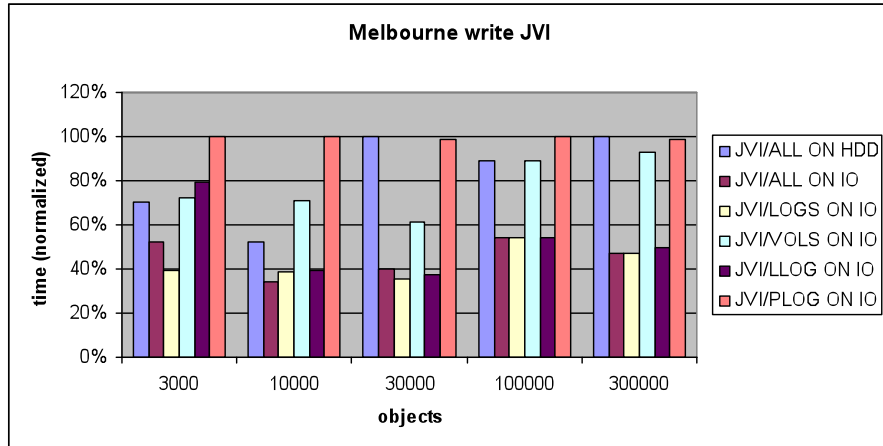
- » **Significant performance gains for write (factor 2-3), significant improvements for delete (10-20%) in all configurations where at least the logical log is placed on the ioDrive.**
- » **hot read and read: As we have seen with the CRUD test there is little difference between HDD and SSD**

## "Melbourne":

Write, read and delete unstructured flat objects of one kind in bulk mode;

19% of the test runtime is used for commit and checkpoint actions

**Important:** All charts are normalized to 100%, so low values are good values.



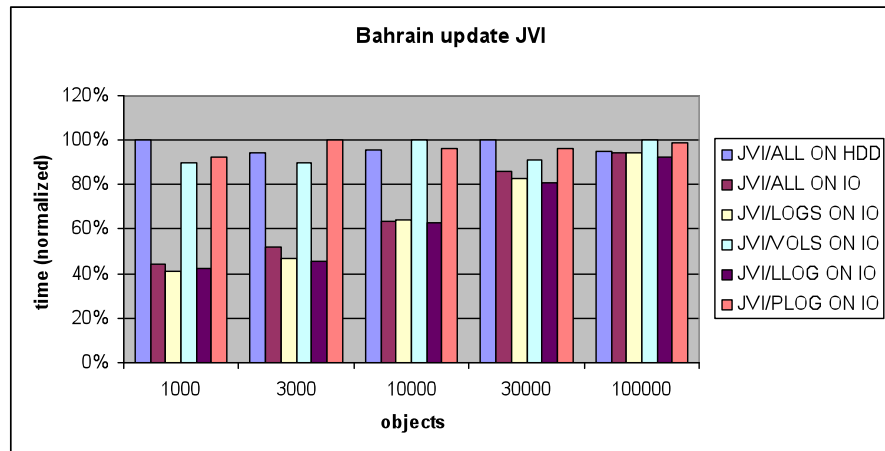
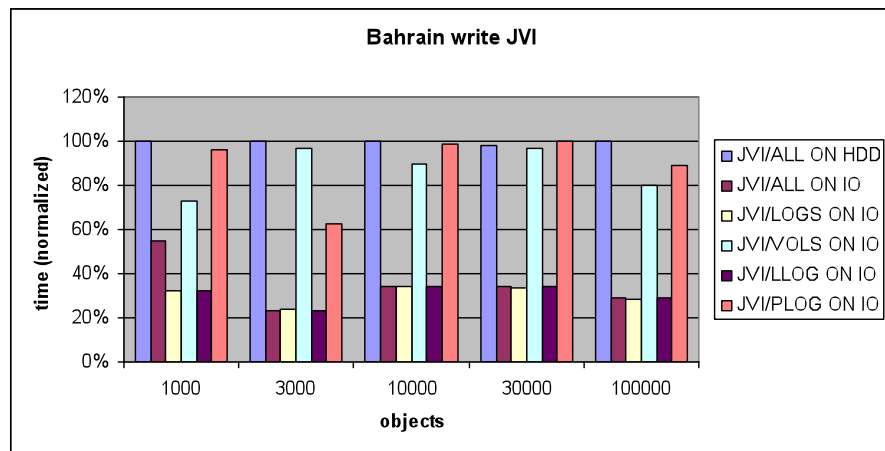
## "Bahrain":

