Energy Aware Autonomic Manager

Damien Borgetto
IRIT
118 Rte de Narbonne
Toulouse, France
borgetto@irit.fr

Patricia Stolf
IRIT
118 Rte de Narbonne
Toulouse, France
patricia.stolf@irit.fr

Georges Da Costa
IRIT
118 Rte de Narbonne
Toulouse, France
dacosta@irit.fr

Jean-Marc Pierson
IRIT
118 Rte de Narbonne
Toulouse, France
pierson@irit.fr

Keywords
Energy efficiency, Virtual Machines, Autonomic Computing, Resource Allocation

1. CONTEXT
It is analyzed that the 1.5 billion computers in the world use about 90000 MW of electric power, which is about 10% of the global consumption.

In this work we will focus on large infrastructure such as large clusters, grids and even clouds. Large scale applications have wide needs, from failure management systems to energy-awareness.

To tackle the application management, we use the TUNe middleware which is an autonomous management system for distributed application, using schemas that act on reaction of events. As current version does not take into account energy, we use TUNe as a basic tool and we add generic schemas to reduce energy consumption of application by using an energy-aware placement algorithm which takes into account consolidation (to switch off nodes) and smart task mapping.

1.1 TUNe
TUNe is an autonomic manager prototype [3]. It can be used to deploy and configure proprietary applications in a distributed environment. It can also monitor the environment and react to events such as failures or overloads and reconfigure applications accordingly and autonomously. Within the TUNe framework, pieces of the legacy system are wrapped into Fractal components. This model provides uniform management interfaces that enable the implementation of complex administration tasks. This approach makes TUNe generic, since once legacy softwares have been wrapped, their administration can be based on the uniform component model. A second aspect of the TUNe prototype is the introduction of high level formalisms composed of :

- A Universal Modeling Language (UML) profile for specifying deployment schemes.
- A UML profile for specifying autonomous policies using activity charts.
- A wrapping description langage (WDL, [4]) to define a set of methods that can be invoked at runtime to configure or reconfigure the application.

1.2 EA-ResAlloc
The EA-ResAlloc[2] algorithm is an energy-aware resource allocation algorithm. This static algorithm allows one to allocate jobs in virtual machines in order to reduce energy consumption of heterogeneous large scale systems. It is used to improve an energy/performance tradeoff.

Two main concepts are used to allocate resources in an energy-efficient way. The first concept is that we try to place jobs where the power consumption it induces is low. The second is that jobs are placed in a way that will reduce the number of switched on hosts, as most of the power consumption of an host comes from the fact it is switched on.

2. ENERGY AWARE AUTONOMIC MANAGER
In this work we will integrate TUNe and a job scheduler. The idea here is to use the reconfiguration procedures of TUNe to react to energy related events. We also deploy application within virtual machines (VM) in order to be able to migrate them. We will do our experiments using the well known NAS Parallel Benchmarks (NPB) suite, by putting each NPB process in a VM. Physical machines are monitored and we assume we can switch on and off remotely hosts.

The system follows the autonomic loop: *Execution, Monitoring, Decision, Action.* TUNe is used as the ground middleware to manage the autonomic loop. We add specific

energy-related TUNe UML diagrams: class diagram, activity diagram. These diagrams use EA-ResAlloc to make mapping decisions. The goal is to reallocate periodically or when particular events are sent by sensors. Those reallocations can focus on a sub-set of the jobs or on all of them.

We implement sensor and define the events they send. For instance, some event types are *Host load threshold*, *Power threshold*, *Arrival/Departure of a job*.

An event can be sent by a specific sensor when the host's load is lower than 5%, thus we could be able to easily migrate the jobs executing on such a host, switching off the host.

At the decision center (DC), we will define action list given by the resource allocation algorithm. This means that the algorithm will calculate an allocation, and the difference between the present and the proposed allocation will result in a list of actions to achieve the new allocation.

