An Autonomous Distributed Computing Infrastructure

Research Vision and Background. My research interests are at the intersection of Distributed Systems and Machine Learning. My research goal is to build an autonomous distributed computing infrastructure to support the continuous operation of data-driven real-time control systems (e.g., self-driving cars, smart cities, industrial robotics, and earthquake and wildfire monitoring). My approach to creating this autonomous infrastructure has three main areas of focus: (1) facilitating the dynamic aggregation of infrastructure along the cloud-edge path, (2) developing a declarative language to specify operational constraints and optimization objectives, and (3) utilizing Machine Learning (ML) and Deep Reinforcement Learning (Deep RL) techniques to translate these constraints and objectives to runtime directives that continuously adapt the infrastructure to workload and resource variability.

In my dissertation, I developed a middleware that enables the dynamic on-demand aggregation of distributed heterogeneous computing infrastructure (e.g., clouds, supercomputers, grids, clusters, etc.). I leveraged both a rule-based and a constraint programming—based declarative language to manage resource coordination. The resulting framework adapted in near real-time to changes in application behavior, infrastructure properties or availabilities, or Quality of Service (QoS) objectives. The framework enabled large-scale science in multiple domains and was first to enable the deployment of Docker containers across multiple clouds.

In my postdoctoral research, I focused on understanding the requirements of real-time applications in the context of the Internet of Things (IoT) and exploring the applicability of ML techniques in real-world settings. In particular, I worked on developing ML models for smart buildings, which can be used in conjunction with Deep RL to control smart buildings. I am also investigating the design trade-offs in hybrid cloud-edge environments and developing a decision engine to support the decomposition & orchestration of serverless applications in such environments.

My future research aims to build a system that can seamlessly combine the capabilities of distributed compute, storage, and network without the need for human intervention (i.e., autonomous). Such a system would enable the composition of infrastructure in a dynamic, adaptive, programmable, and self-optimizing manner. In particular, I plan to a) investigate application malleability and bidirectional negotiation between an application and the underlying infrastructure, b) model and predict application and distributed infrastructure behavior using statistical ML models, c) find near-optimal resource compositions that can continuously evolve based on historical actions using Deep RL techniques, and d) democratize access to the infrastructure and facilitate its usability using a declarative language that specifies application dependencies, data flow, and operational objectives and constraints.

Impact. My research facilitates many scenarios in science, engineering, IoT, AI, and enterprise applications. For example, it prevents vendor lock-in by allowing users to cloudburst their applications while taking advantage of different services and prices from different cloud providers [8]. My research also allows users to easily define the distribution of different workloads across multiple resources or allow resource providers to easily control what resources are available to users [11]. Further, this work enables the composition of distributed infrastructure based on application runtime behavior (e.g., composing different types of resources based on different stages of a workflow), or based on application results that cannot be determined beforehand (i.e., the progress of the execution) [10]. Finally, my work can support emerging IoT applications [5, 6] by combining resources across cloud-edge environments, or enable new scenarios such as (1) follow the user: a location-based service composition – where resources can be dynamically provisioned to remain within a certain proximity of a moving user (e.g., a self-driving car); or (2) follow the data: a data-based service composition – where resources can be dynamically provisioned based on proximity to data sources/sinks.

In this regard, this work is being used by many world-renowned institutions including Rutgers Cancer Institute (to support the acceleration of medical image registration using federated cloud infrastructure), IBM T.J. Watson Research Center (to support the deployment of Docker containers across hybrid on-premise/cloud environments), the University of Texas at Austin (to run large-scale ensembles of oil reservoir history matching simulations across multiple supercomputers), and San Diego Supercomputing Center at the University of California, San Diego (as part of the WIFIRE project, which aims to integrate real-time remote sensor data with computational techniques to help predict a wildfire's rate of spread).

Motivation & Research Challenges. Increasing computational capabilities coupled with an exponential growth of digital data sources has given rise to a data deluge. Transforming this massive quantity of information from data to insight requires innovations in the computational lifecycle. Similarly, emerging applications are becoming increasingly dynamic and inherently data-driven. Supporting this new class of applications requires rethinking current practices by creating a flexible computing infrastructure, which can autonomously combine compute, data, and communication services in order to obtain insights in a timely manner. This infrastructure must also be able to evolve over time in order to support dynamic and complex workflows. While the service model (enabled by cloud computing) provides the necessary flexibility, it is not clear how or when to compose different services to meet (or anticipate) the needs of next-generation applications and workflows. For instance, this system is relevant to the management of natural disasters, such as monitoring and predicting wildfires. In this case, large amounts of data need to be analyzed at a near-real-time rate to predict how a fire would propagate. This data needs to be retrieved from specific locations and the sources can include field sensors, weather forecasting, satellite images, and social media feeds from an evolving crisis event. Combining this data with a simulation framework can provide advanced decision support to effectively manage the crisis situation. However, this data must be analyzed close to the source to obtain timely insights and direct first responders to critical areas. Hence, being able to dynamically compose services (e.g., near these data sources) and automatically (e.g., when an event is detected) is key.

Current approaches to managing dynamic applications often target one aspect of the environment, namely either by scheduling applications while optimizing user/resource providers' QoS objectives [19] or exploiting elasticity of cloud services [20]. However, these approaches do not account for the dynamicity of the underlying infrastructure (e.g., fluctuating performance, time-dependent infrastructure availability or failures). Moreover, they inherently assume that other elements in the environment are fixed. For example, programming a time-sensitive application capable of reacting to dynamic events (e.g., steering a self-driving car) is of little consequence if cloud resources modeling the car trajectory become unavailable (due to network outages) or if edge resources (on the car) are not capable of processing the camera feed at that point in time. Therefore, an effective solution must account for the requirements and dynamic behaviors of the individual elements of the system (i.e., users, service providers, application workflows, and computational services), as well as their impact on the overall environment (i.e., a solution that may benefit one may not necessarily be the best solution for the system as a whole).

Ph.D. Research. In my dissertation research, I adopted a three-pronged approach to address these challenges by: (1) enabling the on-demand aggregation of distributed computational services while facilitating the continuous deployment of representative scientific application workflows on top of them, (2) providing a programmable approach and a runtime framework that allows users, resource providers, and applications to dynamically compose different computational services, and (3) modeling the performance & expected QoS of the resulting environment.

- ***On-demand Federation.*** The goal of this research was to expose traditional resources (e.g., supercomputers, grids, clusters) as services using cloud abstractions [1] by building an abstraction layer that enables on-demand access, elasticity (scale up/down), and dynamic federation (scale out) of otherwise rigid resources. The resulting federated infrastructure was then used to run real-world scientific applications in various domains, e.g., oil reservoir history matching [1], molecular dynamics [2], and dissipative particle dynamics [3]. A demonstration using the framework was awarded first place in the IEEE SCALE Challenge at CCGrid'11. To further support complex workflows, I integrated the framework with Kepler, a workflow description language and enabled the deployment of a metagenomics workflow on top of dynamic infrastructure [9].
- ***Distributed Software-Defined Environment.*** Building on this on-demand federation, I developed a programmable approach to enable the dynamic composition of distributed services (both traditional resources and cloud infrastructure). This approach leverages concepts from software-defined environments to allow users, resource providers, and applications to programmatically control resource availability through predefined APIs [4]. I exposed the programmability of composing distributed services using two different declarative languages: (1) a rule-engine–based language that allows fine-grained control over services [7, 11] and (2) a Constraint Programming (CP) –based language that provides global control over infrastructure services [8, 10, 11]. This approach goes beyond static matchmaking techniques, such as Condor HTC [21], by enabling dynamic directives that are expressed as a function of runtime variables, which in turn allows the

dynamic composition of services along the data path (i.e., edge, in-transit, and core services). To demonstrate the flexibility and extensibility of the CP-based approach, I integrated the framework with Docker containers in order to facilitate the deployment of containers across multiple clouds [8]. The work, which was the first of its kind, was awarded first place in the IEEE/ACM UCC'15 Challenge and featured in Fortune Magazine[1]. The associated runtime created a distributed software-defined environment that evolves over the application lifecycle, by adapting in near real-time to changes in (1) infrastructure properties or availabilities; (2) user, application, or resource provider requirements regulating resource usage; (3) application workload; or (4) QoS optimization objectives imposed on the system.

- **Performance Modeling and QoS Quantification.** I worked on modeling the performance and evaluating the expected QoS of the resulting environment [13]. In particular, I developed mathematical models to quantify various aspects of the environment, which allows users to reason about trade-offs and requirements with respect to throughput, cost, deadline. For example, this work estimated the overall Service-Level Agreement (SLA) based on the combined SLAs from different providers, by calculating the expected application throughput, budget, or time to run an application given the current infrastructure composition. This work also modeled the tradeoffs between different scheduling objectives and helped identify bottlenecks in the current composition (i.e., do we need more resources or network bandwidth to increase application throughput).

Postdoctoral Research. In my postdoctoral research, I focused on the ways in which my work toward an autonomous computing infrastructure can benefit embedded systems and real-time IoT applications. For this class of applications, the composition of resources and services varies significantly because of the need for near real-time data insights and integration of streaming data from a variety of sources at the edge and in the cloud. To this end, I worked on an open source large-scale distributed operating system for smart buildings[2], which provides real-time monitoring of building sensors and aims to control the building in response to Demand Response events in the electrical grid. I helped manage the deployment of the system in 20 medium and large commercial buildings across California and assisted in making the data available for public research [12]. Further, I worked on developing ML models that simulate the thermal response of a building (i.e., a digital twin). The goal of this work [16] is to study the computational requirements and applicability of Deep RL in real-world scenario, e.g., the coordination of smart vehicle charging and HVAC systems control in smart buildings, in order to improve overall energy efficiency and reduce cost, compared to classical control techniques such as Model Predictive Control.

In the context of cloud-edge orchestration, I am currently investigating the efficacy of the serverless abstraction for executing workloads across hybrid cloud/edge infrastructure. In particular, I developed a prototype framework that provides a serverless execution hierarchy across the cloud-to-edge continuum and allows clients to explicitly invoke functions in any tier [17]. I am currently developing a decision engine for the placement of serverless functions across the continuum according to different QoS objectives. Finally, I am leading a survey effort on the emerging computing landscape (e.g., edge and fog computing) and quantifying the systems challenges and tradeoffs in all tiers of distributed computing (cloud, middle, edge, and extreme edge resources) [18].

Future Research. An autonomous computing infrastructure requires combining heterogeneous, complex, and loosely connected data and computing resources in order to process data *in situ* at the edge and *in transit* along the data path. Further, in order to support real-time applications, we must consider not only the dynamic and on-demand composition of services but also near-optimal methods of composing them when the behaviors of both the infrastructure and the workloads are highly volatile and uncertain. For example, can we dynamically adjust the application accuracy in cases where resources are limited? Can we anticipate the application workload or the future state of the underlying infrastructure? Can we use this information to optimize different objectives (e.g., maintain data locality, low response time, or minimize cost)? I plan on addressing these questions by exploring the following four research directions:

1. Investigate application malleability and bidirectional negotiation between an application and the underlying infrastructure. This allows adjusting the infrastructure to meet application needs as well as adjusting the

[1] http://fortune.com/2015/08/27/ibm-deploys-containers-across-clouds/

[2] http://docs.xbos.io/

application when resources are limited. This can be achieved by exposing application requirements in domain-specific metrics with different QoS tradeoffs. For example, users can define different levels of accuracy that corresponds to different price points and execution deadlines. These metrics can be controlled during runtime and translated to specific resource requirements based on the currently available resources.

2. Model and predict application behavior (workload size, data locations and generation/access patterns) and the expected behavior (properties, availabilities, failures, SLA guarantees) of different infrastructure services (compute, network, storage). This can be achieved by using historical information (e.g., resource metrics, application traces) to develop statistical ML models. The predicted information can then be used to dynamically configure the computing fabric available to applications (e.g., allocate more services, pre-fetch data, etc.)
3. Find near-optimal resource compositions that can continuously evolve based on historical actions while taking into consideration the complexity of resource management and the overall uncertainty in applications and infrastructure behavior. Deep RL techniques provide a viable solution, which when combined with the ML predictions and the programmable approach of composing services, can provide a truly autonomous, self-adaptive, and self-optimized computing infrastructure.
4. Democratize access to the autonomous infrastructure and facilitate its usability by average application developers. This can be achieved by leveraging a microservice architecture, a serverless computing paradigm, and a declarative language that specifies application dependencies, data flow, and operational objectives and constraints. This information can then be translated at runtime to dynamically modify the infrastructure composition and the application deployment on top of it.

Other Research Interests

Harvesting idle data center resources for science and education[3]: During my graduate studies, I developed a prototype for harvesting idle resource cycles in large commercial data centers, allowing them to be exposed as cloud services to support scientific applications. Companies that donate these resources can then claim these donations as tax deductions. I am interested in developing the prototype further and using it to support science and engineering applications as well as using these resources for educational purposes (e.g., to run Jupyter notebook for CS or Data Science classes).

Decentralized authentication and blockchains: During my postdoctoral research, I contributed to the development of a decentralized authentication verification engine, WAVE [14]. In particular, I have worked on building a global storage for permissions using AWS S3 and lambda. I have also worked on improving graph building for proof verification. I am interested in leveraging WAVE to manage secure content that is hosted outside of the owner's domain, i.e., content distributed using a Content Delivery Network (CDN), where a user has no control over the underlying infrastructure and cannot enforce any permissions. I have also studied the design space and trade-offs in Blockchains and helped write a survey and tutorial for general CS audience on blockchains and their design, limitations, and challenges [15].

Systems tools for democracy and the role of ethics in Computer Science: Large-scale distributed systems have evolved significantly over the past few decades and are now at a point where they can create new economies (e.g., sharing economy), inflict bias (e.g., AI-based recommendation systems), or affect the democratic process (e.g., Arab Spring, Brexit, US elections). Consequently, it is imperative for researchers to both study and teach the role of Computer Science in a rapidly moving world. Systems engineers are no longer building value-neutral tools that can improve everyone's life. Instead, we must research the societal impact of said tools. I am very interested in the role of distributed systems in improving governance, supporting democracy, and providing means for people to participate in the democratic process.

[3] Joint work with Johannes Watzl and Manish Parashar. "Idle Datacenter Resource Donation," Provisional Patent Filed, Full Patent Pending.

References

1. 2012 - **AbdelBaky, M.**, Parashar, M., Kim, H., Jordan, K.E., Sachdeva, V., Sexton, J., Jamjoom, H., Shae, Z.Y., Pencheva, G., Tavakoli, R. and Wheeler, M.F. Enabling high-performance computing as a service. **IEEE Computer**
2. 2013 - Parashar, M., **AbdelBaky, M.**, Rodero, I. and Devarakonda, A. Cloud paradigms and practices for computational and data-enabled science and engineering. **IEEE Computing in Science & Engineering**
3. 2014 - **AbdelBaky, M.**, Diaz-Montes, J., Johnston, M., Sachdeva, V., Anderson, R.L., Jordan, K.E. and Parashar, M. Exploring HPC-based scientific software as a service using CometCloud. IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing. **CollaborateCom'14**
4. 2014 - Diaz-Montes, J., **AbdelBaky, M.**, Zou, M. and Parashar, M. Software defined federated cyber-infrastructure for science and engineering. The ACM international workshop on Software-defined ecosystems. **HPDC'14**
5. 2015 - Diaz-Montes, J., **AbdelBaky, M.**, Zou, M. and Parashar, M. CometCloud: Enabling software-defined federations for end-to-end application workflows. **IEEE Internet Computing**
6. 2015 - Parashar, M., **AbdelBaky, M.**, Zou, M., Zamani, A.R. and Diaz-Montes, J. Realizing the Potential of IoT Using Software-Defined Ecosystems. IEEE 8th International Conference on Cloud Computing. **IEEE Cloud'15**
7. 2015 - **AbdelBaky, M.**, Diaz-Montes, J., Zou, M. and Parashar, M. A framework for realizing software-defined federations for scientific workflows. The 2nd International Workshop on Software-Defined Ecosystems. **HPDC'15**
8. 2015 - **AbdelBaky, M.**, Diaz-Montes, J., Parashar, M., Unuvar, M. and Steinder, M. Docker containers across multiple clouds and data centers. IEEE/ACM 8th International Conference on Utility and Cloud Computing. **UCC'15**
9. 2016 - Wang, J., **AbdelBaky, M.**, Diaz-Montes, J., Purawat, S., Parashar, M., & Altintas, I. Kepler+ CometCloud: dynamic scientific workflow execution on federated cloud resources. **Procedia Computer Science**
10. 2017 - **AbdelBaky, M.**, Diaz-Montes, J., Unuvar, M., Romanus, M., Steinder, M., Rodero, I. and Parashar, M., 2017. Enabling Distributed Software-Defined Environments Using Dynamic Infrastructure Service Composition. The 2017 IEEE/ACM 17th International Symposium on Cluster, Cloud and Grid Computing. **CCGrid'17**
11. 2017 - **AbdelBaky, M.**, Diaz-Montes, J. and Parashar, M. Software-Defined Environments for Science & Engineering. **The International Journal of High Performance Computing Applications**
12. 2018 - Fierro, G.T., Pritoni, M., **AbdelBaky, M.**, Raftery, P., Peffer, T., Thomson, G., and Culler, D.E. Mortar: An Open Testbed for Portable Building Analytics. The 5th ACM International Conference on Systems for Built Environments. **BuildSys'18**
13. 2019 - **AbdelBaky, M.** and Parashar, M. A General Performance and QoS Model for Distributed Software- Defined Environments. **IEEE Transactions on Services Computing**
14. 2019 - Andersen, M.P., Kumar, S., **AbdelBaky, M.**, Fierro, G.T. , Kolb, J., Kim, H.S., Culler, D.E., and Popa, R.A., WAVE: A Decentralized Authorization Framework with Transitive Delegation. The 28th USENIX Security Symposium. **USENIX Security'19**
15. 2020 - Kolb, J., **AbdelBaky, M.**, Katz, R.H., and Culler D.E., Core Concepts, Challenges, and Future Directions in Blockchain: A Centralized Tutorial. ACM Computing Surveys. **ACM CSUR**
16. **AbdelBaky, M.**, Lengyel, D., Fierro, G.T., Prakash, A.K., Clark, C., Gupta, P., Spangher, L., Panagopoulos, A.A., Pritoni, M., Peffer, T., and Culler, D.E., Reinforcement Learning for Smart Building Control. In prep for submission
17. AbdelBaky, M., Fierro, G.T., Sreekanti, V., Kolb, J., Gonzalez, J.E., Hellerstein, J.M., and Culler D.E., Serverless Computing Across Hybrid Cloud-Edge Infrastructure. In prep for submission.
18. **AbdelBaky, M.**, Sreekanti, V., Kolb, J., Yadwadkar, N., Kim, H.S., Culler D.E., Gonzalez, J.E., Hellerstein, J., Stoica, I., Katz, R.H., and Patterson, D. Cloud vs. Edge: A Berkeley View of Systems Challenges in Emerging Computing Utilities. In prep for submission.
19. Yu, J., & Buyya, R. A taxonomy of workflow management systems for grid computing. Journal of Grid Computing, 3(3-4), 171-200.
20. Zhan, Z. H., Liu, X. F., Gong, Y. J., Zhang, J., Chung, H. S. H., & Li, Y. (2015). Cloud computing resource scheduling and a survey of its evolutionary approaches. ACM Computing Surveys (CSUR), 47(4), 63.
21. Raman, R., Livny, M., and Solomon, M. Matchmaking: Distributed resource management for high throughput computing. HPDC.