ODBMS.ORG
Object Database Management Systems

OBJECTS IN SPACE.

The biggest data-processing challenge to date in astronomy: the Gaia mission.

I became aware of a very interesting project at the European Space Agency: the Gaia mission[1]. It is considered by the experts to be "the biggest data-processing challenge to date in astronomy."

I wanted to know more. I interviewed William O'Mullane, Science Operations Development Manager at the European Space Agency, and Vik Nagjee, Product Manager, Core Technologies at InterSystems Corporation, both deeply involved with the Proof-of-Concept of the data management part of this project.

I hope you enjoy this interview.

RVZ

by Roberto V. Zicari, Editor ODBMS.ORG, www.odbms.org

March 26, 2011
reprinted with permission of ODBMS.ORG

[1] http://sci.esa.int/science-e/www/area/index.cfm?fareaid=26

Q1. Space missions are long-term. Generally 15 to 20 years in length. The European Space Agency plans to launch a satellite called Gaia in 2012. What is Gaia supposed to do?

O'Mullane:
Well, we have now learned we will launch in early 2013 – delays are fairly commonplace in complex space missions. Gaia is ESA's ambitious space astrometry mission, the main objective of which is to astrometrically and spectrophotometrically map 1,000 million celestial objects (mostly in our galaxy) with unprecedented accuracy. The satellite will downlink close to 100 TB of raw telemetry data over five years.
To achieve its required accuracy of a few tens of microarcsecond astrometry, a highly involved processing of this data is required. The data processing is a pan-European effort undertaken by the Gaia Data Processing and Analysis Consortium. The result will be a phase space map of our galaxy, helping to untangle its evolution and formation.

Q2. In your report, "Charting the Galaxy with the Gaia Satellite[2]," you write "the Gaia mission is considered to be the biggest data-processing challenge to date in astronomy. Gaia is expected to observe around 1,000,000,000 celestial objects." What kind of information is Gaia expected to collect on celestial objects? And what kind of information does Gaia itself need in order to function properly?

O'Mullane:
Gaia has two telescopes and a Radial Velocity Spectrometer (RVS). From each telescope images are taken simultaneously to calculate position on the sky and magnitude. Two special sets of CCDs at the end of the focal plane record images in red and blue bands to provide photometry for every object. Then for a large number of objects spectrographic images are recorded. From the combination of this data we derive very accurate positions, distances, and motions of celestial objects. Additionally, we get metalicities and temperatures, etc. which allow the objects to be classified in different star groups. We use the term celestial objects since not all objects that Gaia observes are stars. Gaia will, for example, see many asteroids and improve their known orbits. Many planets will be detected (though not directly observed).
As with any satellite Gaia has a barrage of on-board instrumentation such as gyroscopes, star trackers, and thermometers which are read out and downlinked as "housekeeping" telemetry. All of this information is needed to track Gaia's state and position. Perhaps rather more unique for Gaia is

[2] http://www.odbms.org/downloads.aspx#odbms_ap

the use of the data taken through scientific instruments for self-calibration. Gaia is all about beating noise with statistics.

Q3. *All Gaia data processing software is written in Java. What are the main functionalities of the Gaia data processing software? What were the data requirements for this software?*

O'Mullane:
The functionality is to process all of the observed data and reduce it to a set of star catalogue entries which represent the observed stars. The data requirements vary – in the science operations center we downlink 50–80 GB per day, so there is a requirement on the daily processing software to be able to process this volume of data in eight hours. This is the strictest requirement because if we are swamped by the incoming datastream we may never recover.
The astrometric solution involves a few TB of data extracted from the complete set, the process runs less frequently (each six months to one year) so the requirement on that software is to do its job in four weeks.
There are many systems like this for photometry, spectroscopy, non-single stars, classification, and variability analysis, each of which have their own requirements on data volume and processing time.

Q4. *What are the main technical challenges with respect to data processing, manipulation, and storage this project poses?*

Nagjee:
The sheer volume of data that is expected to be captured by the Gaia satellite poses a technical challenge. For example, one billion celestial objects will be surveyed, and roughly 1,000 observations (100 x 10) will be captured for each object, totaling around 1,000 billion observations; each observation is represented as a discrete Java object and contains many properties that express various characteristics of these celestial bodies. The trick is to not only capture this information and ingest (insert) it into the database very quickly, but to also be able to do this as discrete (non-BLOB) objects so that downstream processing can be facilitated in an easy manner. Additionally, the system needs to be optimized for very fast inserts (writes) and also for very fast queries (reads). All of this needs to be done in a very economical and cost-effective fashion – reducing power, energy, and cooling requirements, etc., and also reducing costs as much as possible. These are some of the challenges that this project poses.

