Semana de Engenharia 2010
Guimarães, 11 a 15 de Outubro

CIVIL PROTECTION APPLICATIONS IN A GRID SUPPORTED ENVIRONMENT

Vítor S. P. Oliveira
CCTC - Centro de Ciências e Tecnologias de Computação
vspo@di.uminho.pt

EXTENDED ABSTRACT

In the last decades there has been a significant trend towards the increase in the number and cost of natural catastrophes and man-made disasters [UN]. This increase in risk asks for adequate computing tools to simulate and control emergency situations in order to minimize casualties and environmental and property damages. Critical applications from Civil Protection (CP) demand increasing amounts of resources such as computing power, data storage, information services and knowledge systems, which are usually deployed in close integration with dedicated infrastructures.

The use of shared computing infrastructures for CP applications has the potential to significantly increase the resources available in emergency situations, while preventing the high cost and the single points of failure usually associated with dedicated infrastructures. But while computing paradigms such as the Grid or the Cloud offer access to a multitude of information repositories and to massive computing infrastructures, there are still important limitations for using them to support critical CP applications. In particular, there is little support for applications that present real-time requirements, require fault tolerance and workload survivability assurances, require access to spatial data infrastructures, strict security and data policies, require sensor network integration and require access to acquisition systems control and knowledge based services (Oliveira et al. 2009).

Several research projects extended the EGEE/EGI Grid middleware in areas related to the thesis theme (see Oliveira 2008). The CYCLOPS project suggested how existing Grid infrastructures can provide coordinated sharing of computing, storage and communication resources for the CP agents involved in the emergency management procedures. The GRIDCC/DORII projects addressed provisioning tools to manage instrumentation as Grid resources and to support the orchestration of multiple Grid resources in a timely workflow, the reservation agreements on Grid resources; and the facility to satisfy quality of service requirements on elements within workflows. The Int.EU.Grid dealt with providing multi-site interactivity and parallel computing in a Grid infrastructure. And the AssessGrid project addressed risk awareness and consideration in Service Level Agreement (SLA) negotiation and capacity planning to increase the transparency, reliability and trustworthiness of Grid activities. The CROSS-Fire project addresses decision support to control forest fires and forest fire simulations and presents the opportunity to explore a demanding CP application that can benefit both from Grid and from HPC infrastructures.

While those projects provided some support for CP applications, we believe that important dependability requirements were not fully addressed. Particularly, current strategy to have CP agencies and resource providers agree on a SLA and to dedicate resources for CP even in crisis situations to meet the contracted Quality of Service (QoS) (Djemame, 2008) but, notwithstanding the penalties, once a provider fails to fulfill the SLA after the deployment of the application the only alternative is to run the application again in another location. For critical applications that cannot be stopped after initiating or have a time limit for providing results the SLA approach is insufficient.

The research presented here focuses primarily on the dependability requirements of critical CP applications in emergency situations. Our approach aims at designing a new system layer that provides the dependability level applications require even if the underlying computing infrastructure has reliability failings. An application can establish a desired level of reliability that will be provided by 1) workload and application state replication and by 2) continuously monitoring the application evolution and relocating the application when needed to allow the system to run the application successfully even in the presence of multiple faults.
In our approach applications are regarded as a set of interconnected opaque virtual machines and the computing environment is a general purpose distributed system, to support not only the Grid in which this project initially originated, but also other computing infrastructures such as the HPC clusters and the Cloud.

It is presently possible to run two instances of an unmodified application in order to support the failure of one of the underlying computing nodes by running both instances in lockstep (Cully et al., 2008). This technique imposes no restrictions on how those applications interact with the rest of the world including – important for interactive user sessions, sensors control and access to multiple database systems – but there is little support for n-way replication, limited WAN synchronization

and workload placement and data storage must either be shared between replicas or completely replicated to each site. While replicating to more than one partner site requires few changes to current systems, efficient application replication to multiple partners over networks with high latencies typical of WAN environments and supporting network address migration for distant sites are challenges being addressed.

