Provisioning of Virtual Machines in Federated Clouds

Jonathan Rouzaud-Cornabas

CNRS – LIP / CC-IN2P3

November 20, 2012

Problem

Using efficiently Cloud for HPC is complex, bad provisioning and task allocation can

- increase cost very fast;
- increase (local and global) makespan;

Problem

Current approachs focus only on selecting compute resource templates

- Application allocation on VMs (after provisioning) must be taken into account;
- Resources are heterogenous;
- Network latency, jitter and packet loss are high;
- Variability is high (interference, not the same underlying hardware, over-provisioning);
- Reliability must be taken into account e.g., Spot Instances;

Motivation

- Ease the usage of Cloud for Scientific Computing;
- Network latency and speed are the current weak points on Clouds for Scientific Computing;

Motivation

- Bag Of Tasks (BoT) are less sensitive to latency than tightly coupled applications and thus better adapted to Clouds;
- BoT represents a large part of applications running on Cluster, Grid, Clouds;

Motivation

- Bag Of Tasks (BoT) are less sensitive to latency than tightly coupled applications and thus better adapted to Clouds;
- BoT represents a large part of applications running on Cluster, Grid, Clouds;

- Many task allocation algorithms for BoT exists;

Goals

- Studying through simulation the different parameters of a BoT;
- Bag Of Tasks with real-world characteristics;
- Take into account the different methods that provide the same services e.g. EBS and S3;
- Being able to test by simulation the different algorithms and policies to provide high quality feedbacks to the user;

Goals

- Studying through simulation the different parameters of a BoT;
- Bag Of Tasks with real-world characteristics;
- Take into account the different methods that provide the same services e.g. EBS and S3;
- Being able to test by simulation the different algorithms and policies to provide high quality feedbacks to the user;

- Propose provisioning and allocation mechanisms to improve a set of requirements (cost, deadline, etc.);

Goals

- Studying through simulation the different parameters of a BoT;
- Bag Of Tasks with real-world characteristics;
- Take into account the different methods that provide the same services e.g. EBS and S3;
- Being able to test by simulation the different algorithms and policies to provide high quality feedbacks to the user;

- Propose provisioning and allocation mechanisms to improve a set of requirements (cost, deadline, etc.);

- Simulator that reflects the real behavior of Clouds: network, storage, compute, virtualization overhead, interference, localisation, etc.;
- Extendable simulator to other type of workloads;

Bag Of Tasks Model

- x Tasks and No dependency between them **but a large number of parameters**;

Bag Of Tasks Model

- x Tasks and No dependency between them **but a large number of parameters**;
- Three parameters (I, O and FLOPS) for tasks in BoT (impact on task allocations):
  - Homogenous;
  - Stochastic (uniform/bimodal/heavytail);

