As we scale to increasingly parallel and distributed architectures and explore new algorithms and machine learning techniques, the fundamental computational models and abstractions that once separated systems and machine learning research are beginning to fail. Some of the recent advances in machine learning have come from new systems that can apply complex models to big data problems. Likewise, some of the recent advances in systems have exploited fundamental properties in machine learning and analytics to reach new points in the system design space. By considering the design of scalable learning systems from both perspectives, we can address larger problems, expose new opportunities in algorithm and system design, and define the new fundamental computational models and abstractions that will accelerate research in these complementary fields.

Research Summary: My research spans the design of scalable machine learning algorithms and data-intensive systems and has introduced:

- *new machine learning algorithms* that leverage advances in asynchronous scheduling and transaction processing to achieve efficient parallelization with strong guarantees
- *new systems* that exploit statistical properties to execute machine learning algorithms orders-of-magnitude faster than contemporary distributed systems
- *new abstractions* that have redefined the boundaries between machine learning and systems.

Machine Learning: The future of machine learning hinges on our ability to learn from vast amounts of high-dimensional data and to train the big models they support. To create scalable machine learning algorithms that fully utilize advances in hardware and system design, I have adopted a systems approach to machine learning research. By decomposing machine learning algorithms into smaller parts and identifying and exploiting common patterns, I have developed new highly scalable algorithms that leverage advances in system design and provide strong guarantees. In my thesis, I developed, analyzed, and implemented a set of Bayesian inference algorithms that exploit the conditional independence structure of graphical models to parallelize computation across distributed asynchronous systems. As a postdoc, I decomposed nonparametric inference and submodular optimization algorithms into exchangeable transactions and applied techniques in scalable transaction processing to derive parallel algorithms with strong guarantees.

Systems: Many recent developments in large-scale system design were driven by applications in analytics and machine learning that exposed new opportunities for parallelism, scheduling, concurrency control, and asynchrony and led to new points in the system design space. My approach to systems research builds on my knowledge of machine learning to identify patterns that yield new abstractions and system design opportunities. My thesis work introduced the graph-parallel abstraction which has served as a platform for subsequent parallel algorithms and applications, and has become the foundation for a wide range of graph-processing systems. Building on variations of the graph-parallel abstraction, I created the GraphLab and PowerGraph systems which leverage fundamental properties of data to execute machine learning and graph analytics algorithms orders of magnitude faster than contemporary data processing systems. As a postdoc, I created GraphX which unifies data-parallel and graph-parallel systems by developing new distributed join optimizations and new tabular representations of graph data.

Impact: From graph-parallel systems to the parameter server my research produced some of the most widely used open-source systems for large-scale machine learning. The combination of work on algorithms and systems led to the creation of GraphLab Inc. which has already commercialized my research, enabling applications ranging from product targeting to cybersecurity.

Thesis Research

Summary: *My thesis work introduced algorithms, abstractions, and systems for scalable inference in graphical models and played a key role in defining the space of parallel inference algorithms and graph processing systems and abstractions.*

Graphical Model Inference Algorithms: I developed, analyzed, and implemented a collection of message passing [1, 2, 3] and MCMC [4] Bayesian inference algorithms that leverage the Markov Random Field structure to efficiently utilize parallel processing resources. By asynchronously constructing prioritized spanning trees, these algorithms expose substantial parallelism and also accelerate convergence. The work on MCMC inference incorporated techniques in graph coloring with new Markov blanket locking protocols to preserve ergodicity while enabling the application of parallel resources. By abandoning the popular bulk synchronous parallel model, and envisioning a new more asynchronous graph-centric model, I was able to design more scalable algorithms with stronger guarantees. This early work on algorithm design laid the foundation for the design of graph-parallel abstractions and systems and was a key factor in the broad adoption of the GraphLab open-source project.

Abstractions: Guided by the work on parallel inference, I introduced the graph-parallel [5, 6], gather-apply-scatter (GAS) [7], and parameter server [8] abstractions which capture the fundamental computational and data-access patterns in a wide range of machine learning algorithms. These abstractions isolate the design of machine learning algorithms from the challenges of large-scale distributed asynchronous systems enabling research into algorithms and systems to proceed in parallel. The graph-parallel and GAS abstractions, exploit common properties in graph algorithms to expose new opportunities to leverage asynchrony and optimize communication and data-layout. The parameter server abstraction exploits the abelian group structure and sparse parameter updates in many machine learning algorithms to expose new opportunities for distributed caching and aggregation. The work on graph parallel abstractions has played a key role in recent graph systems and algorithms research and has influenced systems including GraphChi, X-Stream, and Naiad. The simplicity and widespread applicability of the parameter server has led to its adoption in many large-scale machine learning systems including many of the recent systems for deep learning and language modeling. The work on abstractions reduced the complexity of machine learning algorithms to simple computational patterns that could be analyzed for the parallelism and communication overhead and rendered into a range of systems spanning multicore and distributed architectures.

