Research Statement

Chris A. Mattmann

November 14, 2013

Over the last decade I have been primarily engaged in research associated with the University of Southern California, NASA's Jet Propulsion Laboratory (JPL) 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.

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][2][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 – a transition that I personally led. 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 world's 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 co-created and helped to pioneer including Apache Nutch, Apache Hadoop, Apache Tika, and Apache Lucene/Solr. My contributions at Apache have earned me a spot on its current *Board of Directors* for the 2013-14 term.

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

[1] C. Mattmann, D. Crichton, J. S. Hughes, S. Kelly, S. Hardman, R. Joyner and P. Ramirez. 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)*, pp. 131-135, College Park, Maryland, May 15-18, 2006.

[2] C. Mattmann, D. Crichton, A. Hart, S. Kelly, J. S. Hughes. Experiments with Storage and Preservation of NASAs Planetary Data via the Cloud. *IEEE IT Professional Special Theme on Cloud Computing*, Vol. 12, No. 5, pp. 28-35, September/October, 2010. Selected to appear in Essential Articles on Information Technology - Essence of IEEE IT Pro 2010.

[3] D. Crichton, C. Mattmann, L. Cinquini, A. Braverman, D. Waliser, A. Hart, C. Goodale, P. Lean. Sharing Satellite Observations with the Climate Modeling Community: Software and Architecture. *IEEE Software*, Vol. 29, No. 5., September/October 2012, pp. 63-71.

[4] C. Mattmann, D. Crichton, N. Medvidovic and J. S. Hughes. A Software Architectutre-based Framework for Highly Distributed and Data-Intensive Scientific Applications. Proc. of *ICSE - Software Engineering Challenges and Achievements track*, 2006

[5] Foster, Ian, Carl Kesselman, and Steven Tuecke. "The anatomy of the grid: Enabling scalable virtual organizations." *International Journal of High Performance Computing Applications* 15.3 (2001): 200-222.

[6] Foster, Ian, et al. "The physiology of the grid." *Grid computing: making the global infrastructure a reality (2003)*: 217-249.

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] was the highest reviewed paper at the *Component-based Software Engineering* conference and 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*. I led the paper from its inception, to data collection, to the derivation of the new reference architecture and also managed a team of three USC software architecture PhD students with Dr. Medvidović during this effort.

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 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 a collaborator, 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].

[7] Mattmann, Chris A., et al. "Unlocking the grid." *Component-Based Software Engineering.* Springer Berlin Heidelberg, 2005. 322-336.

[8] Mattmann, Chris A., et al. "The anatomy and physiology of the grid revisited." *Joint Working IEEE/IFIP Conference on Software Architecture, 2009 & European Conference on Software Architecture. WICSA/ECSA 2009.* . IEEE, 2009.

[9] Yu, Jia, and Rajkumar Buyya. "A taxonomy of workflow management systems for grid computing." *Journal of Grid Computing 3.3-4 (2005)*: 171-200.

[10] Massie, Matthew L., Brent N. Chun, and David E. Culler. "The ganglia distributed monitoring system: design, implementation, and experience." *Parallel Computing* 30.7 (2004): 817-840.

[11] C. Mattmann, D. Freeborn, D. Crichton, B. Foster, A. Hart, D. Woollard, S. Hardman, P. Ramirez, S. Kelly, A. Y. Chang, C. E. Miller. A Reusable Process Control System Framework for the Orbiting Carbon Observatory and NPP Sounder PEATE missions. In *Proceedings of the 3rd IEEE Intl Conference on Space Mission Challenges for Information Technology (SMC-IT 2009)*, pp. 165-172, July 19 - 23, 2009.

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 off of the Nutch fetcher system – the new component was called Push Pull[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 wrote the definitive guide to Tika, a full book published by Manning and one that I use to teach CSCI 572: Search Engines and Information Retrieval at USC.

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. CAS PGE 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.

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

[12] M. Cafarella and D. Cutting. "Building Nutch: Open source search." *ACM Queue.* v2 i2 (2004): 54-61.

[13] Y. Kang, S. H. Kung, and H. Jang. Simulation process support for climate data analysis. In *Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference (CAC '13).* 2013.

[14] C. Mattmann and J. Zitting. *Tika in Action.* 256 pages. New York: Manning Publications, November 2011. ISBN: 9781935182856. http://www.amazon.com/Tika-Action-Chris-Mattmann/dp/1935182854

[15] C. Mattmann, D. Crichton, A. Hart, C. Goodale, J. S. Hughes, S. Kelly, L. Cinquini, T. H. Painter, J. Lazio, D. Waliser, N. Medvidovic, J. Kim, P. Lean. Architecting Data-Intensive Systems. In *Handbook of Data Intensive Computing*, B. Furht, A. Escalante, eds. 1st Edition. Springer Verlag, 2011.

