

ATLAS: Big Data in a Small Package?

Larry Denneau, Jr.

Institute for Astronomy, University of Hawaii,
2680 Woodlawn Dr., Honolulu, HI USA
email: denneau@hawaii.edu

Abstract. For even small astronomy projects, the petabyte scale is now upon us. The Asteroid Terrestrial-impact Last Alert System (Tonry 2011) will survey the entire visible sky from Hawaii multiple times per night to search for near-Earth asteroids on impact trajectories. While the ATLAS optical system is modest by modern astronomical standards – two 0.5 m F/2.0 telescopes – each night the ATLAS system will measure nearly 10^9 astronomical sources to a photometric accuracy of $< 5\%$, totaling 10^{12} individual observations over its initial 3-year mission. This ever-growing dataset must be searched in real-time for moving objects and transients then archived for further analysis, and alerts for newly discovered near-Earth asteroids (NEAs) disseminated within tens of minutes from detection. ATLAS's all-sky coverage ensures it will discover many 'rifle shot' near-misses moving rapidly on the sky as they shoot past the Earth, so the system will need software to automatically detect highly-trailed sources and discriminate them from the thousands of low-Earth orbit (LEO) and geosynchronous orbit (GEO) satellites ATLAS will see each night. Additional interrogation will identify interesting phenomena from millions of transient sources per night beyond the solar system. The data processing and storage requirements for ATLAS demand a 'big data' approach typical of commercial internet enterprises. We describe our experience in deploying a nimble, scalable and reliable data processing infrastructure, and suggest ATLAS as steppingstone to data processing capability needed as we enter the era of LSST.

Keywords. methods: data analysis, surveys, minor planets, asteroids, astronomical data bases: miscellaneous

1. Introduction

Fueled by the Moore's Law (Moore 1965), the relentlessly increasing density of transistors and resulting cheaper and faster electronic components have ushered in a new age in astronomical data acquisition and computation. CCDs for astronomical cameras now number hundreds of millions of pixels per single device, with mosaics achieving billions of pixels. Increasingly sophisticated data processing techniques are able to rapidly measure the sources in these single images and perform deeper searches across years' worth of images. Storage and interrogation of modern astronomical images requires entire 'farms' of compute and disk, typically occupying many cabinets of servers in a dedicated compute facility. As compute facilities for astronomical projects continue to grow, greater engineering is required to deploy infrastructure, and maintenance and reliability become serious issues – more hardware means more components can fail, so redundancy and high availability (resistance to outages) must be designed into the infrastructure. A fundamental conflict exists between the low cost of computer resources and the growing difficulty of managing these resources.

In the commercial world, a similar technological revolution has been spurred by the availability of the same faster, cheaper technology applied to the Internet. Billions of

devices, from traditional desktop and laptop computers to mobile phones and appliances, interact via megabit-capacity connections to an Internet supported by many multi-gigabit backbones. Thousands of web services exist that each can support realtime communications and data exchange among millions of users. Computational resources themselves can be provisioned in the 'cloud', meaning that procurement of hardware and physical proximity are no longer a requirement for deploying large-scale compute infrastructure. The software landscape is exploding with free and commercial packages to archive and mine the huge volume of data generated daily by global Internet users. This scale of computational infrastructure, now referred to as 'big data,' was once considered the domain of large astronomy or particle physics projects, but has become routine for a commercial Internet endeavor.

The Asteroid Terrestrial Last-Alert System (Tonry 2011) is a contemporary survey telescope project whose success will depend on the project's ability to grapple with big data scale while remaining a nimble and lean project that can stay within its predicted budget. ATLAS relies on modern technology to achieve scalable, unprecedented performance per unit cost. In the remainder of this paper, we present ATLAS requirements and discuss technological choices available to ATLAS and their ramifications.

2. ATLAS

ATLAS was funded in 2013 by the NASA Near Earth Objects Observations (NEOO) program for \$5M over five years to design, construct and operate for two years a telescope system optimized for the discovery of 'death-plunge' asteroids on impacting trajectories with the Earth. While other near-Earth asteroid (NEA) surveys exist and are funded by NASA NEOO, with Catalina Sky Survey (CSS; Larsen *et al.* 1998) and Pan-STARRS (Kaiser *et al.* 2002) being the dominant NEA discovery surveys, these programs have historically focused on the threat posed by the largest NEAs, typically > 140 m in diameter. This focus requires survey strategies geared toward discovering these objects at opposition when they are brightest and moving slowly. This strategy has been largely successful, having achieved $> 90\%$ completion of 1 km NEAs (Mainzer *et al.* 2011) and increasing the number of known NEAs of any size to more than 10,000.

