# Recent development in DIET: from Grid to Cloud

Frédéric Desprez, Luis Rodero-Merino, Eddy Caron, Adrian Muresan

# Recent Developments in DIET: From Grid to Cloud

by Frédéric Desprez, Luis Rodero-Merino, Eddy Caron and Adrian Muresan

***The Distributed Interactive Engineering Toolkit, or DIET, project started with the goal of implementing distributed scheduling strategies on compute Grids. In recent times, the Cloud phenomenon has you-go billing approach. This led to a natural step forward in the evolution of DIET, with the inclusion of Cloud platforms in resource provisioning. DIET will be used to test resource provisioning heuristics and to port new applications that mix grids and Clouds.***

In 2000, the Grids and Algorithms, or GRAAL, research team of INRIA, located in Ecole Normale Supérieures de Lyon, France, initiated the Distributed Interactive Engineering Toolbox project under the supervision of Frédéric Desprez and Eddy Caron. The project is focused on the development of scalable middleware with initial efforts on distributing the scheduling problem across a hierarchy of agents, at the top of which sits the Master Agent, or MA. At the bottom level of a DIET hierarchy one can find the Service Daemon, or SeD, agents. SeDs are connected to the MA by means of Local Agents, or LAs. Over the last few years, the Cloud phenomenon has been gaining more and more traction in the industry and in research communities because of its qualities, the most interesting of which is its on-demand resource provisioning model and its pay-as-you-go billing approach. We deem these features to be highly interesting for DIET.

### From Grid to Cloud

The first step towards the Cloud was to enable DIET to take advantage of on-demand resources. This should be done at the platform level and be transparent to the DIET user. The authors and David Loureiro have targeted the Eucalyptus Cloud platform as it implements the same management interface as Amazon EC2, but unlike the latter it allows for customized deployments on the user's hardware. The team has implemented the scenario in which the DIET platform sits completely outside of the Eucalyptus platform and treats the Cloud only as provider of compute resources when needed. This opened the path towards new researches around grids and Cloud resource provisioning, data management over these platforms and hybrid scheduling in general.

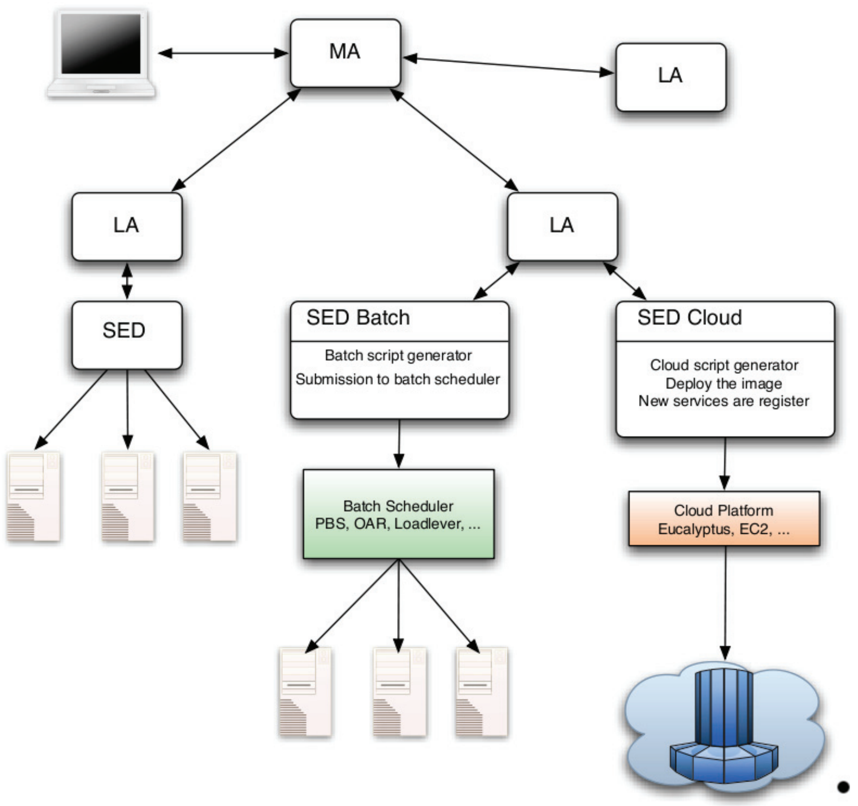


*Figure 1: The cloud-enabled DIET hierarchy.*

### Cloud application resource scaling

The on-demand provisioning model for resource allocation and the pay-as-you-go billing approach that Cloud systems offer makes possible the creation of more cost-effective approaches for application resource provisioning. A Cloud application can scale its resources up or down to better match its usage and to reduce the number of unused, yet paid for, resources. This leads to smart auto-scaling strategies.

By taking into account research done around self-similarities in web traffic, the authors have developed a resource usage prediction model that identifies similar past resource usage patterns from a historic archive. Once identified, they provide an insight into what the short-term usage of the platform will be. This approach can be used to predict the usage of the most important types of resources of a Cloud client and thus give an insight of what type of virtual machine to instantiate or terminate when the Cloud application is rescaled. We have tested this approach against resource usage traces from one Cloud client and three production grids and obtained encouraging results.

