Journal of Big Data
a SpringerOpen Journal

SHORT REPORT Open Access

Cultivating a research agenda for data science

Chris A Mattmann[1,2]

Correspondence:
chris.a.mattmann@nasa.gov
[1]Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive M/S 171-264, 91109 Pasadena, USA
[2]Computer Science Department, University of Southern California, 941 W. 37th Place, 90089 Los Angeles, USA

Abstract

I describe a research agenda for data science based on a decade of research and operational work in data-intensive systems at NASA, the University of Southern California, and in the context of open source work at the Apache Software Foundation. My vision is predicated on understanding the architecture for grid computing; on flexible and automated approaches for selecting data movement technologies and on their use in data systems; on the recent emergence of cloud computing for processing and storage, and on the unobtrusive and automated integration of scientific algorithms into data systems. Advancements in each of these areas are a core need, and they will fundamentally improve our understanding of data science, and big data. This paper identifies and highlights my own personal experience and opinion growing into a data scientist.

Keywords: Data dissemination; Open source; Science algorithm integration; Data science; Software architecture; Big data

Findings

Over the last decade I have been primarily engaged in research associated with NASA's Jet Propulsion Laboratory (JPL), the University of Southern California and the Apache Software Foundation. The research has explored the fundamentally changing paradigm of data-intensive systems and its emerging frontier of *Big Data* and *Data Science*, and on how software architecture and software reuse can assist in bridging the boundary in science from a previously silo'ed and independent nature to one that is increasingly more collaborative, and multi-disciplinary. This research has been applied in the development and delivery of ground data systems software for a number of national scale projects including the next generation of NASA's Earth science missions (OCO/OCO-2, NPP Sounder PEATE, SMAP, etc.); the National Cancer Institute's Early Detection Research Network (EDRN), NSF funded activities in geosciences, and radio astronomy, and also in the recent context of DARPA's BigData initiative called XDATA.

This paper identifies and highlights my own personal experience growing into a *data scientist*. I begin by describing a nexus of training in grid computing and software architecture inspired through work on the Apache Object Oriented Data Technology (OODT) project. A need to compare OODT with similar grid technologies led to precise and specific architectural analyses of grid software and an effort to more completely describe the architecture of grid computing based on a study of nearly twenty topical grid technologies over the last decade. This analysis identified gaps in the grid computing realm, specifically in the areas of data management, and in cataloging and archiving. Grid computing

systems required stronger support for file and metadata management; for workflow processing and for resource management. Furthermore grid systems needed stronger approaches for automated ingestion, for remote content acquisition and for science algorithm integration, *unobtrusively* and *rapidly*. I describe the contributions made in these areas, and the influence that the web search engine community and open source community had on my specific contributions. Cloud computing and its benefits for processing and storage are described in the context of my experience studying data movement and in comparing science processing approaches. I identify the importance of open source communities and of open source foundations including the Apache Software Foundation as well.

Drawing upon the above background, I close with my vision for data science, in the form of four specific areas where fundamental and advanced research must be conducted. The areas are: (1) Rapid Algorithm Integration; (2) Intelligent Data Movement; (3) Use of Clod Computing; and (4) Use of Open Source as a mechanism to bringing the new architectures and contributions to the masses.These areas are influenced by my personal experience and this provenance is highlighted in the conclusion of the paper.

Apache OODT: a data science learning environment

My work has focused on the nexus between software architecture and grid computing, with an eye towards empirically evaluating data movement technologies and developing approaches for rapidly and automatically assessing their suitability for scientific data dissemination scenarios [1-3] in the context of the Apache OODT project [4]. Apache OODT is an open source, data-grid middleware used across many scientific domains, such as astronomy, climate science, snow hydrology, planetary science, defense and intelligence systems, cancer research, and computer modeling, simulation and visualization. The framework itself contains over 10 years of work and 100+ FTEs of investment and holds the distinction as NASA's *first ever* project to be stewarded at the open source Apache Software Foundation. Apache is a 501(c)(3) non-profit focused on developing world-class software for no charge to the public and is home to the some of the most prolific and well-known software technologies including the Apache HTTPD web-server that powers the majority share (53%) of the Internet; as well as emerging *Big Data* technologies that I have helped to pioneer including Apache Nutch, Apache Hadoop, Apache Tika, and Apache Lucene/Solr. I am currently a member of Apache's *Board of Directors* for the 2013-14 term. My experience at Apache and within OODT has influenced research agenda item (4) Use of Open Source, described later in the *Looking towards the future* section of the paper.