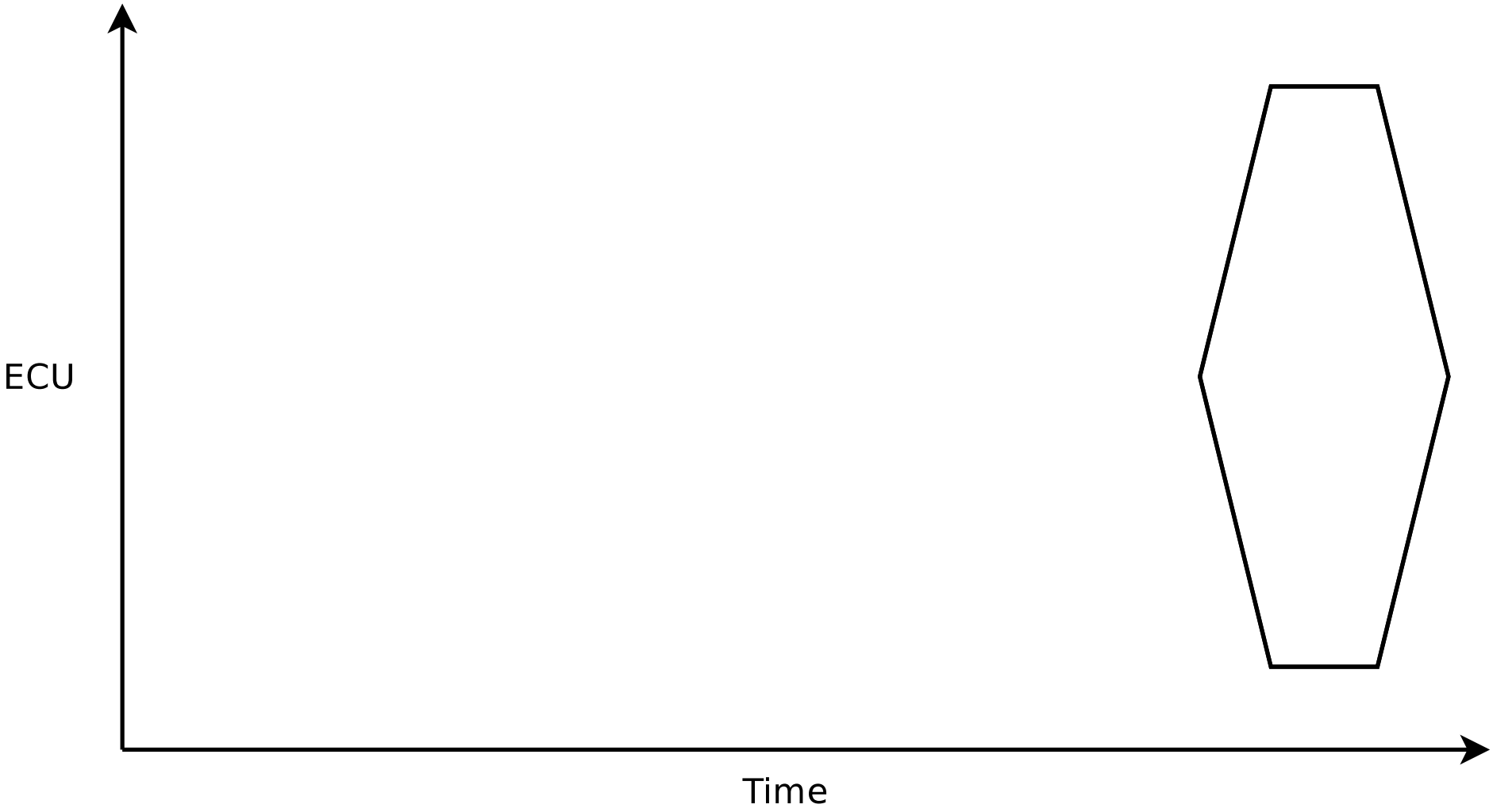
Bag Of Tasks Model

- x Tasks and No dependency between them **but a large number of parameters**;
- Three parameters (I, O and FLOPS) for tasks in BoT (impact on task allocations):
  - Homogenous;
  - Stochastic (uniform/bimodal/heavytail);
- Different task arrival (impact on provisioning) models:
  - At the beginning;
  - Poisson;
  - Dependency and think time;