Despite their orientation toward large objects, these surveys still discover large numbers of small NEAs. Catalina Sky Survey remains the only survey to date to have discovered near-Earth asteroids prior to impact: 2008 TC3 and 2014 AA, both about 2-4 m in diameter (Boattini *et al.* 2009, Kowalski *et al.* 2014).

ATLAS optimizes for small, impacting NEAs via its 30 deg^2 field of view that is capable of observing the entire visible sky three times per night down a limiting magnitude of $m = 20$. Despite its relatively large field of view compared to other NEA survey telescopes, each ATLAS telescope's STA1600 110 megapixel CCD achieves a pixel scale of 1.8 arcseconds. The ATLAS system will obtain around 2000 exposures per night, for a raw pixel volume of about half a terabyte (10^{12} bytes) per night for the duration of a nominal two-year survey, nearly a half petabyte (10^{15} bytes). Exposures obtained under photometric conditions will be assembled into a full-sky template image for image subtraction. With its nightly all-sky coverage, ATLAS can assemble a deep full-sky template in several nights.

Each night ATLAS will also measure between 10^8 and 10^9 astronomical sources (mostly stars and galaxies); by acquiring three measurements at a given sky location per night, ATLAS will obtain nearly 1000 distinct measurements of these 10^9 sources by the end of its nominal two-year operations funding. The ATLAS software is mostly home-grown, consisting of fast, compact code to perform basic reductions (flatfielding, calibration), image subtraction and source detection.

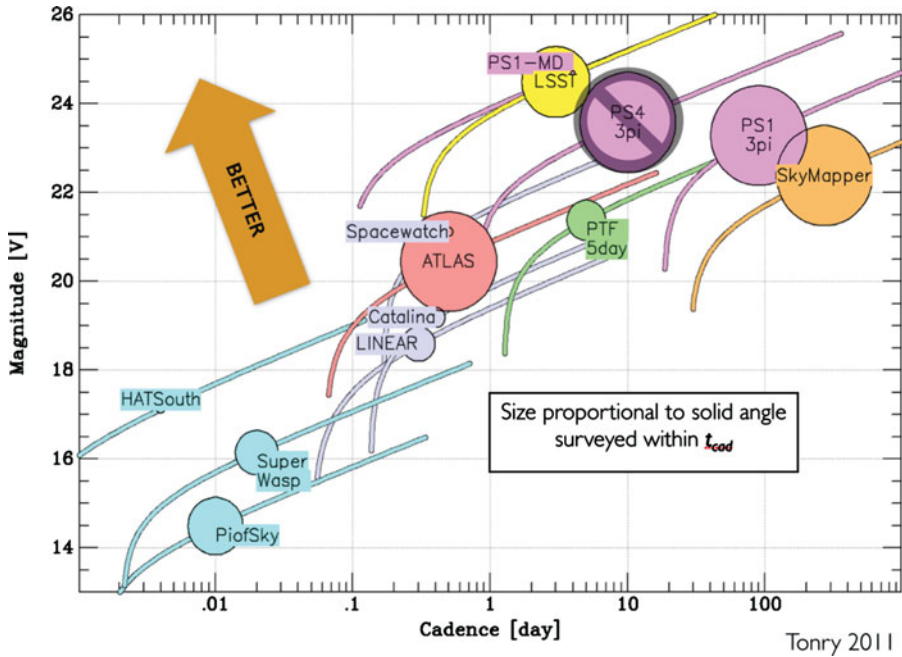


Figure 1. Comparison of ATLAS performance to contemporary survey telescopes at detecting transient phenomena. Better asteroid discovery performance (in particular for NEAs) occurs toward the top left, where single-image sensitivity and short cadences are important. The circles are proportional to the solid angle surveyed for a given cadence interval. The short tails to the bottom left of each indicate the sensitivity at a given cadence; at longer time scales, sensitivity is increased due to image stacking at the expense of the ability to detect asteroids, which require short time scales.

To find moving objects in its data, ATLAS will employ an adapted version of the Pan-STARRS moving object processing system (Denneau *et al.* 2013). Effective identification of transient sources requires ATLAS to catalog all sources detected in its images. Stars and galaxies must be recorded for photometric and astrometric calibration and for identification with transient sources that are known to be variable stars (of which ATLAS will detect millions per night). So in addition to raw pixel data, the ATLAS database must absorb information for $> 10^8$ sources per night. Asteroids will constitute a small fraction of the transient sources measured by ATLAS, around 10^5 per night.