Drawing inspiration from the grid

While studying grid computing and data-intensive systems including OODT, I found that little software engineering and architecture research work was performed to characterize the architectural properties of grid computing, besides the initial pioneering work by Kesselman and Foster to define the grid's *anatomy* [5], and *physiology* [6], respectively. Namely, the grid's reference requirements, its detailed physical architecture and mapping to implementation technologies was missing – especially considering that so many technologies (including OODT) claimed to be a "grid" technology. So, I undertook several studies to develop automated approaches for discerning the grid's reference

architecture and requirements, and its detailed as-implemented architecture as evidenced from code, requirements, free-text documentation, and other information from over 20+ topical open source software systems claiming to be a grid. The initial study I published in 2005 [7] at the *Component-based Software Engineering* conference represented early work only focusing on 5 of the eventual 20 technologies and only on the approach for automatically recovering a grid's architecture – four years later I expanded the work [8] and actually identified a new grid reference architecture, demonstrating how the as-recovered architectures of grid technologies better mapped to it when compared with the original grid's anatomy and physiology. An expanded version of this work is currently under review with *J. Grid Computing*. Our work in identifying a more descriptive grid architecture strong suggests that current architectures and software may not be appropriate and that new architectural approaches and paradigms will be needed in the Big Data domain.

I took the knowledge and research products from studying grid computing systems and better defining their architecture and applied this to the design of several national scale systems across scientific domains. In particular, from 2005-2009, I led the development of NASA's Orbiting Carbon Observatory (OCO) ground data system, as well as the National Polar Orbiter Earth System Satellite (NPOESS) Preparatory Project (NPP) and its Sounder data Product Evaluation and Testbed Element (PEATE), two next generation data systems that took NASA into the realm of *Big Data*. The prior NASA Earth science missions that I had worked on (QuickSCAT/Seawinds) had a database catalog and archive that grew to 10 gigabytes after ten years of operations/extended mission – OCO's catalog and archive would eclipse 150+ terabytes within the *first three months of operations*. QuickSCAT/Seawinds regularly processed in the order of tens of jobs per day – OCO and NPP PEATE would eclipse tens of *thousands* of jobs per day. The application of OODT to these science data systems and the specific architectural description of grid computing derived from the study of grid technologies has influenced research agenda item (3) to better understand cloud computing; and also agenda item (4) the use of open source software to construct these data systems described later in the *Looking towards the future* section of the paper.

Flexible cataloging and archiving

The requirements and shift in paradigm for OCO and NPP PEATE led me to lead a large refactoring and modernization of the Apache OODT data processing subsystem called CAS (for "Catalog and Archive System"). The OODT CAS, under my leadership, underwent a series of changes.

First, I separated the CAS from a monolithic component that handled both aspects of file and metadata management, and split that component into its constituent functionalities – a *File Manager* component to handle ingestion; data movement, and cataloging/archiving of files and metadata – and a *Workflow Manager* component to model data and control flow; tasks, their execution and lifecycle, and workflow metadata. In large part the efforts to refactor the Workflow Manager component were based on the pioneering research by Dr. Raj Buyya, and his Taxonomy of Workflow Management Systems for Grid Computing [9]. Taking the refactoring a step further, and also expanding on my research into the Ganglia and Gexec resource management, monitoring and execution systems [10], I went ahead and expanded the CAS to also include a *Resource Manager* component, separate from the Workflow Manager, whose job was to model the

requirements for job execution (e.g., requires X% CPU, or requires Y disk space; or Z programming language, e.g., IDL/Python/etc., to run), and also the current monitored status (load, CPU, etc.) of the hardware and computing resources for the job to run on.