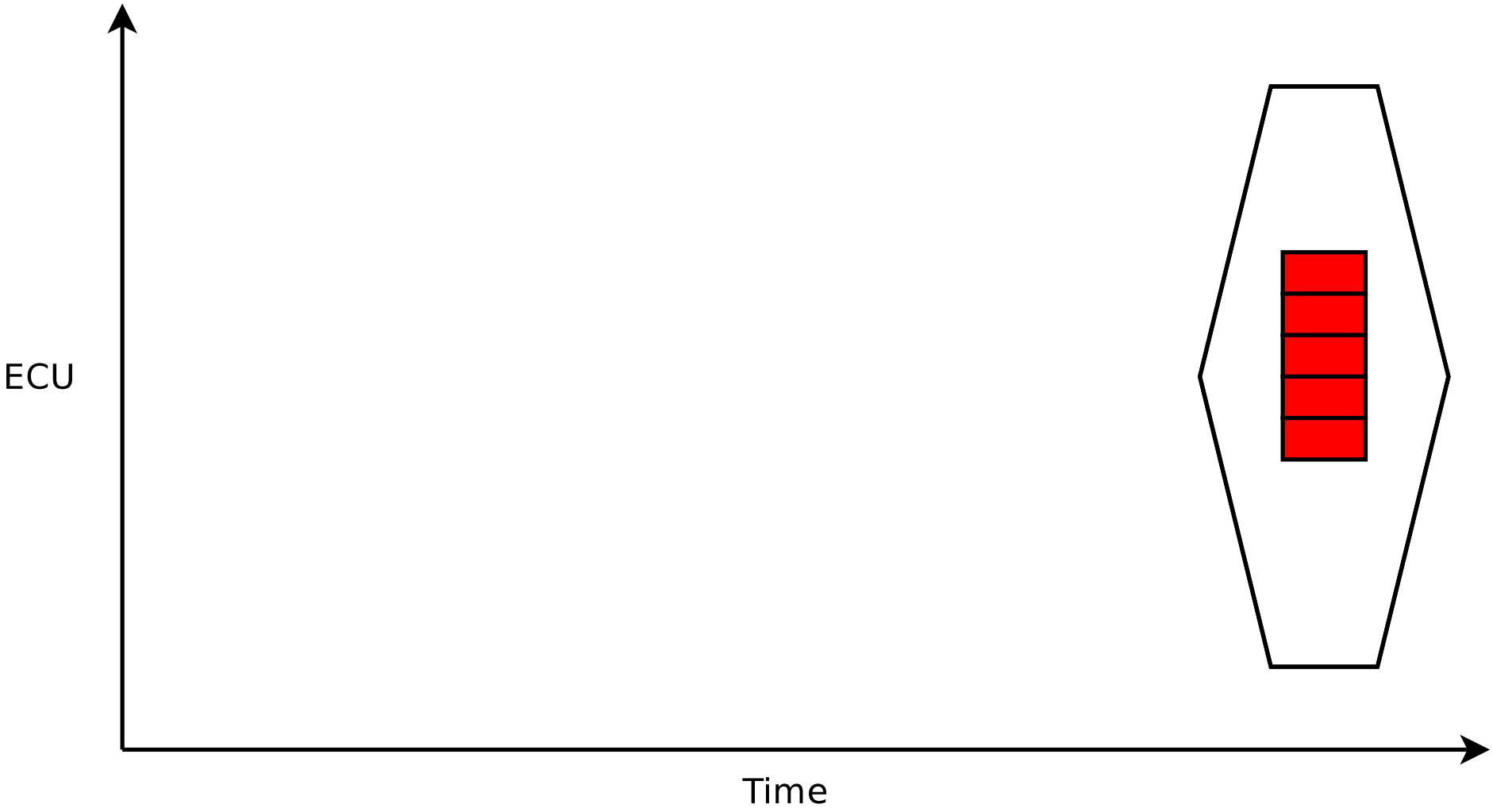
Bag Of Tasks Model

- x Tasks and No dependency between them **but a large number of parameters**;
- Three parameters (I, O and FLOPS) for tasks in BoT (impact on task allocations):
  - Homogenous;
  - Stochastic (uniform/bimodal/heavytail);
- Different task arrival (impact on provisioning) models:
  - At the beginning;
  - Poisson;
  - Dependency and think time;
- Different objectives:
  - Cost;
  - Performance;
  - Deadline;
  - Etc.

