# A Survey of Scientific Applications using SciDB

Paul G. Brown
Paradigm4 Inc
281 Winter Street Suite 360
Waltham MA 02451 USA
pbrown@paradigm4.com

## ABSTRACT

While SciDB is gaining popularity among commercial software developers and data analysts charged with building "Big Data" applications, the platform was conceived as a tool for scientists. Here, we review a number of purely scientific applications of SciDB; bioinformatics at the NIH, experimental physics at NERSC, and remote sensing at INPE. We describe each project in terms of its scientific objectives, the quantity and character of the data involved, the analytic workload, and its hardware architecture. We conclude the talk with a general summary of what these projects have in common, and comment on what experience building these applications implies for data management in the Internet-of-Things (IoT); a computing environment where all interesting data is generated by *machines*.

## 1. INTRODUCTION

The SciDB project took as its design goals a list of features identified as being critical to scientific data management in a survey of working scientists [7]. Some of these features-such as scalable, transactional data storage and high level (ie. query centric) interfaces-were familiar to DBMS developers and researchers. But other items on the list-an emphasis on *arrays* as the data model building blocks and built-in provenance maintenance-were not. And of course, everyone expected the highest standards of software quality and reliability!

Since its release in 2011, SciDB [4] has attracted the attention of users with both commercial and scientific motives. This presentation surveys some of the purely scientific research projects that use SciDB. We report on applications from three distinct scientific disciplines; each selected because it is characteristic of a larger number of use-cases within its scientific domain. We describe each research project in terms of its scientific goals, the nature of the data it is dealing with, its analytic query workload, and hardware environment. Our goal is to provide some insight into how a variety of scientific applications use SciDB.

We conclude by generalizing a little about what the SciDB team has learnt from our work with science data management. Something all these applications have in common is that their data is almost exclusively *machine generated*; a characteristic modern scientific data shares with data generated by the Internet of Things (IoT). Consequently, we anticipate that the techniques and methodologies our scientific users employ are an important research topic because of what they can teach us about the analysis of IoT data.

## 2. BIOINFORMATICS

SciDB's first large deployment was as the DBMS backing the National Center for Biotechnology Information's (NCBI) 1000 Genomes browser [6]. Conceptually, genotype data can be organized as a sparse, two-dimensional array where each *row* corresponds to a potential genetic variation, and each *column* represents one person's DNA sample. Scientists who study the human genome can measure about 39.6 million variants (*rows*), while the number of DNA samples (*column*) is bound by the time and cost of sequencing. Currently there are several thousand DNA samples in the public database. More are added daily.

The 1000 Genomes public workload is dominated by `slice` and `between` operations; queries that extract sub-arrays from the data. For example, people using the repository might like to know all of the DNA samples that share a particular variant (that is, all *columns* for a given *row*), or all variants for a given DNA sample (all *rows* for a given *column*). In addition the database provides aggregated information, such as the number of times a particular variant (or combination of variants) was observed. In detail, the SciDB schema consists of a small number of arrays which share dimensions, thus facilitating complex combinations of `filter` and `cross_join` queries to drill into the data set.

As of mid-2014, NCBI's developers run SciDB on clusters of four nodes: each with 64GB memory, dual quad-core Xeon CPUs, and 10TB of Raid 10 storage. To achieve the site's high availability goals NCBI's developers run several redundant SciDB installations. Each instance of the public dataset contains about 2TB and serves about 3000 analytic user sessions per day.

To facilitate the use of SciDB as a platform for genomic variant analysis, the SciDB team created an Amazon EC2 AMI with a considerable amount of variant data pre-loaded. With the broad scientific objectives of the 1000 Genomes project in mind, and to demonstrate SciDB's statistical abilities, we applied the Principle Components Analysis technique (using the `svd` operator) to 2500 samples and clustered the DNA samples into three sub-groups roughly reflecting each sample donor's ethnicity.

## 3. EXPERIMENTAL PHYSICS

The National Energy Research Scientific Computing Center (NERSC) makes SciDB available as a test-bed computational system. Multiple teams have built applications using SciDB in a variety of scientific problem domains, including the team responsble for analyzing data generated by the Large Underground Xenon (LUX) detector [1]. The pur-

pose of the LUX detector is to gather evidence about the interaction between "dark matter" and ordinary matter on Earth. Specifically, this involves finding rare instances-only 160 of which occurred during the initial 85 detecting days-of protons behaving oddly in tank of 370 kg of liquid xenon immersed in another tank containing 70,000 gallons of water, all buried about a mile underground.

The SciDB prototype database for LUX [3] consists of a single, 3D array; having pulse type, pulse number and timestamp dimensions, and about 50 data attributes per cell. Each array cell corresponds to an energy pulse detected within the liquid xenon. The analytic goal is to sift this data to figure out when a small initial pulse-with certain properties-is followed within one milli-second by larger pulse-again, with certain properties.