I published the results of this initial refactoring at the IEEE Space Mission Challenges for Information Technology conference with my co-authors that included computer scientists, and experts in chemistry and spectroscopy, and in climate science [11]. This particular experience has helped to influence research agenda item (1) rapid and unobtrusive science algorithm integration described later in the *Looking towards the future* section of the paper.

Drawing on the web search experience

In addition to the above initial refactoring, I also drew from my experience helping to develop Apache Nutch [12], a large-scale, distributed search engine, the predecessor to Apache Hadoop, the current industry standard Big Data technology. While developing Nutch, I contributed to (at the time, and still one of the largest and most widely used) web crawler/fetchers that existed.

Drawing upon this experience for Nutch and improving upon it, I modeled a new CAS component for OODT based upon the Nutch fetcher system – the new component was called Push Pul [13], and its responsibility was to negotiate the myriad web and other protocols for acquiring remote content available both on the web, from FTP servers, and from other data servers accessible from a URL protocol scheme. Different from Nutch, I designed Push Pull to separate its remote content acquisition functionality from the actual ingestion and crawling process. This was in direct response to real world experience and also drawing upon my software architecture experience and research when I realized that remote content acquisition is a large enough and complex enough functionality to warrant its own separate stack of services.

Separating remote content acquisition from actual ingestion was also a realization of my PhD dissertation work wherein which I demonstrated that data movement and acquisition technologies experience largely different qualities of service depending on data dissemination scenarios – so by separating Push Pull as its own component, we could isolate a major potential bottleneck in a data-intensive and grid software system, allowing it to evolve independent, and be improved independently of local ingestion. So, with Push Pull in hand, I also drew from Nutch and my experience building the Apache Tika [14] content detection and analysis framework to construct the CAS crawler, an automated ingestion, file detection and classification technology that works in concert with Push Pull to ingest remote and local content. During this time I also co-wrote a full book on Apache Tika published by Manning and one that I use to teach CSCI 572. Search Engines and Information Retrieval at USC.

The work in this area especially in remote content acquisition has helped to bring the importance of research agenda item (2) intelligent data movement and the need for a better understanding of existing remote content acquisition systems and data delivery methods described later in the *Looking towards the future* section of the paper.

Computing the same way the scientists do

The other major research contribution I delivered based on the OODT CAS was the development of a software framework for rapid science algorithm integration. The new

system, called "CAS PGE" [15] codifies a single step in the overall scientific process as a workflow task and leverages Apache OODT, Apache Tika, Apache Solr and other Big Data software systems that I have helped to principally construct.

CAS-PGE uses these software to stage file input and metadata; to allow for automatically selected and optimal data movement services; to seamlessly execute IDL, Matlab, Python, R and other custom scientific codes; to perform automatic metadata and text extraction from the scientific algorithm outputs; and finally to capture of workflow provenance and metadata as produced by the algorithm.

The CAS-PGE framework has proven to be an effective encapsulation for not just the scientific step in an investigation, but also for unobtrusively integrating algorithms into large scale production workflow and Big Data systems, without having to rewrite the algorithm. This is a key insight that I developed from this work to help reduce cost and risk in scientific software and to preserve the stewardship of the algorithms in the scientific communities where they are developed. This experience has heavily influenced research agenda item (1) the rapid and unobtrusive integration of science algorithms described later in the *Looking towards the future* section of the paper.

Harnessing the cloud

I am also interested in cloud computing, and in its use for processing and storage within software systems. I have led several studies since 2010 to investigate: (1) cloud computing as a platform for data movement, and storage [16]; (2) cloud computing as a platform for scientific processing [17]; and (3) a hybrid combination of public and private cloud resources for storage, processing and for platform virtualization [18]

The contributions from these studies involved the identification of when, and where to leverage cloud in a software system's architecture; a comparison model for cloud versus local storage and processing resources, and a set of insights for delivering cloud-based virtual machines with data system software to the Earth science community. These and other contributions were disseminated at the 2011 International Conference on Software Engineering SECLOUD (Software Engineering for Cloud Computing) workshop that I chaired [19]. Experience in this area has suggested research agenda item (3) a better understanding of the implications of cloud computing for storage and processing described later in the *Looking towards the future* section of the paper.