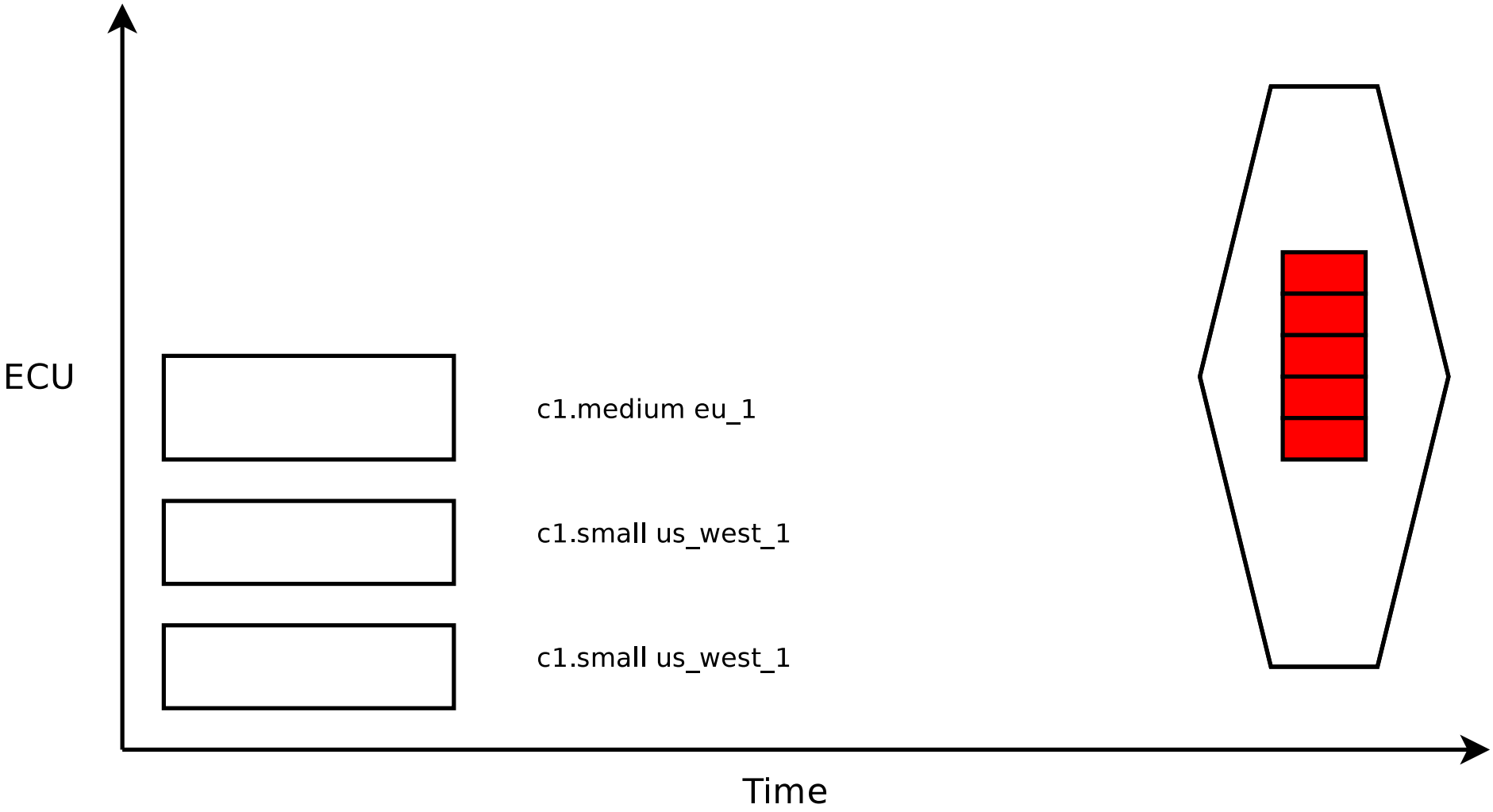
Bag of Tasks Example



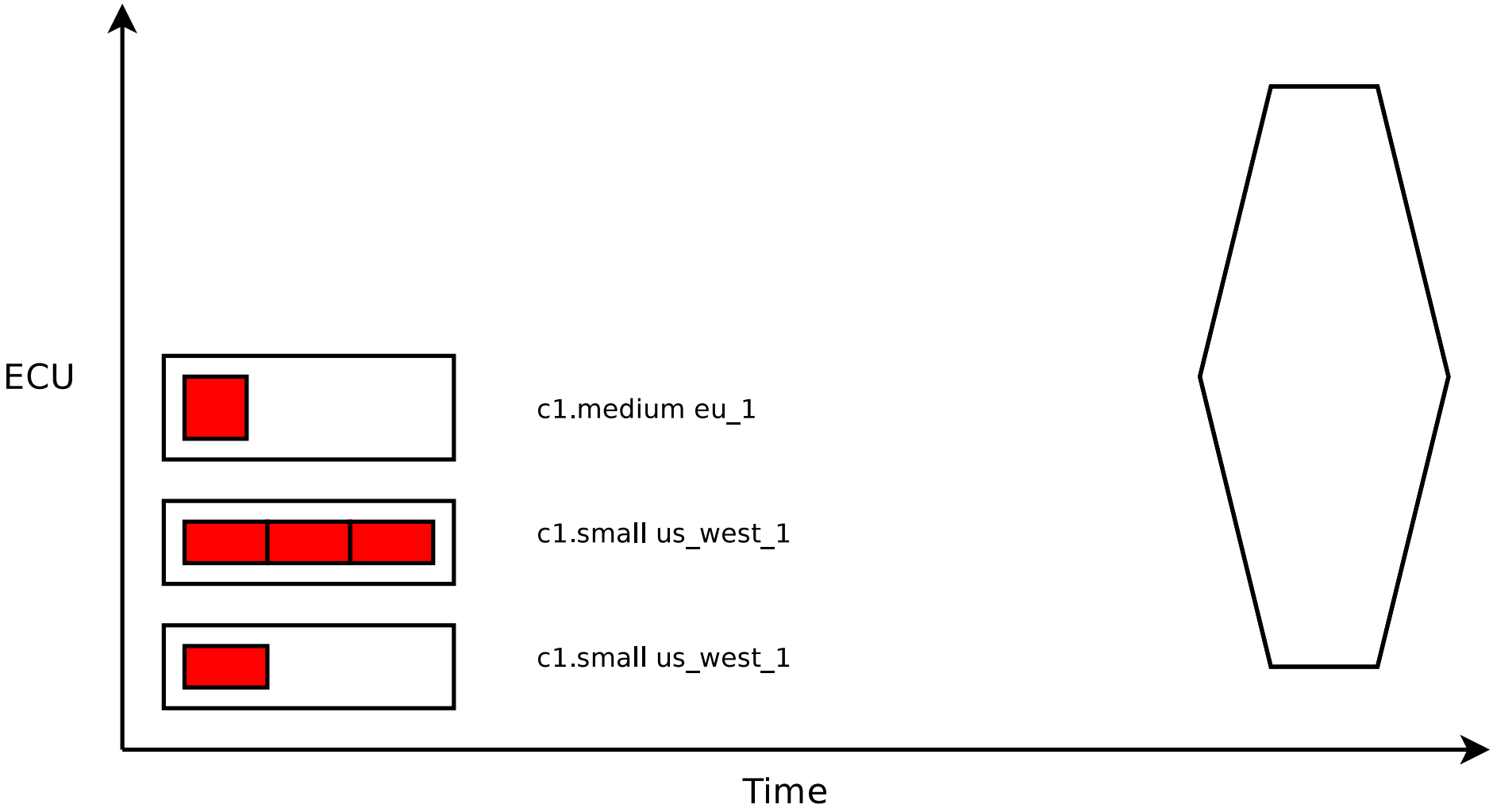