Systems: Building on the abstractions and common properties in data, I developed the GraphLab [6] and PowerGraph [7] systems. The GraphLab system introduced asynchronous prioritized scheduling in conjunction with concurrency control primitives to enable efficient parallel execution of graph algorithms while ensuring serializability. The PowerGraph system exploited the power law graph structure and vertex-cut partitioning to efficiently compute on large real world graphs in a distributed environment. These systems were able to achieve several orders-of-magnitude performance gains over contemporary map-reduce systems and remain the gold standard for general purpose graph processing systems. Through these projects, I helped lead a group of junior graduate students to develop their research and cultivate a growing open-source community. Inspired by the wide-spread adoption of the GraphLab open-source project and commercial interest in applications ranging from product targeting to language modeling, I co-founded GraphLab Inc. which has successfully commercialized these systems with customers in industries as diverse as e-commerce and defense.

Post-doctoral Research

Summary: *As a post-doc advising an exceptional group of graduate students, I adopted a database systems perspective on the design of machine learning systems to unify graph-processing and general purpose dataflow systems and introduce scalable transaction processing techniques to the design of parallel machine learning algorithms.*

The GraphX System: Driven by the need to support the entire graph analytics pipeline which combines tabular pre-processing and post-processing with complex graph algorithms, I led the development of the GraphX system [9] to unify tables and graphs. I revisited my earlier work in the design of graph-processing systems through the lens of distributed database systems. GraphX recast the essential patterns and optimizations in graph-processing systems as new distributed join strategies and new techniques for incremental materialized view maintenance. Through this alternative perspective, GraphX integrates graph computation into a general purpose distributed dataflow system, enabling users to view data as tables or graphs without data movement or duplication and efficiently execute graph algorithms at performance parity with specialized graph-processing systems. GraphX is now part of Apache Spark and has been put into production at major technology companies (e.g., Alibaba Taobao).

Transaction Processing for Machine Learning: I introduced techniques in scalable transaction processing to the design of new parallel algorithms for nonparametric clustering and submodular optimizations. By applying classic ideas in optimistic concurrency control to the nonparametric DP-means clustering algorithm, I developed a parallel inference algorithm which preserves the guarantees of the original serial algorithm [10]. Inspired by escrow techniques, which maintain bounds on the global state, I derived a parallelization of the double greedy algorithm for non-monotone submodular maximization which retains the original approximation guarantees [11]. One of the key contributions of this work is to flip the design and analysis of asynchronous algorithms from *"fast and sometimes correct"* to *"correct and often fast"*. As a consequence, this work exposes the opportunity for a new line of approaches to parallel algorithm design and analysis.

Future Research

In addition to continuing research on the design of scalable Bayesian inference algorithms, graph-processing systems, and transaction techniques for machine learning I plan to explore two new directions in the design of systems for machine learning: *machine learning lifecycle management* and *the unification of synchronous and asynchronous abstractions*.

Model Serving and Management: Machine learning and systems research have largely focused on the design of algorithms and systems to train models at scale on static datasets. However, training models is only a small part of the greater *lifecycle of machine learning* which spans training, model serving, performance evaluation, exploration, and eventually re-training. Furthermore, in many real-world applications there are often many models and modeling tasks which may share common data and incorporate user level personalization (e.g., spam prediction and content filtering). This bigger picture introduces a wide range of exciting new challenges in both the design of models and algorithms as well as the systems needed to support each of these stages. Below I enumerate just a few of these opportunities:

- **Low-Latency Model Serving:** Serving predictions can be costly, requiring the evaluation of complex feature functions, retrieval of user personalization information, and potentially slow mathematical operations. Existing systems resort to pre-materialized predictions or specialized bespoke prediction services limiting their scalability and broader adoption. I believe that we can leverage advances in distributed query processing, caching, partial materialization, and model approximations to enable more general purpose low-latency prediction serving.
- **Hybrid Online/Offline Learning:** Typically models are trained offline at fixed intervals (e.g., every night) resulting in stale models. While online learning algorithms exist, there is limited systems support and offline re-training can often be more efficient and improve estimation quality (e.g., by iterating multiple times). I believe that by splitting models into online and offline components we can achieve a compromise by enabling fast updates to rapidly changing parameters (e.g., personalization parameters) while leveraging existing offline training systems for slowly evolving parameters.
- **Managing Multiple Models:** Often there will be multiple models for the same task (e.g., models built by different employees). Choosing the right model and sharing training and prediction computation across models can improve accuracy and system performance. Model selection is often accomplished using A/B testing, however this approach is deficient as model performance can vary across user groups and time. I believe that by applying active learning techniques we can adaptively select the best models for different groups at different times. This will require the development of new systems to support active learning in a low-latency serving environment.

I have already started initial work on systems for model management and serving [12] and I believe that answering these questions could fill a decade of exciting research and shape the future of machine learning systems. By developing the algorithms and systems needed address the entire machine learning lifecycle we will be able to make better use of data, incorporate predictive analytics in a wide range of services, and enable the new highly responsive, personalized intelligent services that will drive everything from advertising to health-care.