Q5. A specific part of the Gaia data processing software is the so-called Astrometric Global Iterative Solution (AGIS). This software is written in Java. What is the function of such a module? And what specific data requirements and technical challenges does it have?

Nagjee:
AGIS is a solution or program that iteratively turns the raw data into meaningful information.

O'Mullane:
AGIS takes a subset of the data, so-called well-behaved or primary stars, and basically fits the observations to the astrometric model. This involves refining the satellite attitude (using the known basic angle and the multiple simultaneous observations in each telescope) and calibration (given attitude and known star positions they should appear at definite points in the CCD at given times). It's a huge (too huge, in fact) matrix inversion but we do a block iterative estimation based on the conjugate gradient method. This requires looping (or iterating) over the observational data up to 40 times to converge the solution. So we need the I/O to be reasonable – we know the access pattern, though, so we can organize the data such that the readings are almost serial.

Q6. In your high-level architecture you use two databases, a so-called main database and an AGIS database. What is the rationale for this choice and what are the functionalities expected from the two databases?

O'Mullane:
Well, the main database will hold all data from Gaia and the products of processing. This will grow from a few TBs to few hundreds of TBs during the mission. It's a large repository of data. Now we could consider lots of tasks reading and updating this but the accounting would be a nightmare – and astronomers really like to know the provenance of their results. So we made the main database a little slower to update declaring a version as immutable. One version is then the input to all processing tasks, the outputs of which form the next version.
AGIS meanwhile (and other tasks) only require a subset of this data. Often the tasks are iterative and will need to scan the data frequently or access it in some particular way. Again, it is easier for the designer of each task to also design and optimize his data system than try to make one system work for all.

Q7. *You write that the AGIS database will contain data for roughly 500,000,000 sources (totaling 50,000,000,000 observations). This is roughly 100 terabytes of Java data objects. Can you please tell us what are the attributes of the Java objects and what do you plan to do with these 100 terabytes of data? Is scalability an important factor in your application? Do you have specific time requirements for handling the 100 terabytes of Java data objects?*

O'Mullane:
Last question first – four weeks is the target time for running AGIS, that's ~40 passes through the dataset. The 500 million sources plus observations for Gaia is a subset of the Gaia data which we estimate to be around 10 TB. Reduced to its quintessential element each object observation is a time, the time when the celestial image crosses the fiducial line on the CCD. For one transit there are ten such observations with an average of 80 transits per source. Carried with that is the small cut-out image over the five-year mission lifetime, from the CCD. In addition, there are various other pieces of information such as which telescope and residuals which are carried around for each observation. For each source we calculate the astrometric parameters which you may see as six numbers plus errors. These are: the position (alpha and delta), the distance or parallax (varPi), the transverse motion (muAlpha and muDelta), and the radial velocity (muRvs).
Then there is a magnitude estimation and various other parameters. We see no choice but to scale the application to achieve the run times we desire and it has always been designed as a distributed application.

Nagjee:
The AGIS Data Model comprises several objects and is defined in terms of Java interfaces. Specifically, AGIS treats each observation as a discrete AstroElementary object. As described in the paper, the AstroElementary object contains various properties (mostly of the IEEE long data type) and is roughly 600 bytes on disk. In addition, the AGIS database contains several supporting indexes which are built during the ingestion phase. These indexes assist with queries during AGIS processing, and also provide fast ad hoc reporting capabilities. Using InterSystems Caché, with its Caché eXTreme for Java capability, multiple AGIS Java programs will ingest the 100 terabytes of data generated by Gaia as 50,000,000,000 discrete AstroElementary objects within five days (yielding roughly 115,000 object inserts per second, sustained over five days).

Internally, we will spread the data across several database files within Caché using our Global and Subscript Mapping capabilities (you can read more about these capabilities here[3]) ,while providing seamless access to the data across all ranges. The spreading of the data across multiple database files is mainly done for manageability.

Q8. *You conducted a proof-of-concept with Caché for the AGIS database. What were the main technical challenges of such proof-of-concept and what are the main results you obtained? Why did you select Caché and not a relational database for the AGIS database?*

O'Mullane:
We have worked with Oracle for years and we can run AGIS on Derby. We have tested MySql and Postgress (though not with AGIS). To make the relation systems work fast enough we had to reduce our row count – this we did by effectively combining objects in blobs with the result that the RDBMs became more like a file system. Tests with Caché have shown we can achieve the read and write speeds we require without having to group data in blobs. This obviously is more flexible. It may have been possible to do this with another product but each time we had a problem InterSystems came (quickly) and showed us how to get around it or fixed it. For the recent test we requested the writing of specific representative datasets within 24 hours on our hardware – this was achieved in 12 hours. Caché is also a more cost-effective solution.