### Economy-based resource allocation

The dynamics that Cloud systems bring in combination with the agent-based DIET platform led us towards an economic model for resource provisioning. The ultimate goal is to guarantee resource sharing fairness and avoid starvation. The authors have done this by simulating the dynamics of a tender/contract-net market. In this market contracts are established between platform users (the DIET clients) and resource providers (the

DIET SeDs). Users send requests to the DIET platform for the execution of their tasks and resource providers reply with offers, each containing the cost and duration of the task execution. A user-defined utility function is applied to identify the best offer and the corresponding SeD will run the task.

In this scenario, platform users compete against each other for resource usage while the resource providers compete against each other for profit. Resource prices, which determine the offer costs, fluctuate depending on each provider's resource usage level. Hence, users will tend to choose SeDs with more free resources and so lower prices.

### What's next?

Future directions include implementing a complete automatic resource scaling strategy for Cloud clients and testing against real-life situations.

We are also looking forward towards integrating Cloud-specific elements into the DIET scheduler for existing applications. The final goal is to see if deployment on a Cloud platform would yield a better performance and if so then with what scheduling modifications. Finally, we plan to study hybrid scheduling strategies mixing static grids and dynamic Clouds for a more efficient resource management of large scale platforms.

**Link:**
The DIET project:
http://graal.ens-lyon.fr/DIET

**Please contact:**
Adrian Muresan
Ecole Normale Supérieure de Lyon, France
Tel: +33 4 37 28 76 43
E-mail: Adrian.Muresan@ens-lyon.fr

# Addressing Aggregation of Utility Metering by using Cloud – The Power Grid Case Study

by Orlando Cassano and Stéphane Mouton

***Utility grids are generating increasingly huge amounts of metering information. Grid operators face rising costs and technical hurdles to aggregate and process data. Can Cloud Computing tools, developed notably by Web companies to deal with large data sets, also be used for power grid management?***

As electricity in the current state of technology cannot be stored, consumption in power grids is continuously counterbalanced by production. Both producers and consumers are connected to the grid through metered Access Points (AP). Every month, grid operators have to determine the amount of energy produced or used by each stakeholder, knowing that the sum of produced energy, either out of the grid by power plants or within the grid (eg by wind turbines), equals the sum of consumed energy, from effective use and losses in the grid. Amounts are aggregated in order to obtain amount of "allocated" energy by stakeholder. The volume of data at stake in the allocation computation depends on the size of the grid.Data are produced every 15 minutes, and datasets may be huge. For example, there are roughly 8 million APs in Belgium, each producing 96 metering data outputs per day: allocation for a month would therefore require handling of more than 23 billion records.

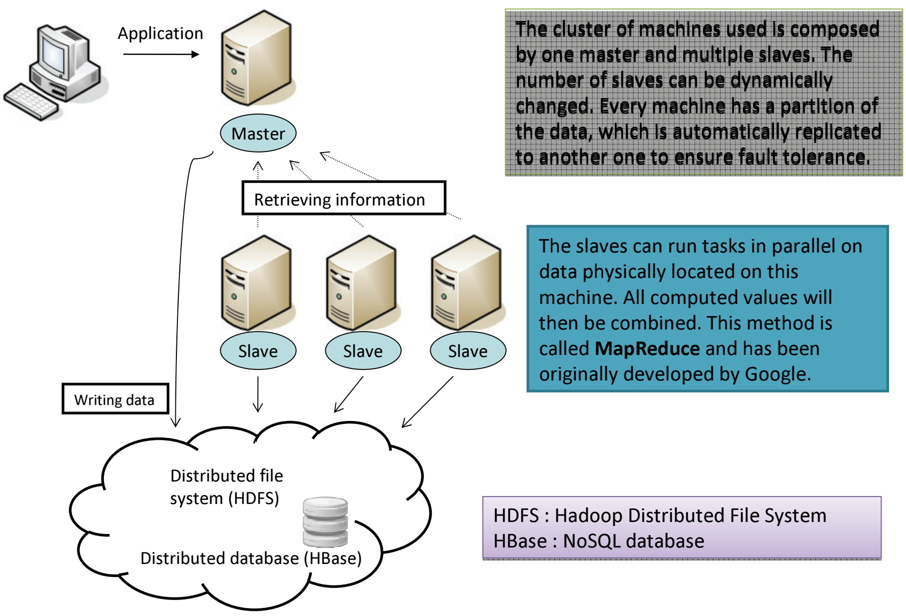


*Figure 1: Set up of the Cloud architecture.*

Data aggregation is currently based on existing Relational DataBase Management Systems (RDBMS). However performance of such software is declining with the increasing volume of data to process. Performance can be improved by investing in hardware and sophisticated software setups, like database clusters, but such an investment is not necessarily economical, with the cost of such a setup increasing disproportionately in relation to data processing capacity.

The goal of our research was to overcome the limitations of RDBMS by scaling performance according to growth of aggregated data. Moreover the allocation algorithm is a good candidate for parallelization as sums have to be performed on distinct data sets, ie, per stakeholder. For this reason we investigated the use of programming platforms and frameworks, identified as providing Platform as a Service (PaaS) on Cloud infrastructures, to enable scalability in data storage and processing.