Keeping the door open

Experience working throughout many life, physical, natural, Earth and planetary scientific domains has increased my interest in collaboration both in terms of science but also software – making it *open source* and its nexus within software reuse, and software engineering. I have led and published several topical studies exploring open source as a framework for enabling scientific collaboration, and as a framework for software reuse, including the cover feature [20] of the IEEE IT Professional magazine's special issue on NASA's contributions to IT, as well as a study published [21] exploring the role of open source in NASA's program called ESDIS, for Earth Science Data and Information System, the program under which the Earth science Distributed Active Archive Centers (DAACs) are housed; and a study of open source in the National Cancer Institute's Early Detection

Research Network (EDRN) program [22] which includes over 40+ institutions all performing cancer biomarker research for early stage detection, a program funded for over 10+ years by the NCI.

I have also chaired several open source topical meetings of relevance exploring its connection to science including the Apache in Space! (OODT) track in 2011 at ApacheCon, and the Apache in Science track at the 2013 meeting, as well as several organized open source meetings at the American Geophysical Union (AGU) Fall meeting for the past three years, and at the Earth Science Information Partners (ESIP) Foundation meetings during that same time. I am also one of the lead organizers of the Open Source Summit [23], a meeting that originally began with only NASA participation and has grown to include over 12 government agencies including NASA, NSF, NIH/NCI, NLM, DARPA, DOD, the State Department, the Census Bureau and other agencies.

My primary research contribution in this area is an identification of a classification and comparison framework for open source software based on nine dimensions of importance including licensing; community-structure (open, closed, etc.); redistribution strategy; attribution strategy and more. The work in this area has influenced research agenda item (4) the understanding and application of open source in Big Data and in data science described next in the *Looking towards the future* section of the paper.

Looking towards the future

Based on the above research history and background, I published an article in *Nature* magazine in January 2013 [24] identifying the four thrusts of my research vision for *Data Science* and *Big Data*. The four main areas of advancement that I plan to investigate over the next decade are:

Rapid science algorithm integration Researchers need to do a better job at rapidly and unobtrusively integrating scientific algorithms into Big Data production systems and workflow systems. The current state of the art is to tell a scientist to rewrite her algorithm in Map Reduce in order to make it faster, or to integrate it into a data system – this takes away from the scientific stewardship of the algorithm and transfers it to the software engineering team, who may lack the necessary background and training to maintain that algorithm, and furthermore, largely computer scientists are not trained in scientific programming environments like Matlab, R, Python, IDL, etc. Scientific Workflow Systems can help here [25], and also current efforts for DARPA XDATA, NASA's RCMES project, and for NSF EarthCube will provide an evaluation environment for future work in this area.

Intelligent data movement At a recent Hadoop Summit meeting, I recall the VP of Amazon Web Services explaining to an audience member what the best way to send 10+ terabytes of data to Amazon would be in order to process it on EC2. The VP made some joke about "Well, you know how Amazon is *really great* at shipping things to *you* – in this case, you ship things to us, that is, *your data*". This is very much still the state of the art and practice for data movement – shipping "data bricks" around. This is an extremely cost effective and viable option, however the decisions and rationale and scientific reasons as to *why* data movement selections be them electronic (GridFTP, bbFTP, HTTP, REST, etc.) or hardware ("brick") based are made are largely undocumented, not reproducible and an art form. In

other words, the selection of a data movement technology *does matter*, can affect all sorts of functional properties in a Big Data system, and ultimately is a key portion of the architecture yet as a field we do not have good reasons as to why particular data movement technologies are chosen, and others ignored. This is an exciting future area of research, since it both continues my PhD work, and also has practical applications for technology transfer e.g., into Amazon, the open source community, NASA Earth Science missions, the SKA project, etc., as well as very fruitful domains for evaluation in industry, climate science, astronomy and future and current Big Data projects.