Bag of Tasks Example



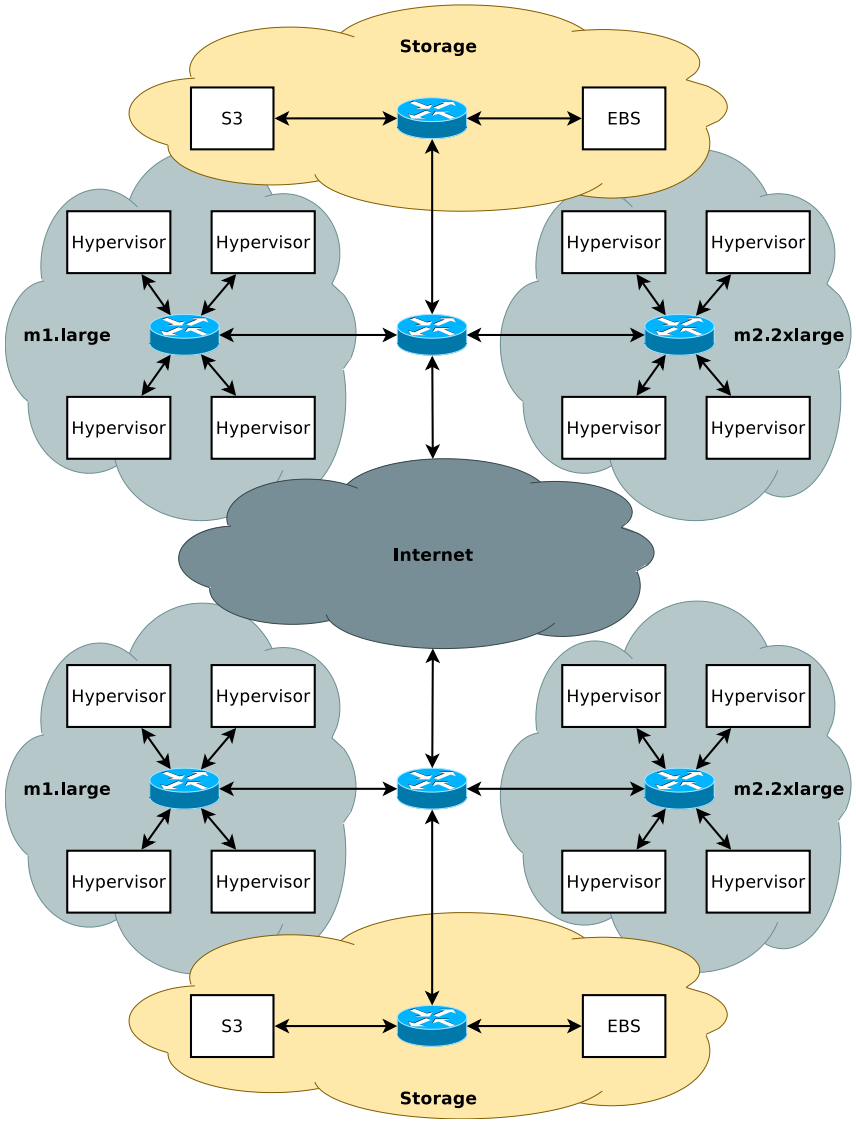