It's worth noting that one of the reasons the LUX team turned to SciDB and NERSC was that they operate under considerable resource constraints. Where the Large Hadron Collider is blessed with about 3000 active investigators-many able and willing to write code-LUX has only about 100. Consequently, the LUX team saw the key impediment to making progress on their analysis as software *write-time*-measured in weeks-rather than software *run-time*–measured in hours.

LUX's analytic queries are relatively complex. They involve first calibrating pulse sizes to a distribution calculated using the `regrid` operator. These normalized pulses are `filter`ed to focus on pulses with specific characteristics. Then each pair of pulses needs to be checked using `cross_join` to find small followed by large pulse pairs. As things turned out, the LUX team were able to perform this entire analysis on 32TB of data-100 days of continuous detection-consisting of 600,000,000 pulses in about four hours using a SciDB installation of 32 instances running on 16 physical nodes.

# 4. IMAGE DATA ANALYSIS

Scientists who work with remote sensing data were well represented among those polled as part of the SciDB design process, so it should come as no surprise that the platform has been embraced by that community. In this talk we focus on work done by geoinformatics researchers at the Brazilian National Institute for Space Research (INPE). One of the early SciDB projects at INPE involved an attempt to reproduce (and thereby validate) an important but controversial result published by a different team in *Science* in 2007 [5]. This earlier work counter-intuitively showed that Brazil's rainforests got *greener* during drought [2].

Loading the source data proved to be a significant challenge. The INPE team used MODIS HDF5-formatted images-which contain data from both visible and infrared bands-covering Brazil for the time period 2000 to 2012 at 8 days temporal resolution. They organized this large data set into a single, 3D array with latitude, longitude and time dimensions. Each of the 11,968 image consisted of 4,800 x 4,800 pixels, so in total, this resulted in an array of 275 billion cells, each cell storing 3 values in double precision, for a total of about 7 TB.

The advantages of SciDB's high level query language then became clear when the considerable amount of custom C++ and Python written to perform the original analysis was replaced by a single, seven line SciDB query. SciDB took 4.6 hours to reproduce the result in the *Science* paper on a single, 24 core / 128 G memory physical node, running five instances of SciDB.

# 5. SCIENTIFIC DATA METHODS AND THE INTERNET OF THINGS

These diverse projects all share one important characteristic. In contrast with applications that are the bread and butter of SQL RDBMS platforms–management information systems for business data processing–or even many of the so-called "Big Data" applications for which Hadoop has become a popular technology choice–social networks, click-stream logs, and large text corpuses-almost every byte of data in these applications is *machine generated*. That is, each of these application acquired its data through some form of sensor technology, and organized it into a multi-dimensional reference framework where *co-locality*–adjacency in space or time, or relative position in a sequence–is essential to the data's *semantics*. SciDB's ability to cluster co-located data in a scalable fashion is essential to efficiently executing the queries in these applications' workloads.

Why this emphasis on machine generated data? Because it offers us a glimpse into what data management for the Internet of Things (IoT) might look like. The most straightforward vision of the IoT is a vast, ubiquitous collection of sensors distributed in space, all collecting timeseries data. Provisioning out such infrastructure will be enormously expensive, and such costs can only be justified (on economic grounds) if investors foresee some satisfactory return. Developers who have built the applications we have examined here have all operated under resource constraints that limited their ability to write lots of code, but their analytic methods–large scale clustering, rare event counting, sensor time series–are very sophisticated.

If the IoT is to become a reality, then platforms like SciDB which are ideal to flexibly support a wide range of "scientific" data analysis, will be necessary.

# 6. REFERENCES


[1] D. Akerib et al. The Large Underground Xenon (LUX) Experiment. *Nucl.Instrum.Meth.*, A704:111–126, 2013.

[2] G. Camara, M. J. Egenhofer, K. Ferreira, P. Andrade, G. Queiroz, A. Sanchez, J. Jones, and L. Vinhas. Fields as a Generic Data Type For Big Spatial Data, 2014.

[3] L. Gerhardt, J. Kepner, M. Matz, A. Poliakov, and Y. Yao. SciDB - Manage and Analyze Terabytes of Array Data. In *The International Conference for High Performance Computing, Networking, Storage and Analysis*, 2014.

[4] Paradigm4 Inc. *SciDB Reference Manual*. Paradigm4 Inc, 281 Winter Street Suite 360, Waltham, MA 02451, 14.12 edition, Jan. 2015.

[5] S. R. Saleska, K. Didan, A. R. Huete, and H. R. da Rocha. Amazon forests green-up during 2005 drought. *Science*, 318(5850):612, 2007.

[6] D. Slotta. NCBI Genotype Archive. In *25th International Conference on Scientific and Statistical Database Management*, 2013.

[7] M. Stonebraker, J. Becla, D. J. DeWitt, K. T. Lim, D. Maier, O. Ratzesberger, and S. B. Zdonik. Requirements for science data bases and scidb. In *CIDR 2009, Fourth Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 4-7, 2009, Online Proceedings*, 2009.