Statement of Research Interests

Kaoutar El Maghraoui

Even though advances in hardware technologies are constantly pushing the limits of computational and storage resources, there are always applications whose computational demands exceed even the fastest technologies available. Hence, it has become critical to look into ways to aggregate efficiently distributed resources to benefit a single application. Achieving this vision requires being able to run applications on dynamic and heterogeneous environments such as grids and shared clusters. New challenges emerge in such environments, where performance variability is the rule and not the exception. The complexity lies not only in how to optimize large applications to run on a given set of distributed resources, but also in how to maintain the desired performance when the pool of resources may change anytime, the characteristics of the resources are variable, and the application's demands vary during its life-time. Therefore, applications require the ability to dynamically reconfigure to adjust to the dynamics of the underlying resources.

My primary research interest is facilitating the way in which scientific and engineering applications interact with dynamic execution environments and reconfigure themselves as needed. My work focuses on exploring methods and developing tools to allow for middleware-driven dynamic application reconfiguration in grid environments, with the goal of allowing applications to fully exploit the potential of such heterogeneous and rapidly evolving environments and facilitating this task to applications' developers. I am also highly interested in interdisciplinary research, which bridges between computer science and disciplines in need for high-performance computing resources. Interdisciplinary research has revolutionized the way science is being done today and has opened up a vast range of opportunities for solving complex problems that scientists could not even think about solving a decade ago.

Middleware-driven Application Reconfiguration

Dynamic reconfiguration implies the ability to modify the mapping between application entities and physical resources (migration) and/or to modify the application entities' granularity (malleability) while the application continues to operate without any disruption of service. Dealing with reconfiguration issues over large-scale dynamic environments is beyond the scope of application developers. It is therefore imperative to adopt a middleware-driven approach. My research focuses on investigating both process migration and malleability as dynamic application reconfiguration mechanisms.

We proposed and prototyped the Internet Operating System (IOS) middleware [5]. The goal of the Internet Operating System (IOS) project is to provide the necessary tools for application reconfiguration mechanisms in dynamic execution environments. Our approach to dynamic reconfiguration is twofold. On the one hand, the middleware layer is responsible for resource discovery, monitoring of application-level and resource-level performance, and deciding when, what, and where to reconfigure applications. On the other hand, the application layer is responsible for dealing with the operational issues of migration and malleability and the profiling of application communication and computational patterns. IOS has been designed with the following key characteristics: 1) **Architecture Modularity** to allow for extensible and pluggable reconfiguration mechanisms to accommodate various execution environments and different applications, 2) **Generic Interfaces** to allow various programming models and technologies to leverage IOS dynamic reconfiguration mechanisms, 3) **Decentralized Strategies** to achieve scalable decisions, and 4) **Dynamic Resource Availability** to allow applications to adjust their resource allocations following the availability of the resources. IOS is implemented in Java and SALSA [10], an actor-oriented programming language. IOS agents leverage the autonomous nature of the actor model and use several coordination constructs and asynchronous message passing provided by the SALSA language.

IOS is being used as an experimental testbed for two multi-disciplinary scientific applications. The first application is in the area of astrophysics with the goal of analyzing data from the Sloan Digital Sky Survey to find structure in the outer layers of the Milky Way. The second one is a nuclear physics' application that performs partial wave analysis.

IOS embodies resource profiling and reconfiguration decisions into software agents. IOS agents are capable of organizing themselves into various virtual topologies. I investigated two representative virtual topologies for inter-agent coordination: a peer-to-peer and a cluster-to-cluster coordination topology [3, 6]. The coordination topology of IOS agents has a great impact on how fast reconfiguration decisions are made. In a more structured environment such as a grid of homogeneous clusters, a hierarchical topology generally performs better than a purely peer-to-peer topology [6].

To determine how reconfiguration should happen, we developed two reconfiguration strategies: random work stealing (RS) and application topology sensitive work stealing (ATS) [1]. In the RS strategy, nodes with light computation and communication loads attempt to steal work from heavily loaded nodes. The ATS strategy is an extension of RS. It takes into account the application's communication topology. ATS collocates highly synchronized application's entities within nodes with low latency interconnections. The ATS reconfiguration decision relies on a mathematical model that predicts when it is beneficial to perform a given reconfiguration such as a split, merge, or migrate. Based on heuristics, the model determines the group of entities that need to be reconfigured in such ways to minimize communication latencies between entities.