The ATLAS hardware is modest by modern standards – a 0.5 m primary mirror F/2.0 in a Wright-Schmidt configuration, with a construction cost of about \$1M per telescope, yet it will measure nearly 1/3 the number of sources measured by the Pan-STARRS1 survey at nearly two orders of magnitude more in cost. The etendue, or product $A\Omega$ of collecting area times solid angle subtended, is similar to Pan-STARRS or Skymapper, making up the 1/10 sensitivity with $10\times$ the solid angle. Figure 1 compares ATLAS to contemporary surveys, revealing ATLAS to be competitive with other modern surveys with respect to discovery performance for transient phenomena, especially once total project cost is considered.

Figure 2 shows the ATLAS sensitivity for 30 m NEAs – ATLAS will be sensitive to asteroids in a ‘candle flame’ shape around the Earth, able to detect incoming asteroids in all directions except toward the Sun. There is an enhancement toward opposition where phase effects allow asteroids to be detected more easily. The candle flame suggests that

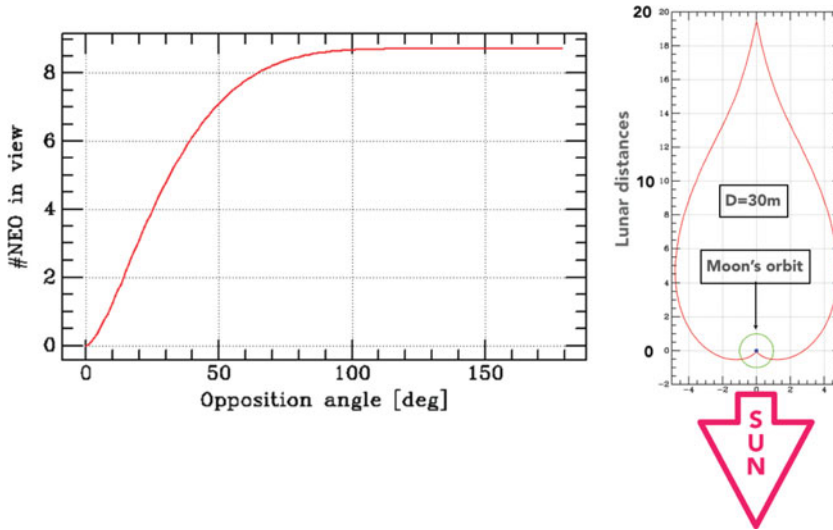


Figure 2. Visibility of 30m NEAs by ATLAS. The ‘candle flame’ shape on the right shows the volume for which ATLAS is sensitive to NEAs of a given size.

ATLAS will provide, on average, seven days warning for an impacting 50 m asteroid and three weeks for a 140 m asteroid, allowing for civil defense of at-risk areas.

One additional feature of the ATLAS project is its small size in human terms: the project employs only five full-time employees (FTEs) who have been responsible for management of the project, design of the ATLAS camera, and development of ATLAS software. ATLAS operates in many ways like a startup company – the small staff is responsible for project management and engineering.

3. Big Data

The ATLAS data stream presents several challenges that encroach upon the ‘big data’ regime of data management and processing. Broadly, big data refers to volume or complexity of data that requires unconventional methods for acquisition, storage, analysis, search, visualization and sharing. For modern Internet services, these issues might refer to real-time ingest and classification of geospatially-encoded end-user communications; storage and indexing of this data in a distributed, redundant database; and real-time analytics of this data and the on-demand dispatch of search results from such a database.

In the ATLAS context, big data approaches are needed for a) storage and retrieval of 0.5 TB of pixel data every night; b) database storage and analysis of 10^8 astronomical sources per night; c) real-time spatial searching of 10^6 transient detections per night for new and unknown asteroids; and d) image processing to integrate photometric images into the deep full-sky template.

Table 1 describes the processing rates for several large-scale Internet services. Some of these services have become so prevalent that they have cultivated entire software ecosystems that support and extend their data services, for example:

- External curation of video (*e.g.* YouTube) into specialty channels
- Games and discussion groups using a service’s application programming interface (API)
- Realtime-discussion groups and news feeds generated from billions of messages daily