Nagjee:
The main technical challenge of such a Proof-of-Concept is to be able to generate realistic data and load on the system, and to tune and configure the system to be able to meet the strict insert requirements, while still optimizing sufficiently for downstream querying of the data.
The white paper „Charting the Galaxy with the Gaia Satellite[4]" discusses the results, but in summary we were able to ingest 5,000,000,000 Astro-Elementary objects (roughly 1/10th of the eventual projected amount) in around 12 hours. Our target was to ingest this data within 24 hours, and we were successful at being able to do this in half the time.
Caché is an extremely high-performance database, and as the proof-of-concept outlined in the white paper proves, Caché is more than capable of handling the stringent time requirements imposed by the Gaia project, even when run on relatively modest hardware.

[3] http://docs.intersystems.com/cache20102/csp/docbook/DocBook.UI.Page.cls?KEY=GSA_config
[4] http://www.odbms.org/downloads.aspx#odbms_ap

Q9. How do you handle data ingestion in the AGIS database, and how do you publish back updated objects into the main database?

Nagjee:
We use the Caché eXTreme for Java capability to interact between the Java AGIS application.

Q10. One main component of the proof-of-concept is the new Caché eXTreme for Java. Why is it important, and how did it get used in the proof-of-concept? How do you ensure low latency data storage and retrieval in the AGIS solution?

Nagjee:
Caché eXTreme for Java[5] is a new capability of the InterSystems Caché database that exposes Caché's enterprise and high-performance features to Java via the JNI (Java Native Interface). It enables "in-process" communication between Java and Caché, thereby providing extremely low-latency data storage and retrieval.

Q11. What are the next steps planned for the AGIS project and the main technical challenges ahead?

Nagjee:
The next series of testing will focus on ingesting even more data sources – up to 50% of the total projected objects. Next, we'll work on tuning the application and system for reads (queries), and will also continue to explore additional deployment options for the read/query phase (for example, ESA and InterSystems are looking at deploying hundreds of AGIS nodes in the Amazon EC2 cloud so as to reduce the amount of hardware that ESA has to purchase).

O'Mullane:
Well, on the technical side we need to move AGIS definitively to Caché for production. There are always communication bottlenecks to be investigated which limit scalability. AGIS itself requires several further developments such as a more robust outlier scheme and a more complete set of calibration equations. AGIS is in a good state but needs more work to deal with the mission data.

[5] http://www.intersystems.com/java/

William O’Mullane, Science Operations Development Manager, European Space Agency

William O’Mullane has a background in Computer Science and has worked on space science projects since 1996 when he assisted with the production of the Hipparcos CDROMS. During this period he was also involved with the Planck and Integral science ground segments as well as contemplating the Gaia data processing problem. From 2000–2005 he worked on developing the US National Virtual Observatory (NVO) and on the Sloan Digital Sky Survey (SDSS) in Baltimore, USA. In August 2005 he rejoined the European Space Agency as Gaia Science Operations Development Manager to lead the ESAC development effort for the Gaia Data Processing and Analysis Consortium.

Vik Nagjee, Product Manager, Core Technologies, InterSystems Corporation

Vik Nagjee is the Product Manager for Core Technologies in the Systems Development group at InterSystems. He is responsible for the areas of Reliability, Availability, Scalability, and Performance for InterSystems’ core products – Caché and Ensemble. Prior to joining InterSystems in 2008, Nagjee held several positions, including Security Architect, Development Lead, and head of the performance and scalability group, at Epic Systems Corporation, a leading health care application vendor in the US.

About the Author
Roberto V. Zicari

Prof. Roberto V. Zicari is editor of ODBMS.ORG (www.odbms.org), the most up-to-date collection of free materials on object database technology on the Internet. The portal was created to serve software developers in the open source community or at commercial companies as well as faculty and students at educational and research institutions. ODBMS.ORG is designed to meet the fast-growing need for resources focusing on Service platforms, Scalable Cloud platforms, Analytical Data Platforms, Object databases, Object-relational bindings, NoSQL databases and new approaches to concurrency control. Roberto is Full Professor of Database and Information Systems at Frankfurt University and representative of the OMG in Europe. Previously, Roberto served as associate professor at Politecnico di Milano, Italy; Visiting scientist at IBM Almaden Research Center, USA, and the University of California at Berkeley, USA; Visiting professor at EPFL in Lausanne, Switzerland, at the National University of Mexico City, Mexico and the Copenhagen Business School, Danemark.