[16] C. Mattmann, D. Crichton, A. Hart, S. Kelly, J. S. Hughes. Experiments with Storage and Preservation of NASAs Planetary Data via the Cloud. *IEEE IT Professional Special Theme on Cloud Computing*, Vol. 12, No. 5, pp. 28-35, September/October, 2010. Selected to appear in Essential Articles on Information Technology - Essence of IEEE IT Pro 2010 http://yamashita.computer.org/readynotes/SEBundle/ES0000035-ITPro.pdf

[17] O. Kwoun, D. Cuddy, K. Leung, D. Crichton, C. Mattmann, and D. Freeborn. A Science Data System Approach for the DESDynI Mission. In *Proceedings of IEEE Radar*, pp. 1265-1269, Washington, D.C., May 10-14, 2010.

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 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 large $150M+ dollar research 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 the lead organizer 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.

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

[18] C. Mattmann, D. Waliser, J. Kim, C. Goodale, A. Hart, P. Ramirez, D. Crichton, P. Zimdars, M. Boustani, H. Lee, P. Loikith, K. Whitehall, C. Jack, B. Hewitson. Cloud Computing and Virtualization Within the Regional Climate Model and Evaluation System. *Earth Science Informatics*, accepted, July 2013.

[19] Mattmann, Chris A., et al. "Workshop on software engineering for cloud computing:(SECLOUD 2011)." *Proceedings of International Conference on Software Engineering (ICSE),* IEEE 2011

[20] C. Mattmann, D. Crichton, A. Hart, S. Kelly, C. Goodale, R. R. Downs, P. Ramirez, J. S. Hughes, F. Lindsay. Understanding Open Source Software at NASA. *IEEE IT Professional Special Theme on NASA Contributions to IT*, Vol. 14, No. 2, pp. 29-35, March/April 2012. Selected to appear in Essential Articles on Information Technology - Essence of IEEE IT Pro 2012

[21] C. Mattmann, R. R. Downs, P. Ramirez, C. Goodale, A. Hart, Developing an Open Source Strategy for NASA Earth Science Data Systems. In *Proceedings of the IEEE Information Reuse and Integration*, Las Vegas, NV, August 8-10, 2012.

[22] A. Hart, R. Verma, C. Mattmann, D. Crichton, S. Kelly, H. Kincaid, S. Hughes, P. Ramirez, C. Goodale, K. Anton, M. Colbert, R. R. Downs, C. Patriotis, S. Srivastava. Developing an Open Source, Reusable Platform for Distributed Collaborative Information Management in the Early Detection Research Network. In *Proceedings of IEEE Information Reuse and Integration*, Las Vegas, NV, August 8-10, 2012.

[23] `http://ossummit.org/`

[24] C. Mattmann. A vision for data science. *Nature*, Vol. 493, No. 7433, pp. 473-475, January 24, 2013. http://www.nature.com/nature/journal/v493/n7433/full/493473a.html

computer scientists are not trained in scientific programming environments like Matlab, R, Python, IDL, etc. Scientific Workflow Systems can help here, and my own work with USC alum Dr. David Woollard, and Professors Medvidović and Gil from 2008[25], and also current efforts with Dr. Gil 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 Which cloud computing vendors and providers make sense to integrate into your Big Data and software system? For processing? For storage needs? What are the canonical software components and services that are both reusable, and that fulfill software architecture requirements, and ultimately the requirements of the Big Data system? How can we 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? This is an area that will have large applicability and technology transfer potential and I can leverage my background and practical experience towards it.

Harnessing the power of open source in software development for science How can open source foundations, legal frameworks, and licenses affect software development, and scientific collaboration? What are the right software ecosystems for housing software components? How can we track the evolution of software components at these foundations, and what is the role of emerging distributed versus centralized configuration management (e.g., Git versus Subversion) at these foundations? Can we collaborate and team with 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? I have the background and current positions to perform research studies in this area, and I am also funded by DARPA, NASA and the NSF to perform research in this area, and we have real scientific applications, systems, and targets, as well as large industry and market share to explore this research.

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 at the University of Southern California and am confident that the results of the pursuit will have a potential for tremendous impact in science and industry and the broader community.

[25] D. Woollard, N. Medvidovic, Y. Gil, and C. Mattmann. Scientific Software as Workflows: From Discovery to Distribution. *IEEE Software Special Issue on Developing Scientific Software*, Vol. 25, No. 4, July/August, 2008.