To efficiently access storage from multiple application workloads deployed in multiple sites it is important to minimize local state and to isolate the sites processing power from the application state. We are currently developing a DHT based file systems to be used in this architecture, simultaneously de-duplicating repeated data and generalizing local caches for higher throughput for both data and virtual machines access. This system tries to replicate the information as necessary with little overhead possible to set up replicas without incurring in a large performance penalty.

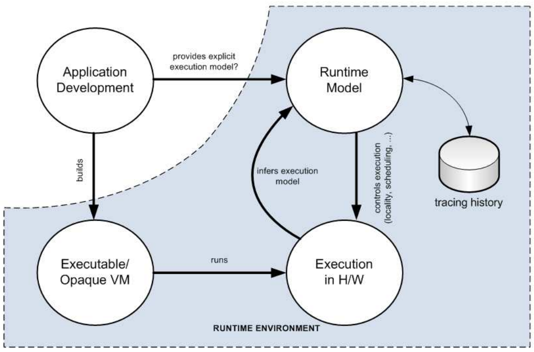

Figure 1: overview of the monitoring process

Given the unknown reliability of the infrastructures it is necessary to continuously monitor the application execution progress not only to optimize the workload placement in each moment, but also to estimate the remaining execution time for a particular application. While it is not possible to provide hard real-time guarantees, by observing the progress the system can make early decisions regarding resource management and the users expectations regarding the applications output availability can be adjusted. But monitoring of applications in runtime is very demanding, particularly if applications are not changed to provide information regarding their evolution. Our current research is working on the mechanisms required to process efficiently large volumes of log data gathered at runtime, taking into account the processor hardware support, application instrumenting, large volume log file structure and compression, program phase detection and program phase estimation (see Figure 1.).

The applications execution is analyzed in runtime to build an application model that captures the behavior of the executables, including the relations between tasks, between tasks and the outside world, between tasks and the storage and the memory usage pattern inside tasks. The observed behavior is then mapped to concepts such as domains, tasks, groups, variables, transitions, junctions, timelines and ports of the revisited CoR (Pina et al. 2002) computing model, while there is also the possibility that application libraries aware of the CoR computing model provide more directed information to the runtime model. The infrastructure concepts from the CoR model are transported to the runtime model, in order to facilitate the dynamic mapping between application requirements and the hardware resources available in each moment. The information about the applications behavior is gathered to fulfill the survivability requirements of critical applications by replicating the workloads with fault tolerant mechanisms in a WAN scenario. The historic and live tracing information are also used to optimize and dynamically adjust the execution of the application according to the runtime model in order to use the resources efficiently.

REFERENCES

Cully, B., Lefebvre, G., Meyer, D., Feeley, M., Hutchinson, M., Warfield, A., 2008, "Remus: high availability via asynchronous virtual machine replication", Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, p.161-174, San Francisco, California, April 16-18.

Djemame, K., Gourlay, I., Padgett, J., Voss, K., Kao, O., 2008, "Risk Management in Grids", in: Market-Oriented Grid Computing, R. Buyya and Kris.Bubendorfer (Editors), Wiley.

Oliveira, V., 2008, "Civil Protection Applications in a Grid Supported Environment", Pre-thesis proposal, MAP-i Doctoral Programme in Computer Science, Universidade do Minho.

Oliveira, V., Pina, A., Ayral, P., Thierion, V., Verlato, M., Mazzeti, P., Macedo, M., Oliveira, J., Pires, P., 2009, "D19 Final Plan for Using and Disseminating Knowledge", FP6 CYber-Infrastructure for CiviL protection Operative ProcedureS (CYCLOPS) project.

Pina, A., Oliveira, V., Moreira, C., Alves, A., 2002, "pCoR: a prototype for resource oriented computing", 7th International Conference on Applications of High-Performance Computers in Engineering (HPC '02), WIT Press, 251-262, Bologna, Italy, September.

AUTHOR BIOGRAPHY

VÍTOR S. P. OLIVEIRA is currently working on the PhD thesis "Civil Protection Applications in a Grid Supported Environment", supervised by Prof. António Pina.

He is enrolled in the MAP-I Doctoral Programme in Computer Science and presented his pre-thesis in July 2008.

This research is being supported by the FCT in partnership with the UT Austin | Portugal program (grant SFRH/BD/47840/2008, since October 2009).