Appropriate use of cloud computing for storage/processing To develop effective architectural and software engineering techniques for cloud computing services to both assess their cost, and also the suitability of their processing and storage components, researchers need to perform an assessment of cloud computing vendors and providers and their integration capabilities, processing and storage capabilities. Further, studies attempting to discern the canonical software components and services for the cloud are timely and needed. Understanding cloud providers that are both reusable, and that fulfill software architecture requirements, and ultimately the requirements of the Big Data system is an important step in this process. This research and software engineering understanding of the cloud is an area that will have large applicability and technology transfer potential.

Harnessing the power of open source in software development for science Identifying the methods and the approach in which open source foundations, legal frameworks, and licenses affect software development, and scientific collaboration is an important near-term research study that is required. We currently lack strong empirical research and data that identifies the most appropriate and inappropriate software ecosystems for housing software components. Methods must be developed to track the evolution of software components at these foundations, and to identify the role of emerging distributed versus centralized configuration management (e.g., Git versus Subversion) at these foundations. Strategies that employ social scientists to investigate the community implications of open source, and the effectiveness of open source as a software engineering development process and architectural strategy are in need of development.

Conclusion

I am committed to the above four areas of research and see them as both necessary and exciting if we are to advance the fields of Big Data and data science. I plan to attack the above research areas with a multi-disciplinary eye and to make a contribution in software architecture, design, reuse, and open source. I am excited to pursue these topics and am confident that the results of the pursuit will have a potential for tremendous impact in science and industry and the broader community.

Competing interests
The author declare that he/she has no competing interests.

Acknowledgements
Effort supported by NSF awards PLR-1348450, ICER-1343800, ACS-1125798, and GEO-1229036 and GEO-1343583. Effort also partially supported by the Jet Propulsion Laboratory, managed by the California Institute of Technology under a contract with the National Aeronautics and Space Administration.