The action list will be treated and each action will lead to an execution of the corresponding actuator. Each actuator has a specific action to do:

- migrate(job)
- shutdown(host)
- boot(host)
- DVFS(processor)

Each event received generates a call to an activity which is described in a UML activity diagram. For instance on a low loaded host, the sensor sends to the DC a *load threshold* event. The DC will parse the activity diagram which will make it call the EA-ResAlloc scheduler to calculate the reallocation, the necessary migration(s). The DC will enforce the migration(s) and then will send the order to the actuator to swich off the host. TUNe provides a plateform that simplify the description of such process.

The actual benchmark will be a NPB master and several NPB slaves, each in a different VM.

3. CONCLUSIONS AND PERSPECTIVES

In this poster we describe our approach to integrate a job scheduler and an autonomic manager. Our aim is to provide generic methods to reconfigure applications and their mapping to reduce energy consumption in large scale infrastructure composed of heterogeneous computers. We will achieve this reduction by smart taks mapping and by trying to free computers in order to switch them off. The autonomic tool is responsible of the application reconfiguration accordingly to the received events detected. We have presented a list of events related to energy consumption. We propose in TUNe generic reconfigurations to reduce energy consumption. During reconfigurations the scheduling algorithm is called to migrate tasks, to reschedule tasks in a list of hosts where some are shutdown...

In the literature we can find works on autonomic computing and on energy savings in distributed systems. In the field of autonomic computing, we find different autonomic systems. Two different approaches can be found : systems that incorporate autonomic mechanisms for problem determination, monitoring, analysis, management (like OceanStore [6], AutoAdmin [1], Q-Fabric [7]) and systems that investigate models, programming paradigms and development environments to support the development of autonomic systems and applications (like Kx(Kinesthetics eXtrem) [5], Autonomia [8]). TUNe is close to the second category.

Here we propose a generic approach to include in an autonomic tool an energy aware scheduling. We integrate the TUNe prototype with the EA-ResAlloc algorithm. This work is still in progress. Even if the approach is a generic one, the user can add specific reconfiguration methods for its application. Then in near future we will study different perspectives. First we will study other scheduling tool integration. Then, we will take into account ressource reservation tool and study their integration in the control loop of autonomic manager (we will focus on the OAR tool which is used on the GRID'5000 infrastructure).

4. REFERENCES

[1] AutoAdmin: Self-Tuning and Self-Administering Databases.

[2] D. Borgetto, G. Da Costa, J.-M. Pierson, and A. Sayah. Energy-Aware Resource Allocation. In *Energy Efficient Grids, Clouds and Clusters Workshop*. IEEE, 2009.

[3] E. Bruneton, T. Coupaye, M. Leclercq, V. Quéma, and J.-B. Stefani. The FRACTAL component model and its support in Java: Experiences with Auto-adaptive and Reconfigurable Systems. Software - Practice and Experience, special issue on Experiences with Auto-adaptive and Reconfigurable Systems, 36, 2006.

[4] D. Hagimont, P. Stolf, L. Broto, and N. Depalma. Component-based autonomic management for legacy software. In M. Denko, L. Yang, and Y. Zhang, editors, *Autonomic Computing and Networking*, pages 83–104. Springer, 2009.

[5] G. Kaiser, P. Gross, G. Kc, and J. Parekh. An approach to autonomizing legacy systems. In *in Workshop on Self-Healing, Adaptive and Self-MANaged Systems*, 2002.

[6] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao. OceanStore: an architecture for global-scale persistent storage. *SIGPLAN Not.*, 35(11):190–201, 2000.

[7] C. Poellabauer. *Q-fabric: system support for continuous online quality management*. PhD thesis, Atlanta, GA, USA, 2004. Adviser-Schwan, Karsten.