Existing approaches to application reconfiguration have mainly focused on migration mechanisms at a fine granularity (e.g process-level) or a coarse granularity (e.g application-level). I investigated another novel reconfiguration tool that allows dynamic granularity of application's processes or *malleability*. In dynamic environments, application's reconfiguration using migration is further limited by the process granularity [2]. Therefore, a reconfiguration strategy that combines both migration and malleability is more effective and flexible.

Reconfiguring Message Passing Applications

Part of my PhD research consists of providing the necessary library tools to make Message Passing (MPI) applications reconfigurable. I designed and implemented the Process Checkpointing and Migration (PCM) library [9]. PCM [9, 7] is a user-level library that provides checkpointing, profiling, and migration capabilities for a large class of iterative applications. Programmers need only specify the data structures that must be saved and restored to allow process migration. PCM also provides process split and merge functionalities to MPI programs [4]. Common data distributions are supported like block, cyclic, and block-cyclic. PCM implements IOS generic profiling and reconfiguration interfaces, and therefore enables MPI applications to benefit from IOS reconfiguration policies.

The PCM API is simple to use and hides many of the intrinsic details of how to perform reconfiguration through migration, split and merge. Hence, with minimal code modification, a PCM-instrumented MPI application becomes malleable and ready to be reconfigured transparently by the IOS middleware. In addition, legacy MPI applications can benefit tremendously from the reconfiguration features of IOS by simply inserting a few calls to the PCM API. I have validated the practical implementation of the PCM library on two realistic applications: a fluid-dynamic application that models heat diffusion over a solid and an astronomical application that uses linear regression techniques to fit models to the observed shape of the galaxy. The latter application operates on data collected from the SLOAN Digital Sky Survey. Both these applications are iterative and exhibit various degrees of communication to computation ratio. They are representative of a large number of scientific applications.

Model-driven Application Deployment using AI Planning

Beside my thesis research, I investigated the use of AI planning for the automatic provisioning of distributed applications in data centers [8] as part of two summer internships at IBM T.J. Watson Research Center. I designed and implemented a prototype for the automatic generation of provisioning actions (workflows) in a data center. The approach relied on using the partial-order planning algorithm, POP to infer the partial order of provisioning operations and their inputs to deploy a given application in a data center. Several heuristics were introduced to improve the performance of POP in the provisioning domain. The prototype has been integrated with IBM's Rainforest project and was capable of generating deployment plans that can be executed using IBM's Tivoli Intelligent Orchestrators (TIO) provisioning manager.

Future Work

I intend to continue working in the area of application dynamic reconfiguration in grid environments. I am also interested in exploring scientific and engineering multi-scale applications that may benefit from dynamic grid infrastructures. There are still a lot of unresolved issues that I would like to vividly investigate.

Replication as another reconfiguration strategy: My PhD research investigates mainly process migration, split and merge as possible ways of reconfiguring applications in dynamic grid environments. Another natural direction is to use process or data replication as another reconfiguration strategy. Replication strategies can improve the performance, availability, and fault-tolerance of applications. I am interested in developing new replication reconfiguration strategies and integrating them to the IOS middleware.

Statistical and data mining methods for scalable profiling and measurements: Application-level and resource-level profiling are key to determining how to adapt applications dynamically to the constantly changing resources of dynamic grids. However, profiling imposes a performance penalty on applications. Also communicating large amounts of profiled data to the middleware can consume a large amount of computational and communication resources. I plan to investigate statistical and data mining methods that extract useful patterns from the gathered profiled information and approximate global knowledge for less intrusive profiling and more scalable profiling techniques.

Graph clustering for resource optimization: When new resources join or existing resources leave, middleware services need to collaborate with the application to reconfigure it appropriately to utilize the new available resources or migrate away from the slow or leaving resources. In many instances, it makes more sense to perform collective reconfiguration of a group of application's entities. Clustering the application communication graph can help determine potential groups for reconfiguration. I am interested in applying graph clustering techniques to the application graph. In a largely distributed environment that relies on decentralized coordination techniques, having access to the full graph of the application is not feasible. Clustering techniques have mainly been studied on full graphs. I plan to develop new algorithms for partial graph clustering and study their composition.

