Scalability Performance Assessment of Large Astronomical Databases

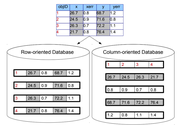
A comparison of row-oriented vs column-oriented databases for the Vista Variables in the Via Lactea (VVV) Survey

Maria Patterson[1], Iraklis Klampanos[2], Ross Collins[3], Nicholas Cross[3], Mark Holliman[3], Robert Mann[3], and Malcolm Atkinson[2]

[1]Open Science Data Cloud, NSF PIRE Fellow; [2]University of Edinburgh, School of Informatics; [3]University of Edinburgh, Royal Observatory, Edinburgh

Background

The success of large astronomical sky surveys such as the Sloan Digital Sky Survey (SDSS) has revolutionized data collection in observational astronomy. Small scale telescope observations performed by individual astronomers are becoming increasingly less common, in favor of dedicated surveys producing massive data sets. SDSS has produced 50 TB of data with an 18 TB object catalog of 357 million unique objects (Abazajian et al., 2009). The upcoming Large Synoptic Survey Telescope (LSST) will produce 60 PB with an object catalog of 20 billion rows (Ivezic et al., 2008). Accompanying the wealth of new data are the problems associated with handling very large data sets. Astronomers are no longer limited by the amount of data, but by the ability to quickly and easily handle large data sets and perform analysis.



This paper assesses the relative performance of two database management systems (DBMS) for data from the VISTA Variables in the Via Lactea (VVV) Survey a large with the goal of detecting variable stars across the Galactic plane and bulge (Saito et al., 2012, Cross et al., 2012). The data collected have a detection table expected to grow to over 1e11 rows and 132+ columns. Using these detection data, we compare the performance of MS SQL Server 2012, the currently implemented row-oriented DBMS to MonetDB, a column-oriented DBMS. The goal of this research is to quantify the possible gain in time and resource use efficiency with the implementation of a column-oriented DBMS.

Experimental Setup

Virtual Machines

We run each database on a dedicated virtual machine created through the KVM Virtualization platform.

Each VM is given the same hardware configuration, as shown in Table 1 below.

	MS SQL Server	MonetDB
Operating System	Windows Server 2008 R2	Debian Linux jessie
Physical Disks	1	
Cores	12	
Memory	12 GB	
Disk Space	6 TB	

Table 1: Virtual Machine Configuration

Small and Large Data Sets

For scalability testing, we compare each database's performance handling SQL queries on

1) a small table that fits in memory
2) a large table that does not.

These are subsets of the actual VVV Survey object detection table and are described in Table 2 below. Currently, we have only used vvvDetectionSmall.

	vvvDetectionSmall	vvvDetection
Columns	132	
Primary Key	3 columns (multiframeID, extNum, seqNum)	
Rows	1e7	5e8
File Size	~12 GB	~600 GB

Table 2: Test Data Set Characteristics

System Monitoring with Python 'psutil'

We have written and implemented a monitoring program that retrieves performance statistics via the Python module **psutil** (`code.google.com/p/psutil/`, version 1.0.1).

Because psutil is cross-platform, it offers a consistent method for tracking process and system utilization information in both Windows and Linux.

Throughout each test query, we retrieve the following information, in addition to the total runtime of the task.

System statistics

- CPU utilization (percentage)
- disk I/O statistics (number, size in bytes, and time in ms of each read and write)

Process stats (per instance of database server)

- disk I/O statistics (number and size in bytes)
- threads used
- voluntary and involuntary context switches
- CPU user and system times
- memory – RMS and VMS (bytes)

MS SQL Server vs MonetDB Performance

We compare the performance of MS SQL Server vs MonetDB with the small **vvvDetectionSmall** data table by monitoring three tasks described below.

These tasks are currently used by the Vista Science Archive for the VVV Survey and are typical of tasks used for astronomical catalog data curation.

Task A: Bulk loading data

This task involves inserting all data for the entire test table vvvDetectionSmall into each database. The data is read from a csv file mounted in a shared location to each VM. We use BULK INSERT to load the data in MS SQL Server and the equivalent command COPY INTO in MonetDB.

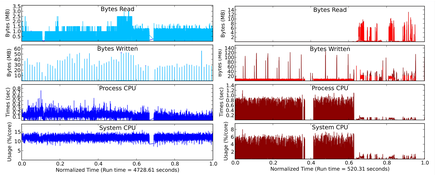


Figure 1: Bulk loading data into vvvDetectionSmall with MS SQL Server (left blue) and MonetDB (right red).

Task B: Inserting & updating sequential IDs

This task involves two queries to add rows of new data to the existing object detection table, which might be done after a night's or month's worth of observations are completed.

The current process for adding data to the VVV Survey detection table involves

1) inserting the new rows of data with the objID attribute as some negative number (top figures) and
2) updating these negative objIDs to provide sequential unique IDs for every detection in the table (bottom figures).

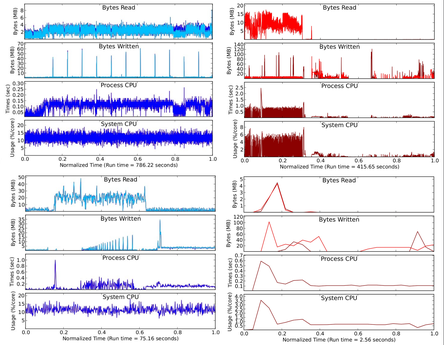


Figure2: Inserting data (top) and updating seq. Ids (bottom) in vvvDetectionSmall with MS SQL Server (left blue) and MonetDB (right red).