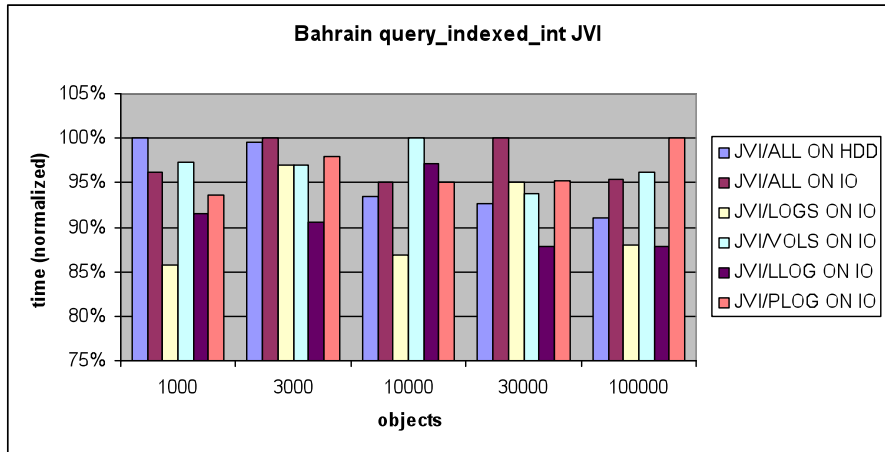
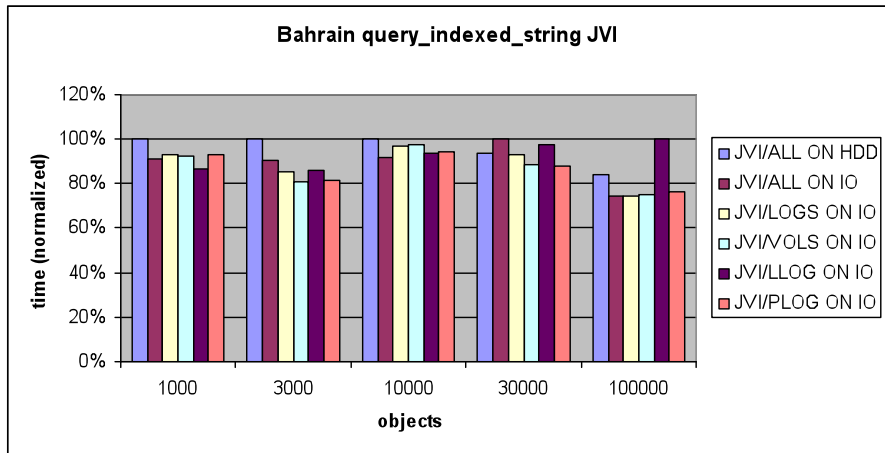
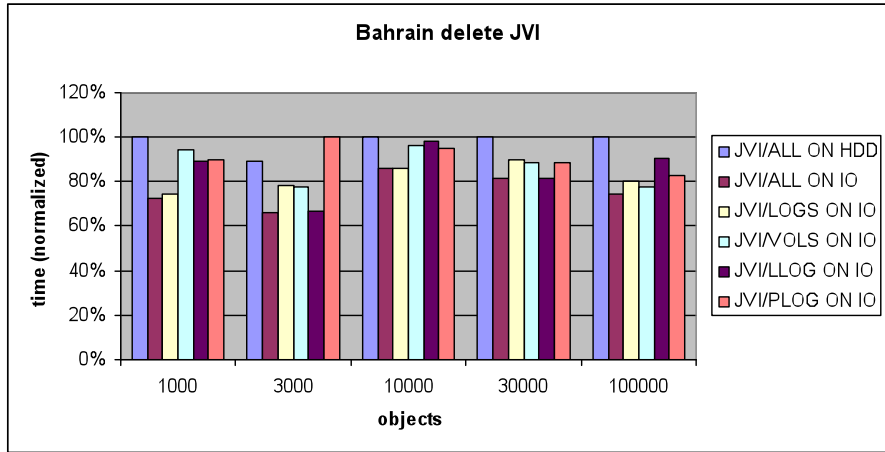
Write, query, update and delete simple flat objects individually;  
22% of the test runtime is used for commit and checkpoint actions

## Bahrain Circuit

The charts illustrate:

- » Significant performance gains for write (at least factor 3) in all configurations where at least the logical log is placed on the ioDrive.
- » Noticeable improvements in for delete and for update, again especially for configurations where at least the logical log is placed on the ioDrive.
- » For query\_indexed\_string and query\_indexed\_int a rather mixed picture, but again performance gains for the SSD settings.







## Per Jochumsen

Senior Software Engineer,  
Versant Corp.

Per Jochumsen is responsible for server side transaction, logging performance and scalability. He is a member of the V/NET and V/JDO team.

Per Jochumsen joined Versant in 2009. Prior to working for Versant he designed and implemented Data Capturing and Data Management Solutions, cooperating with various DBMS.

---

## Conclusion

Using high-performance SSD techniques as provided by the fusion-io ioDrive does improve Versant Object Database performance. The limited capacity of SSD as well as the measurements presented in this document suggest that the first part of a database to move to a SSD should be the logical log.

The results in this document were reached by comparing a consumer class hard disk with an expensive high end SSD solution.

Using enterprise RAID systems or a less expensive SSD will lead to results more in favor of the hard disk solution (but keep in mind that a SSD can and should be used in RAID systems as well).

The performance gain also depends on the I/O insensitivity of the application. The performance advantages of ioDrive can only affect the portion of the total application runtime where I/O data transfer is done.

The more I/O intense an application is, the more it benefits from a quick ioDrive.

---

## Appendix

### PROFILE.BE

```
sysvol                1024M system
datavol               voll  4096M voll
plogvol              50M   physical.log
llogvol              50M   logical.log
extent_size          2
logging              on
locking              on
commit_flush         off
polling_optimize     off
async_buffer_cleaner 1
async_logger         1
event_registration_mode transient
event_msg_mode       transient
event_msg_transient_queue_size 20480
bf_dirty_high_water_mark 512
bf_dirty_low_water_mark 204
class                2000
db_timeout           -1
index                2000
llog_buf_size        20M
lock_wait_timeout    60
max_page_bufs        8192
multi_latch          on
plog_buf_size        20M
heap_size            0
heap_arena_size      128K
heap_arena_size_increment 64K
heap_arena_trim_threshold 2M
heap_max_arenas      -1
heap_arena_segment_merging on
transaction          200
user                 20
volume               16
stat                 all   off
assertion_level      0
trace_entries        10000
```

```
trace_file          systrace
versant_be_dblogginglevel 1
be_syslog_level     1
treat_vstr_of_lb_as_string_in_query  off
blackbox_trace_comps  rpc
```