Automatic programming for grid applications: Automatic programming is a future direction for developing grid applications. I intend to utilize code analysis techniques and automatic code generation to transparently modify existing applications into reconfigurable ones. Another interesting research direction is to come up with high level specification or scripting languages that allow developers to outline the functionality needed in high-level terms. Then, automatic programming tools will translate such descriptions into specialized grid-aware and reconfigurable code. This will ease the development and deployment of applications in grid environments. I plan to investigate and apply such techniques to MPI applications and Unified Parallel C applications.

Interdisciplinary Research: Last but not least, it is important to apply and validate the devised strategies on realistic large-scale applications in production grid environments such as the TeraGrid facility. I am interested in collaborating with other researchers from compilers, systems, and applications groups and researchers from other disciplines such as biology and physics to pursue the research directions I outlined above.

References

[1] T. Desell, K. E. Maghraoui, and C. Varela. Load balancing of autonomous actors over dynamic networks. In *Proceedings of the Hawaii International Conference on System Sciences, HICSS-37 Software Technology Track*, pages 1–10, January 2004.

[2] T. Desell, K. E. Maghraoui, and C. Varela. Malleable Components for Scalable High Performance Computing . In *Proceedings of the HPDC'15 Workshop on HPC Grid programming Environments and Components (HPC-GECO/CompFrame)*, pages 37–44, Paris, France, June 2006. IEEE Computer Society.

[3] K. E. Maghraoui, T. Desell, B. K. Szymanski, J. D. Teresco, and C. A. Varela. Towards a middleware framework for dynamically reconfigurable scientific computing. In L. Grandinetti, editor, *Grid Computing and New Frontiers of High Performance Processing*, volume 14 of *Advances in Parallel Computing*, pages 275–301. Elsevier, 2005.

[4] K. E. Maghraoui, T. J. Desell, B. K. Szymanski, and C. A. Varela. Dynamic malleability in MPI applications. Technical Report 06-17, Rensselaer Polytechnic Institute Department of Computer Science, 2006. Submitted to CCGrid 2007.

[5] K. E. Maghraoui, T. J. Desell, B. K. Szymanski, and C. A. Varela. The Internet Operating System: Middleware for adaptive distributed computing. *International Journal of High Performance Computing Applications (IJHPCA), Special Issue on Scheduling Techniques for Large-Scale Distributed Platforms*, 20(4):467–480, 2006.

[6] K. E. Maghraoui, T. J. Desell, and C. A. Varela. Network sensitive reconfiguration of distributed applications. Technical Report 05-03, Rensselaer Polytechnic Institute Department of Computer Science, 2005.

[7] K. E. Maghraoui, J. Flaherty, B. Szymanski, J. Teresco, and C. Varela. Adaptive computation over dynamic and heterogeneous networks. In *Proc. of the Fifth International Conference on Parallel Processing and Applied Mathematics (PPAM'2003)*, number 3019 in LNCS, pages 1083–1090, Czestochowa, Poland, September 2003.

[8] K. E. Maghraoui, A. Meghranjani, T. Eilam, M. Kalantar, and A. V. Konstantinou. Model driven provisioning: Bridging the gap between declarative object models and procedural provisioning tools. In *Proceedings of Middleware 2006, ACM/IFIP/USENIX, 7th International Middleware Conference, Melbourne, Australia*, November 2006.

[9] K. E. Maghraoui, B. Szymanski, and C. Varela. An architecture for reconfigurable iterative MPI applications in dynamic environments. In R. Wyrzykowski, J. Dongarra, N. Meyer, and J. Wasniewski, editors, *Proc. of the Sixth International Conference on Parallel Processing and Applied Mathematics (PPAM'2005)*, number 3911 in LNCS, pages 258–271, Poznan, Poland, September 2005.

[10] C. Varela and G. Agha. Programming dynamically reconfigurable open systems with SALSA. *ACM SIGPLAN Notices. OOPSLA'2001 Intriguing Technology Track Proceedings*, 36(12):20–34, Dec. 2001.