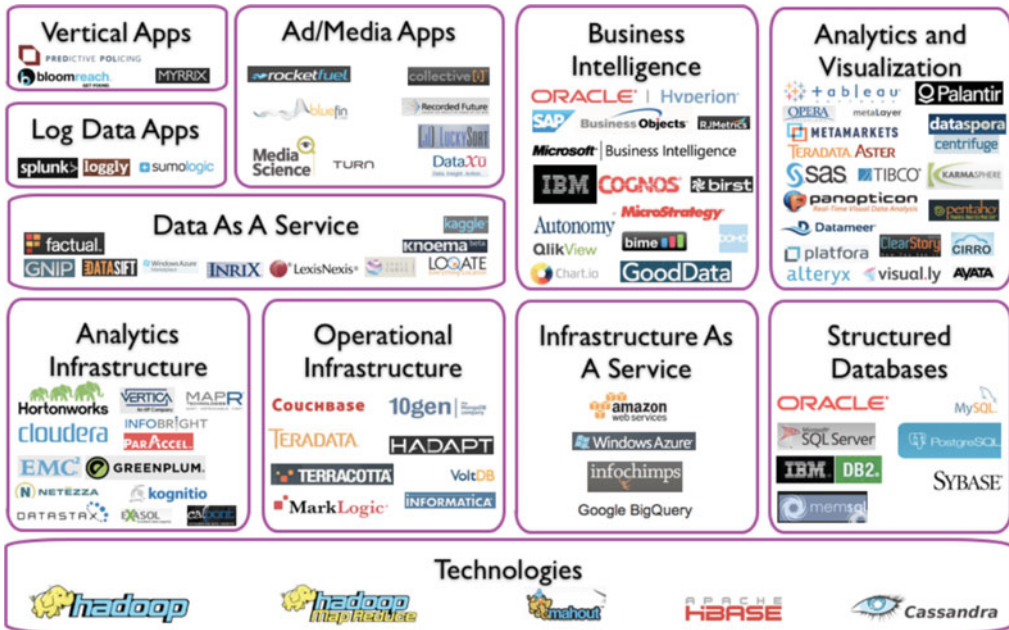


Figure 3. The big data landscape as of 2012. (Feinleib 2012)

An important by-product of these software ecosystems is the proliferation and availability of general-purpose open-source software to build and manage infrastructure in support of these services. The image- and video-based services in particular have led to a multitude of frameworks to enable distributed multiprocessing over large datasets. Inspired by Google, Inc.'s MapReduce programming model for parallel, distributed programming (Dean & Ghemawat 2008), the Apache Software Foundation developed the Java-based Hadoop framework (White 2009), which in turn has spun off many supporting projects: HBase, a scalable, distributed Hadoop database; Hadoop Distributed File System (HDFS), for high-throughput access to application data; Cassandra, a scalable multiple-master database; Spark, a compute engine for Hadoop data, to name a few. As of September 2015, the Twitter service operates a 300PB Hadoop infrastructure federated over tens of thousands of servers (Shegalov 2015).

The unstructured document-based nature of Internet services has also led to rapid growth of 'no-SQL' databases, a departure from traditional sequential query language (SQL) databases that enforce rigid structure of input data into tables. As with distributed parallel processing, there now exists a large number of open-source packages to serve huge volumes of unstructured data in such databases: MongoDB, CouchDB, DynamoDB, MapR are among the dozens available. Figure 3 conveys the large array of choices in the big data software marketplace.

All of these packages perform admirably and support successful commercial enterprises. They are relevant to ATLAS in that they *could* facilitate real-time processing of geospatially-encoded (*e.g.* celestial) pixel and catalog data and can grow to petabyte scales. Since the software is open-source and freely available, these packages would seem to be an excellent fit to the ATLAS project.

Table 1. Daily transaction rates, per *minute*, of modern ‘big data’ Internet services. Most of these services are capable of performing real-time, geo-indexed searches against databases of millions of users and billions of messages.

Service	Action	Data	Number/day
Twitter	Send	Tweets (text messages)	5×10^8
Netflix	Stream	Hour of video	1×10^8
Skype	Send/receive	Audio calls	1.5×10^8
Snapchat	Share	Photo	4×10^8
Instagram	Share	Photo	2.5×10^9
Facebook	Like	Post	5×10^9
Vine	Play	Video	1.4×10^9
YouTube	Upload	Hour of video	4×10^5
Pinterest	Pin (upload)	Image	1.3×10^7

4. Discussion

Mesmerized by the prospects of open-source distributed parallel processing software used by successful companies, ATLAS undertook investigation of modern big data software within its infrastructure. Constrained by a limited staff, budget and schedule, the results have been underwhelming and largely unsuccessful for the following reasons:

Big hardware deployments. Environments such as Hadoop and Cassandra require significant hardware just to deploy for accurate evaluation under test scenarios. Simultaneous comparison of competing technology requires even more hardware. We found the procurement and setup of evaluation systems to be slow and time-consuming. Cloud services can alleviate some of these issues at much greater expense, exceeding the cost of hardware after several months of operation.