Task C: Column updates for recalibration

This task involves updating several columns for all detections in a single detector (i.e., updating several columns for all rows of a subset of the referenced primary key), which is necessary for the recalibration of photometric or astrometric measurements.

We monitor a 20-column update for two subsets of the detection tables equivalent to two types of detectors:

1) *pawprint* with 2e4 rows
2) *tile* with 1e6 rows of detections

These subsets are determined by an appropriate WHERE clause selecting on a combination of multiframeID and extNum.

These *pawprint* and *tile* detection subsets are chosen to be completely contained within the vvvDetectionSmall table, which is itself a subset of the vvvDetection table, ensuring that the same number of rows are updated in each table and only the total number of rows is different between the queries run on vvvDetectionSmall and vvvDetection.

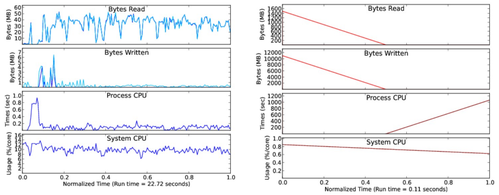
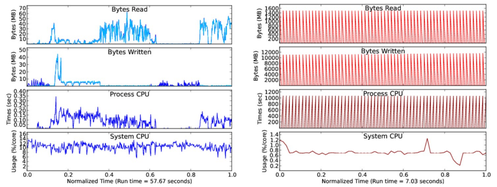


Figure: Column updates for recalibration in vvvDetectionSmall with MS SQL Server (left blue) and MonetDB (right red). Above shows the results for updating a 'pawprint' detector (~2e4 rows of detections). Below shows the results for updating a 'tile' detector (~1e6 rows of detections.

In this task, the difference in run time between MS SQL and MonetDB is the most significant.



Results

We present a run-time summary of the two database systems with the three tasks in Table 3. MonetDB significantly outperforms MS SQL Server in terms of speed for each query tested. The tasks run in MonetDB are anywhere from a factor of 2-200 times faster than the same tasks run in MS SQL Server. Column updates in particular are much faster with MonetDB than MS SQL Server, which is expected given its column-oriented nature.

In the various figures, we show CPU and I/O activity during the execution of the queries involved in each Task in MS SQL Server (blue) and MonetDB (red). The run time for each query is given along the x-axis. Given our results from the three tasks tested, MonetDB appears to be a faster and more efficient database management system for the VVV detection table data.

An ongoing analysis of the two DBMSs for use with the VVV survey can continue this work in a number of ways. First, an extended analysis with various larger data sets would be useful to test the scalability of each DBMS for databases too large to fit in memory, like the full VVV detection table. Additionally, this analysis was limited to testing of queries with one table. It would also be useful to compare each DBMS's performance across multiple tables using JOIN.

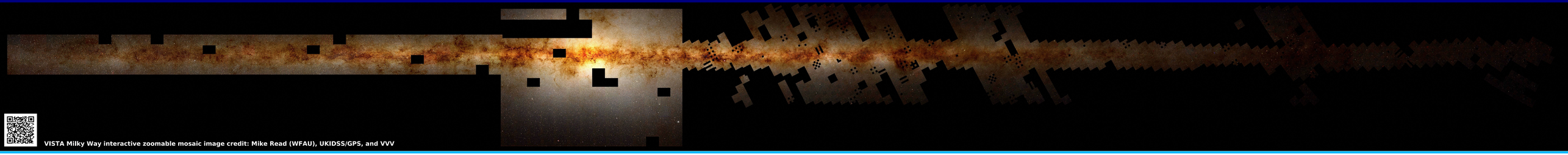
Table 3: Run-time summary (seconds)	vvvDetectionSmall				
	Task A	Task B		Task C	
		Part 1	*Part 2*	*Part 1*	*Part 2*
MS SQL Server	4729	786	75.2	22.7	57.7
MonetDB	520	416	2.6	0.1	7.0

Summary

We assessed the relative performance of two database management systems (DBMS) for a large astronomical detection catalog from the VISTA Variables in the Via Lactea (VVV) Survey, a sub-survey of the Visible and Infra-Red Survey Telescope for Astronomy (VISTA). The VVV is a multi-epoch and multi-band survey aimed at detecting variable stars across the Galactic plane and bulge. The data collected have a detection table growing to over 1e11 rows and 132+ columns. For database performance testing, we used a subset of the data (1e7 rows) and compare the performance of MS SQL Server 2012, the currently implemented row-oriented DBMS, to MonetDB, a column-oriented DBMS. The two DBMSs are evaluated in three realistic tasks in terms of time efficiency and resource (disk I/O and CPU) use. We find that, in these three tasks, MonetDB significantly outperforms the existing MS SQL database.

References

Abazajian, K. N., et al. 2009, ApJS, 182, 543
Cross, N. J. G., et al. 2012, A&A, 548, A119
Ivezic, Z., et al. 2008, ArXiv e-prints
Saito, R. K., et al. 2012, A&A, 537, A107



VISTA Milky Way interactive zoomable mosaic image credit: Mike Read (WFAU), UKIDSS/GPS, and VVV

Find me here: