An Autonomous Computing Infrastructure

Motivation. 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.

Research Background & Vision. In my Ph.D., I developed a middleware that enables the dynamic aggregation of distributed heterogeneous computing infrastructure. I leveraged both a rule-based and a constraint programming-based declarative languages to manage resource coordination in near real-time. The resulting framework enabled large-scale science in multiple domains and enabled the deployment of Docker containers across multiple clouds.

My research aims to address the needs of emerging data-driven real-time applications (e.g., industrial robotics) by creating an adaptive computing infrastructure, backed by a high-level declarative language and advanced machine learning techniques, that combines the capabilities of distributed infrastructure. My overall vision is to build a system that can seamlessly combine distributed cloud-edge compute, storage, network, and data sources without the need for human intervention (i.e., autonomous). Such an autonomous system would enable the composition of infrastructure in a (1) dynamic/adaptive, (2) programmable, (3), customizable/flexible, and (4) self-optimizing manner. This work is of particular applicability to emerging dynamic and data-driven applications, such as those driven by the Internet of Things (IoT), Artificial Intelligence (AI), or Computational and Data-Enabled Science and Engineering (CDS&E) applications, where data sources, types, rates, and sizes are dynamic and unpredictable.

Related Work & Research Challenges. Current approaches to managing dynamic applications often target one aspect of the complex environment, namely either by scheduling applications while optimizing user/resource providers' Quality of Service (QoS) objectives [20] or exploiting elasticity of cloud services [21]. 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 on-demand aggregation of distributed 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 and expected QoS of the resulting distributed software-defined environment. In this regard, this work is being used by many world-renowned institutions including Rutgers Cancer Institute, IBM T.J. Watson Research Center, the University of Texas at Austin, and San Diego Supercomputing Center at the University of California, San Diego.

1. 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].

2. Programmable & Adaptive 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 services). 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 [8, 10] and (2) a constraint-programming–based language that provides global control over services [6, 10, 11, 12]. This approach goes beyond static matchmaking techniques, such as Condor HTC [22], 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 [6]. 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 [12] that evolves over the application lifecycle, especially in a latency-sensitive environments, by adapting in near real-time to changes in (1) infrastructure behavior, properties, or availabilities; (2) user, application, or resource provider requirements regulating resource usage; (3) application behavior or workload; or (4) QoS optimization objectives imposed on the system.

3. Performance Modeling and QoS Quantification

Finally, I worked on modeling the performance and evaluating the expected QoS of the resulting distributed software-defined environment [13]. The overall goal of this phase was to quantify the various aspects of the environment, which allows users to reason about trade-offs and requirements with respect to throughput, cost, deadline, etc. This research aimed to answer questions such as (1) What is the expected Service-Level Agreement (SLA) of the environment based on the combined SLAs from different providers? For example, what is the expected application throughput? What is the estimated budget or deadline to run a certain workflow given the current composition of services? (2) What are the tradeoffs between different scheduling objectives? and (3) Can we identify bottlenecks in the current composition? For example, do we need more resources or more network bandwidth to increase application throughput?

Impact. My dissertation research enables the dynamic, on-demand, and programmable composition of distributed infrastructure services, which can facilitate many scenarios in science, engineering, IoT, and enterprise. For example, my research can prevent vendor lock-in by allowing users to cloudburst their applications while taking advantage of different services and prices from different cloud providers [6]. It can also allow 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 [10]. Further, this work enables the composition of distributed services 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). Finally, this work can support emerging data-driven applications [5, 7] by combining resources across edge-cloud 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.

---

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

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 deploy the system in 20 medium/large commercial buildings across California and assisted in making the data available for public research [14]. Further, I am currently developing ML models that simulate the thermal response of a building (i.e., a digital twin). The goal of this work [15] is to study the computational requirements and potential impact of Deep Reinforcement Learning (Deep RL) when applied to a specific 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 developed a micro-service architecture for the ML models and controls required for managing smart buildings. I am currently developing a decision engine for the placement of these micro-services in a hybrid cloud-edge environment. Finally, I am leading a survey effort of the emerging computing landscape (e.g., Fog computing) and quantifying the systems challenges and tradeoffs in all tiers of distributed computing (cloud, middle, edge, and extreme edge resources) [16].

Future Research. My overarching goal is to build an autonomous computing infrastructure, which spans cloud-edge services, uses a declarative language to specify the overall desired system QoS, and leverages ML/RL techniques to continuously operate based on application runtime behavior and despite failures or changes in the underlying infrastructure. This is essential for real-time applications where continuous operation is required (e.g., self-driving cars, industrial robots, wildfire monitoring, etc.). This system 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 these 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/structure of the underlying infrastructure? Can we use this information to optimize different objectives (e.g., maintain data locality, real-time processing, 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 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/resolution 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 the application behavior (workload size, data requirements, data locations, data generation/access patterns) and the expected behavior (properties, availabilities, 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 data-driven 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 infrastructure and facilitate its usability by average application developers.

---

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

Other Research Interests

Harvesting idle data center resources for science and education[3]: During my graduate studies, I developed a prototype for monetizing and 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 [17]. 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 [18]. 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 [19].