Big software stacks. Modern big data infrastructure requires complicated supporting software environments to be installed underneath them, spanning an array of programming languages (*e.g.* Go, Java, Python, JavaScript, Scala) and high-level operating environments (Linux, Docker). Proper exploitation of these environments depends on highly-skilled staff fluent in all of these tools.

Big engineering. Installation of many big data infrastructures continues to be simplified, but adaptation to ATLAS-specific needs requires significant integration engineering.

Uncertain software lifespan. The rapidly-changing commercial software world in which these tools originate means that a tool that was ‘industry standard’ several years ago may be superseded by new technology. When a project such as ATLAS needs stability over a 5-10 year window, the correct choice of tools (if there is one) is not at all obvious.

CAP theorem considerations. A properly deployed big data infrastructure must cope with theoretical limitations as described by (Brewer 2000), now known as Brewer’s CAP Theorem (Gilbert & Lynch 2002). CAP refers to a) consistency, all nodes see the same data at the same time; b) availability, a guarantee that requests receive a response indicating success or failure; and c) partition tolerance, or ability to perform during network failures. The CAP theorem broadly states that two of these three features are available in a distributed system. ATLAS operations need to be evaluated against the tradeoffs imposed by CAP limitations.

Going forward, ATLAS has retreated somewhat to a traditional relational database management system (RDBMS) with a pool of locally-managed storage nodes, growing on-demand as the ATLAS dataset grows. But compute and storage hardware that a decade ago filled a room occupies a single enclosure today. Our take-away lessons from the ATLAS big data experiment:

Big data is hard – how do you find the right model? Optimal use of big data technology requires intense evaluation of hard-to-deploy scenarios. There are dozens of competing infrastructures for distributed parallel data processing, and some of them will not exist several years from now. What expertise is needed and what are the proper evaluation techniques? Strong engineering is needed to answer these questions.

Boring technology is OK. Old technology has not stood still – ‘boring’ RDBMSes such as MySQL have grown in capability and can now handle the 10^{12} records that ATLAS will obtain and can operate in replicated and/or distributed modes. Traditional parallel processing environments such as HTCondor (Thain *et al.* 2005) remain under development and are able to scale over thousands of CPUs on standard Linux systems.

Don't overreach. It is better to employ well-understood technology that strains to keep up with ATLAS data processing needs than to adopt new technology whose engineering demands and long-term performance are unknown and can possibly overwhelm information technology staff.

Acknowledgements

The design and construction of ATLAS is funded by the National Aeronautics and Space Administration Near Earth Objects Observations program under Grant No. NNX12AR55G.

References

- Busso, A. *et al.* (2009) 2009, *AAS Abstracts*, 41
- Brewer, E. (2000) 2000, *PODC Keynote*
- Dean, J. & Ghemawat, S. (2008) 2008, *Communications of the ACM*, 51, 1, 107
- Denneau, L. *et al.* (2013) 2013, *PASP*, 125, 357
- DOMO (2015) 2015, <https://www.domo.com/blog/2015/08/data-never-sleeps-3-0>, 2015
- Feinleib, D. (2012) 2012, <http://www.forbes.com/sites/davefeinleib/2012/06/19/the-big-data-landscape/>
- Gilbert, S. & Lynch, N. (2002) 2002, *ACM SIGACT News*, 33, 2, 51
- Kaiser, N. *et al.* (2002) 2002, *SPIE Conference Series*, 4836, 154
- Kowalski, R. A. *et al.* (2014) 2014, <http://www.minorplanetcenter.net/mpec/K14/K14A02.html>
- Larson, S., Brownle, J., Hergenrother, C., & Spahr, T. (1998) 1998, *Bulletin of the American Astronomical Society*, 30, 1037
- Mainzer, A. K. *et al.* (2011) 2011, *ApJ*, 743, 156
- Moore, G. (1965) 1965, *Electronics Magazine*, 38, 8
- Shegalov, G. (2015) 2015 <https://blog.twitter.com/2015/hadoop-filesystem-at-twitter>, 2015
- Thain, D., Tannenbaum, T., & Livny, M. (2005) 2005, *Concurrency - Practice and Experience*, 17, 323
- Tonry, J. L. (2011) 2011, *PASP*, 123, 58
- White, T. (2009) 2009, *Hadoop: The Definitive Guide* (O'Reilly Media, Inc.)