Received: 6 January 2014 Accepted: 26 May 2014
Published: 6 August 2014

References

1. Mattmann C, Crichton D, Hughes JS, Kelly S, Hardman S, Joyner R, Ramirez P (2006) A classification and evaluation of data movement technologies for the delivery of highly voluminous scientific data products. In: Proceedings of the NASA/IEEE Conference on Mass Storage Systems and Technologies (MSST2006). IEEE Computer Society, Maryland, pp 131–135
2. Mattmann C, Crichton D, Hart A, Kelly S, Hughes JS (2010) Experiments with storage and preservation of nasas planetary data via the cloud. IEEE IT Prof Spec Theme Cloud Comput 12(5): 28–35
3. Crichton D, Mattmann C, Cinquini L, Braverman A, Waliser D, Hart A, Goodale C, Lean P, Kim J (2012) Sharing satellite observations with the climate modeling community: software and architecture. IEEE Softw 29(5): 63–71
4. Mattmann C, Crichton DJ, Medvidovic N, Hughes S (2006) A software architecture-based framework for highly distributed and data intensive scientific applications. In: ICSE. IEEE Computer Society, Shanghai, China, pp 721–730
5. Foster IT, Kesselman C, Tuecke S (2001) The anatomy of the grid: Enabling scalable virtual organizations. IJHPCA 15(3): 200–222
6. Foster I, Kesselman C, Nick JM, Tuecke S (2002) The physiology of the grid: an open grid services architecture for distributed systems integration. Open grid service infrastructure WG, Global grid forum. http://www.globus.org/research/papers/ogsa.pdf.
7. Mattmann C, Medvidovic N, Ramirez PM, Jakobac V (2005) Unlocking the grid. In: Heineman GT, Crnkovic I, Schmidt HW, Stafford JA, Szyperski CA, Wallnau KC (eds) CBSE. Lecture notes in computer science, vol. 3489. Springer, St. Louis, MO, pp 322–336
8. Mattmann C, Garcia J, Krka I, Popescu D, Medvidovic N (2009) The anatomy and physiology of the grid revisited In: WICSA/ECSA. IEEE/IFIP, London, UK
9. Yu J, Buyya R (2005) A taxonomy of workflow management systems for grid computing. J Grid Comput 3(3–4): 171–200
10. Massie ML, Chun BN, Culler DE (2004) The ganglia distributed monitoring system: design, implementation, and experience. Parallel Comput 30(7): 817–840
11. Mattmann C, Freeborn D, Crichton D, Foster B, Hart A, Woollard D, Hardman S, Ramirez P, Kelly S, Chang AY, Miller CE (2009) A reusable process control system framework for the orbiting carbon observatory and npp sounder peate missions. In: 3rd IEEE Intl' Conference on Space Mission Challenges for Information Technology (SMC-IT 2009). IEEE Computer Society, Pasadena, CA, pp 165–172
12. Cafarella M, Cutting D (2004) Nutch, building open source search. ACM Queue 2(2): 54–61
13. Kang Y, Kung SH, Jang H-J (2013) Simulation process support for climate data analysis In: Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference. ACM, Vietri sul Mare, Italy, p 29
14. Mattmann C, Zitting J (2011) Tika in action. Manning Publications Co., NY, USA
15. Mattmann CA, Crichton DJ, Hart AF, Goodale C, Hughes JS, Kelly S, Cinquini L, Painter TH, Lazio J, Waliser D, Medvidovic N, Kim J, Lean P (2011) Architecting data-intensive software systems. In: Handbook of Data Intensive Computing. Springer, pp 25–57. http://www.springer.com/computer/database+management+%26+information+retrieval/book/978-1-4614-1414-8
16. Mattmann CA, Crichton DJ, Hart AF, Kelly SC, Hughes JS (2010) Experiments with storage and preservation of nasa's planetary data via the cloud. IT Prof 12(5): 28–35
17. Kwoun O-i, Cuddy D, Leung K, Callahan P, Crichton D, Mattmann CA, Freeborn D (2010) A science data system approach for the desdyni mission. In: Radar Conference, 2010 IEEE. IEEE, Arlington, VA, pp 1265–1269
18. Mattmann CA, Waliser D, Kim J, Goodale C, Hart A, Ramirez P, Crichton D, Zimdars P, Boustani M, Lee K, Loikith P, Whitehall K, Jack C, Hewitson B (2013) Cloud computing and virtualization within the regional climate model and evaluation system. Earth Sci Inf: 1–12
19. Mattmann CA, Medvidovic N, Mohan T, O'Malley O (2011) Workshop on software engineering for cloud computing: (secloud 2011). In: Software Engineering (ICSE), 2011 33rd International Conference On. IEEE, Honolulu, HI, pp 1196–1197
20. Mattmann CA, Crichton DJ, Hart AF, Kelly SC, Goodale CE, Ramirez P, Hughes JS, Downs RR, Lindsay F (2012) Understanding open source software at nasa. IT Prof 14(2): 29–35
21. Mattmann CA, Downs RR, Ramirez PM, Goodale C, Hart AF (2012) Developing an open source strategy for nasa earth science data systems. In: Information Reuse and Integration (IRI), 2012 IEEE 13th International Conference On. IEEE, Las Vegas, NV, pp 687–693
22. Hart AF, Verma R, Mattmann CA, Crichton DJ, Kelly S, Kincaid H, Hughes S, Ramirez P, Goodale C, Anton K, Colbert M, Downs RR, Patriotis C, Srivastava S (2012) Developing an open source, reusable platform for distributed collaborative information management in the early detection research network. In: Information Reuse and Integration (IRI), 2012 IEEE 13th International Conference On. IEEE, Las Vegas, NV, pp 263–270
23. Open Source Summit 3.0: Communities. http://ossummit.org/. Accessed 3 March 2014
24. Mattmann CA (2013) Computing: A vision for data science. Nature 493(7433): 473–475
25. Woollard D, Medvidovic N, Gil Y, Mattmann C (2008) Scientific software as workflows: From discovery to distribution. IEEE Softw 25(4): 37–43

doi:10.1186/2196-1115-1-6
Cite this article as: Mattmann: **Cultivating a research agenda for data science.** *Journal of Big Data* 2014 **1**:6.