COMPUTING IN ASTRONOMY: TO SEE THE UNSEEN

Victor Pankratius, *MIT, Haystack Observatory*
Chris Mattmann, *NASA Jet Propulsion Laboratory and University of Southern California*

Advances in computing have empowered astronomers to explore the universe in greater detail. Software-defined instruments relying on digital data capture and processing are more powerful than ever and continue to bring us new knowledge about the universe and our place in it.

The exciting evolution from an era of scarce scientific data to an era of overabundant data is particularly evident in astronomy. Instruments and sensors are increasingly capable of generating data on the order of petabytes per second. This data comes from a multitude of ground-based instruments, such as the Low Frequency Array (LOFAR) for radio astronomy, the Jansky Very Large Array, and the Atac-

0018-9162/14/$31.00

ama Large Millimeter/submillimeter Array (ALMA), as well as space-based instruments such as Kepler, Hubble, the James Webb Space Telescope (JWST), and the Chandra X-ray Observatory. The data these special instruments capture helps us understand our universe's origin and development, as well as search for habitable planets.

Recently, computer science has begun to play a major role in enabling astronomy's scientific progress. Specifically, hardware and software advances, as well as the economies of scale leading to more affordable computers, have resulted in new capabilities in data collection, analysis, and visualization.

For computer scientists, astronomy is an interesting testbed because of its extreme requirements in massive data handling, computational speed, numerical precision, and networking.

Consider ALMA. One of the most powerful radio telescopes ever built, it is located in Chile 16,500 feet above sea level and connects 66 different dish antennas. Using computational interferometry techniques, a parallel computer now cross-correlates signals to make all these antennas appear as one virtual dish. The very long baseline interferometry (VLBI) technique can scale even further to harness antennas all over the world in such a way that the diameter of the virtual telescope becomes Earth's diameter. Signals become bits and bytes that have to be correlated with atomic-clock precision; and computationally intensive signal-processing algorithms have to separate signals from noise and then compose a result that is useful for astronomers.

It would be impossible to build such instruments without computers. ALMA software development is a global endeavor with collaborators across multiple continents—from Socorro, New Mexico, to Japan and Chile. Future projects and associated instruments will bring additional advances and capabilities to the astronomy field: the Square Kilometre Array (SKA) in South Africa and Australia will be the world's largest radio telescope and produce data on the order of 700 Tbytes per second; the Large Synoptic Survey Telescope (LSST) in Chile will regularly photograph the entire sky and trace billions of galaxies. These advances will require a stronger emphasis of computer science in astronomy.

Facing this data deluge, scientists must rely on computing solutions to complement and support discovery processes that used to be exclusively human. For computer scientists, astronomy is an interesting testbed because of its extreme requirements in massive data handling, computational speed, numerical precision, and networking.

IN THIS ISSUE

Through astronomy, excellent research and teaching opportunities present themselves in sensor networks; parallel, high-performance, and cloud computing; distributed systems; software engineering; open source development; programming languages; databases; algorithms; and many more. The cover features in this special issue reflect the topical diversity of computing in astronomy.

In the first article, "Studying the Milky Way Galaxy Using ParaHeap-*k*,'' Mark Jenne, Owen Boberg, Hasan Kurban, and Mehmet Dalkilic offer a glimpse into the algorithms involved in astronomical computing. Presenting a parallel version of *k*-means clustering, they aim to recover major components of our galaxy—such as the halo, the thick disk, and the thin disk—in a simulation with nearly a million stars. Study results are a starting point for data mining in the large datasets that future instruments will produce. In addition, the article's tutorial emphasis helps introduce some basic astronomical vocabulary to computer scientists.

In "High-Performance Computing of Self-Gravity for Small Solar System Bodies,'' Daniel Frascarelli, Sergio Nesmachnow, and Gonzalo Tancredi revisit the classic *n*-body problem in the context of fine-grained multithreading. They examine small solar system bodies that agglomerate smaller grain-like structures subject to collisions, elastic interactions, and frictional interactions. The authors also evaluate parallelization on multicore machines and explain the basics of parallel programming.

In "Scaling Astroinformatics with Pydron: Python + Automatic Parallelization," Stefan Müller, Gustavo Alonso, and André Csillaghy introduce Pydron—a set of Python extensions for exploratory data analysis in astronomy. Pydron can be used to automatically parallelize select parts of the code based on a dataflow graph, provided the programmers use certain decorators. Decorator annotations furthermore simplify the deployment and parallel code execution in cloud environments.