Hybrid Synchronous and Asynchronous Systems: Much of my early work on graph processing systems, the parameter servers, and even transactional models for machine learning leveraged nondeterminism and asynchrony to improve performance. Meanwhile, many contemporary data processing systems have abandoned asynchrony, in favor of the simpler Bulk Synchronous Parallel (BSP) execution model and the determinism it affords. This leads to the question: can we combine the benefits of asynchronous systems and the simplicity and determinism of synchronous systems? I have already begun [13] to explore a hybrid approach to algorithm and system design that provides more frequent, fine-grained communication, while retaining determinism at different levels of granularity. Building on the concept of mini-batch algorithms in machine learning, I believe their is a pattern and corresponding abstraction that can interpolate between the extremes enabling users to choose the level of coordination that leads to the optimal trade-off between algorithm convergence rates and system performance. Understanding how and when to exploit non-determinism while preserving guarantees on algorithm correctness will be a key part of managing the machine learning lifecycle and exploiting the high-performance systems of the future.

References

[1] J. Gonzalez, Y. Low, and C. Guestrin. Residual splash for optimally parallelizing belief propagation. In *Artificial Intelligence and Statistics (AISTATS)*, April 2009.

[2] J. Gonzalez, Y. Low, C. Guestrin, and D. O'Hallaron. Distributed parallel inference on large factor graphs. In *Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2009.

[3] Joseph Gonzalez, Yucheng Low, and Carlos Guestrin. *Scaling Up Machine Learning*, chapter Parallel Inference on Large Factor Graphs. Cambridge U. Press, 2010.

[4] Joseph Gonzalez, Yucheng Low, Arthur Gretton, and Carlos Guestrin. Parallel gibbs sampling: From colored fields to thin junction trees. In *Artificial Intelligence and Statistics (AISTATS)*, May 2011.

[5] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, and J. M. Hellerstein. Graphlab: A new parallel framework for machine learning. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2010.

[6] Yucheng Low, Joseph Gonzalez, Aapo Kyrola, Danny Bickson, Carlos Guestrin, and Joseph M. Hellerstein. Distributed graphlab: A framework for machine learning and data mining in the cloud. In *Proceedings of Very Large Data Bases (PVLDB)*, 2012.

[7] Joseph E. Gonzalez, Yucheng Low, Haijie Gu, Danny Bickson, and Carlos Guestrin. Powergraph: Distributed graph-parallel computation on natural graphs. In *OSDI '12*, 2012.

[8] Amr Ahmed, Mohamed Aly, Joseph Gonzalez, Shravan Narayanamurthy, and Alex Smola. Scalable inference in latent variable models. In *Conference on Web Search and Data Mining (WSDM)*, 2012.

[9] Joseph E. Gonzalez, Reynold S. Xin, Ankur Dave, Daniel Crankshaw, Michael J. Franklin, and Ion Stoica. Graphx: Graph processing in a distributed dataflow framework. In *Symposium on Operating Systems Design and Implementation (OSDI 14)*, 2014.

[10] Xinghao Pan, Joseph E. Gonzalez, Stefanie Jegelka, Tamara Broderick, and Michael I. Jordan. Optimistic concurrency control for distributed unsupervised learning. In *NIPS '13*, 2013.

[11] Xinghao Pan, Stefanie Jegelka, Joseph E. Gonzalez, Joseph K. Bradley, and Michael I. Jordan. Parallel double greedy submodular maximization. In *NIPS '14*, 2014.

[12] Daniel Crankshaw, Peter Bailis, Joseph E. Gonzalez, Haoyuan Li, Zhao Zhang, Michael J. Franklin, Ali Ghodsi, and Michael I. Jordan. The missing piece in complex analytics: Low latency, scalable model management and serving with velox. In *CIDR '15*, 2015.

[13] Peter Bailis, Joseph E. Gonzalez, Ali Ghodsi, Michael J. Franklin, Joseph M. Hellerstein, Michael I. Jordan, and Ion Stoica. Asynchronous complex analytics in a distributed dataflow architecture. In *Under Review for SIGMOD '15*, 2015.

[14] Veronika Strnadova, Aydin Buluc, Leonid Oliker, Joseph Gonzalez, Stefanie Jegelka, Jarrod Chapman, and John Gilbert. Fast clustering methods for genetic mapping in plants. In *16th SIAM Conference on Parallel Processing for Scientific Computing*, 2014.

[15] David Bader, Aydın Buluç, John Gilbert, Joseph Gonzalez, Jeremy Kepner, and Timothy Mattson. The graph blas effort and its implications for exascale. In *SIAM Workshop on Exascale Applied Mathematics Challenges and Opportunities (EX14)*, 2014.

[16] Evan Sparks, Ameet Talwalkar, Virginia Smith, Xinghao Pan, Joseph Gonzalez, Tim Kraska, Michael I Jordan, and Michael J Franklin. Mli: An api for distributed machine learning. In *International Conference on Data Mining (ICDM)*. IEEE, December 2013.

[17] Reynold Xin, Joseph E. Gonzalez, Michael Franklin, and Ion Stoica. Graphx: A resilient distributed graph system on spark. In *SIGMOD Grades Workshop*, 2013.