Serverless computing[4]: During my postdoctoral research, I have also explored Serverless computing and its role in facilitating application deployment in an autonomous computing infrastructure. In particular, I am currently studying a systematic way to model and reverse-engineer the container scheduling and management algorithms underlying serverless computing platforms. I am also interested in developing a framework that uses AWS lambda to improve Amazon's S3 data consistency model.

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 academics 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.

References

1. **AbdelBaky, M.**, Parashar, M., Jordan, K., Kim, H., Jamjoom, H., Shae, Z.Y., Pencheva, G., Sachdeva, V., Sexton, J., Wheeler, M. and Wheeler, M.F., 2012. Enabling high-performance computing as a service. **Computer**, 45(10), pp.72-80.
2. Parashar, M., **AbdelBaky, M.**, Rodero, I. and Devarakonda, A., 2013. Cloud paradigms and practices for computational and data-enabled science and engineering. **Computing in Science & Engineering**, 15(4), pp.10-18.
3. **AbdelBaky, M.**, Diaz-Montes, J., Johnston, M., Sachdeva, V., Anderson, R.L., Jordan, K.E. and Parashar, M., 2014, October. Exploring HPC-based scientific software as a service using CometCloud. In Collaborative Computing: Networking, Applications and Work sharing (**CollaborateCom**), 2014 International Conference on (pp. 35-44). IEEE.
4. Diaz-Montes, J., **AbdelBaky, M.**, Zou, M. and Parashar, M., 2014, June. Software defined federated cyber-infrastructure for science and engineering. In Proceedings of the 2014 ACM international workshop on Software-defined ecosystems (pp. 9-12). **HPDC**.
5. Diaz-Montes, J., **AbdelBaky, M.**, Zou, M. and Parashar, M., 2015. CometCloud: Enabling software-defined federations for end-to-end application workflows. **IEEE Internet Computing**, 19(1), pp.69-73.
6. **AbdelBaky, M.**, Diaz-Montes, J., Parashar, M., Unuvar, M. and Steinder, M., 2015, December. Docker containers across multiple clouds and data centers. In 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (**UCC**) (pp. 368-371). IEEE.

---

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

[4] Joint work with Jack Kolb and David E. Culler

7. Parashar, M., **AbdelBaky, M.**, Zou, M., Zamani, A.R. and Diaz-Montes, J., 2015, June. Realizing the Potential of IoT Using Software-Defined Ecosystems. In 2015 IEEE 8th International Conference on Cloud Computing (pp. 1149-1158). **IEEE Cloud**.
8. **AbdelBaky, M.**, Diaz-Montes, J., Zou, M. and Parashar, M., 2015, June. A framework for realizing software-defined federations for scientific workflows. In Proceedings of the 2nd International Workshop on Software-Defined Ecosystems (pp. 7-14). **HPDC**.
9. Wang, J., **AbdelBaky, M.**, Diaz-Montes, J., Purawat, S., Parashar, M., & Altintas, I. (2016). Kepler+ CometCloud: dynamic scientific workflow execution on federated cloud resources. Procedia Computer Science, 80, 700-711.
10. **AbdelBaky, M.**, Diaz-Montes, J. and Parashar, M., 2017. Software-Defined Environments for Science & Engineering. **The International Journal of High Performance Computing Applications**, 1094342017710706.
11. **AbdelBaky, M.**, Diaz-Montes, J. and Parashar, M., 2017. Towards Distributed Software-Defined Environments. In the 2017 IEEE/ACM 17th International Symposium on Cluster, Cloud and Grid Computing (**CCGrid**).
12. **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. In the 2017 IEEE/ACM 17th International Symposium on Cluster, Cloud and Grid Computing (**CCGrid**).
13. **AbdelBaky, M.** and Parashar, M., 2018. A General Performance and QoS Model for Distributed Software- Defined Environments. Submitted to **IEEE Transactions on Services Computing**.
14. Fierro, G.T., Pritoni, M., **AbdelBaky, M.**, Raftery, P., Peffer, T., Thomson, G., and Culler, D.E., 2018, November. Mortar: An Open Testbed for Portable Building Analytics. The 5th ACM International Conference on Systems for Built Environments (**BuildSys'18**), China.
15. **AbdelBaky, M.**, Lengyel, D., Fierro, G.T. , Panagopoulos, A.A., Pritoni, M., Gonzalez, J.E., and Culler D.E., Reinforcement Learning for Smart Building Control. In prep for submission to the 6th ACM International Conference on Systems for Built Environments (**BuildSys'19**).
16. **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 to **IEEE Communications Surveys & Tutorials**.
17. 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. In prep for submission to USENIX Symposium on Security (**USENIX Security'19**).
18. **AbdelBaky, M.**, Andersen, M.P., and Culler D.E., A Practical Subset Sum. In prep for submission to ACM-SIAM Symposium on Discrete Algorithms (**SODA'19**)
19. Kolb, J., **AbdelBaky, M.**, and Culler D.E., Demystifying Blockchain. In prep for submission to **IEEE Communications Surveys & Tutorials**.
20. Yu, J., & Buyya, R. (2005). A taxonomy of workflow management systems for grid computing. Journal of Grid Computing, 3(3-4), 171-200.
21. 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.
22. Raman, R., Livny, M., and Solomon, M. Matchmaking: Distributed resource management for high throughput computing. HPDC.