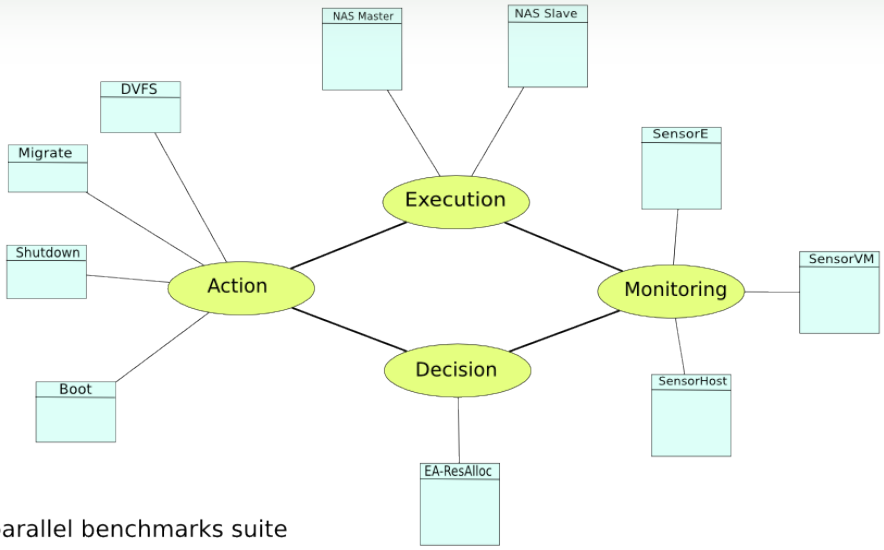
[8] H. C. M. Z. S. P. S. Hariri, L. Xue and S. Rao. Autonomia : An autonomic computing environment. 2003.

ENERGY-AWARE AUTONOMIC MANAGER

{borgetto,stolf,dacosta}@irit.fr

By using the TUNe middleware to provide us the ability to autonomically control our environment, we have developped a solution to reduce the energy consumption of a large scale sytem.

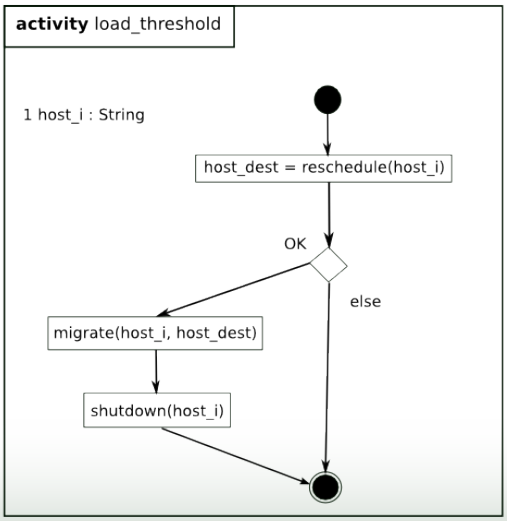
The autonomic manager TUNe allows us to plug-in an energy-aware resource allocation algorithm, thus modeling a complete generic autonomic energy reduction infrastructure.



- Execution of parallel applications : NAS parallel benchmarks suite
- One virtual machine per process
- Different granularity sensors : power, VM, host
- Behaviour modeling through activity diagram

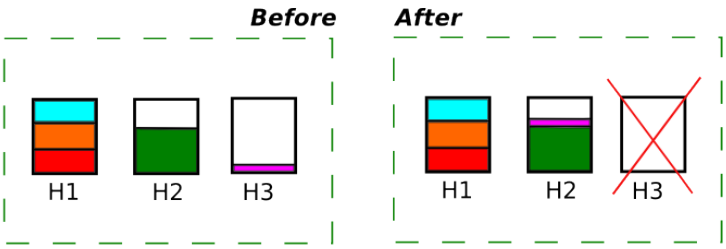
EVENT TYPE

-> Host load threshold
-> System power threshold
-> Arrival of a new job
-> End of a job
-> System load threshold



Example:

```
event.received(from:HostSensor, host_i.load <= 5%)
if (reschedule(host_i.jobs))
then migrate(host_i.jobs);
        shutdown(host_i);
endif
```



Each event generates a call to an activity which is described in a UML activity diagram.

The Decision Center calculates the necessary actions. Here `migrate(h3.jobs)` and `shutdown(h3)`.

The Decision Center enforces the actions via the corresponding actuators.

TUNe provides a platform that simplifies this process.

cnrs
IRIT
CNRS - INPT - UPS - UT1 - UTM
Université de Toulouse