In the fourth article, "An End-to-End Computing Model for the Square Kilometre Array," Rik Jongerius, Stefan Wijnholds, Ronald Nijboer, and Henk Corporaal discuss the future SKA instrument. When finished, it will be the largest radio telescope ever built. In the first construction phase, 250,000 dual dipole antennas and 350 dishes will be deployed. This article's quantitative discussion helps give a thorough understanding of requirements for SKA's data acquisition and processing pipeline.

In "Adventures in Antarctic Computing, or How I Learned to Stop Worrying and Love Neutrino," Lisa Gerhardt, Juan Carlos Díaz Vélez, and Spencer R. Klein describe the IceCube neutrino telescope, which consists of 5,160 optical sensors buried a mile deep under the South Pole's ice. The article explains why complex instrumental approaches like this are necessary and how such constrained environments affect the computation.

The last article presents a collection of briefly described astronomy-related projects and libraries—the existence of which are likely to be of wide general interest.

EXPANDING ASTROINFORMATICS: THE ROAD AHEAD

The challenges described above clearly show the need for additional interdisciplinary research between computer science and astronomy. We should not make the mistake of looking at computing in astronomy as simply an applied form of computer science. That would imply the existence of a toolbox containing suitable techniques that can easily be adapted to work in astronomy. Experience shows, however, that these tools either do not match our needs or have yet to be created; computer science, digital instrumentation, and instruments configured by software face unique challenges in the context of actual astronomical research questions, technical requirements, and real-world big data and data science.

Astronomy and computer science have begun to influence each other in profound ways. Scaling in hardware and software—for instance, through more powerful parallel computing—enables scientists to ask and answer more complex research questions and to examine the universe in more detail. Conversely, astronomy can drive strategic directions in computer science. For example, datacenters and parallel computers designed to work under astronomy's extreme demands and constraints will also certainly work in less challenging everyday environments. Extreme testbeds provide real-world benchmarks, and offer clues about important frontiers of computer science research.

Various community efforts and interest groups are emerging in astroinformatics to advance the field at the intersection of computer science and astronomy. For example, portals like the Astrostatistics and Astroinformatics Portal[1] are providing material, and astronomers and information scientists have signed a position paper[2] to advocate for astroinformatics as a new discipline that includes data organization, data description, astronomical classification taxonomies, astronomical concept ontologies, data mining, machine learning, visualization, and astrostatistics. Other parts of the community[3] have a somewhat different perspective and propose the inclusion of other areas, such as computational infrastructure, astrophysical simulations, and scientific software engineering. Still other authors classify their work more broadly as "astronomical computing" or "computational astronomy."

As computer scientists, we believe that astroinformatics should include all of the aforementioned areas, as we require such cross-boundary collaboration in order to make progress. In addition, we also need to consider specialized signal and image processing, research in fusing different types of observation data, emerging areas such as crowdsourcing for data collection and analysis, and open source community development. It will take some time to converge on an exact definition of astroinformatics, but throughout this discourse, problems are being solved along the way nevertheless.

A future where an astroinformatics community includes computer scientists, astronomers, physicists, engineers, mathematicians, and other interested scientists will enable even more significant breakthroughs. We hope you enjoy this special issue and look forward to seeing you contribute to the exciting future of astroinformatics.

References

1. E. Feigelson and J. Hilbe, eds., "Astrostatistics and Astroinformatics Portal," 2014; http://asaip.psu.edu.
2. K.D. Borne, et al. "Astroinformatics: A 21st Century Approach to Astronomy," Arxiv, 2009; http://arxiv.org/abs/0909.3892.
3. A. Accomazzi et al., "Astronomy and Computing: A New Journal for the Astronomical Computing Community," *Astronomy and Computing*, vol. 1, 2013, pp. 1–4.

Victor Pankratius *is a research scientist at MIT, Haystack Observatory, and a collaborative partner of the Computer Science and Artificial Intelligence Laboratory (CSAIL). His research interests include parallel and multicore computing, software engineering, and astro- and geoinformatics. Pankratius received a Habilitation degree in computer science from KIT and a Dr.rer.pol. with distinction from the University of Karlsruhe, Germany. Contact him at pankrat@mit.edu or the Web at www.victorpankratius.com, or on Twitter @vpankratius.*

Chris Mattmann *is chief architect in the Instrument and Science Data Systems at NASA Jet Propulsion Laboratory and adjunct associate professor at the University of Southern California. His research interests include software architecture, big data, and search engines and information retrieval. Mattmann received a PhD in computer science from USC, and is a senior member of IEEE. Contact him at mattmann@jpl.nasa.gov, on the Web at http://sunset.usc.edu/~mattmann, or on Twitter @chrismattmann.*