Bag of Tasks Example



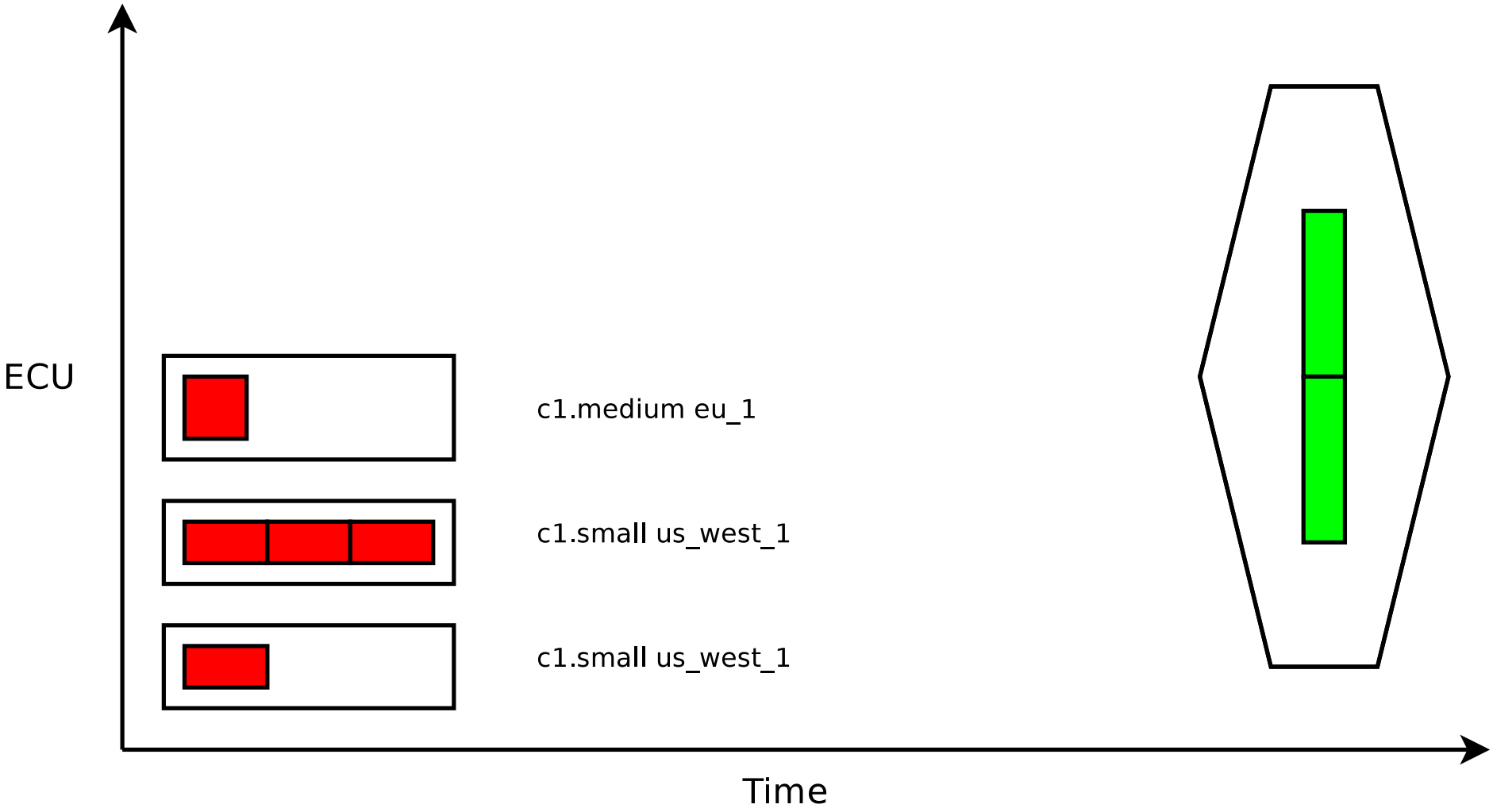
Bag of Tasks Example



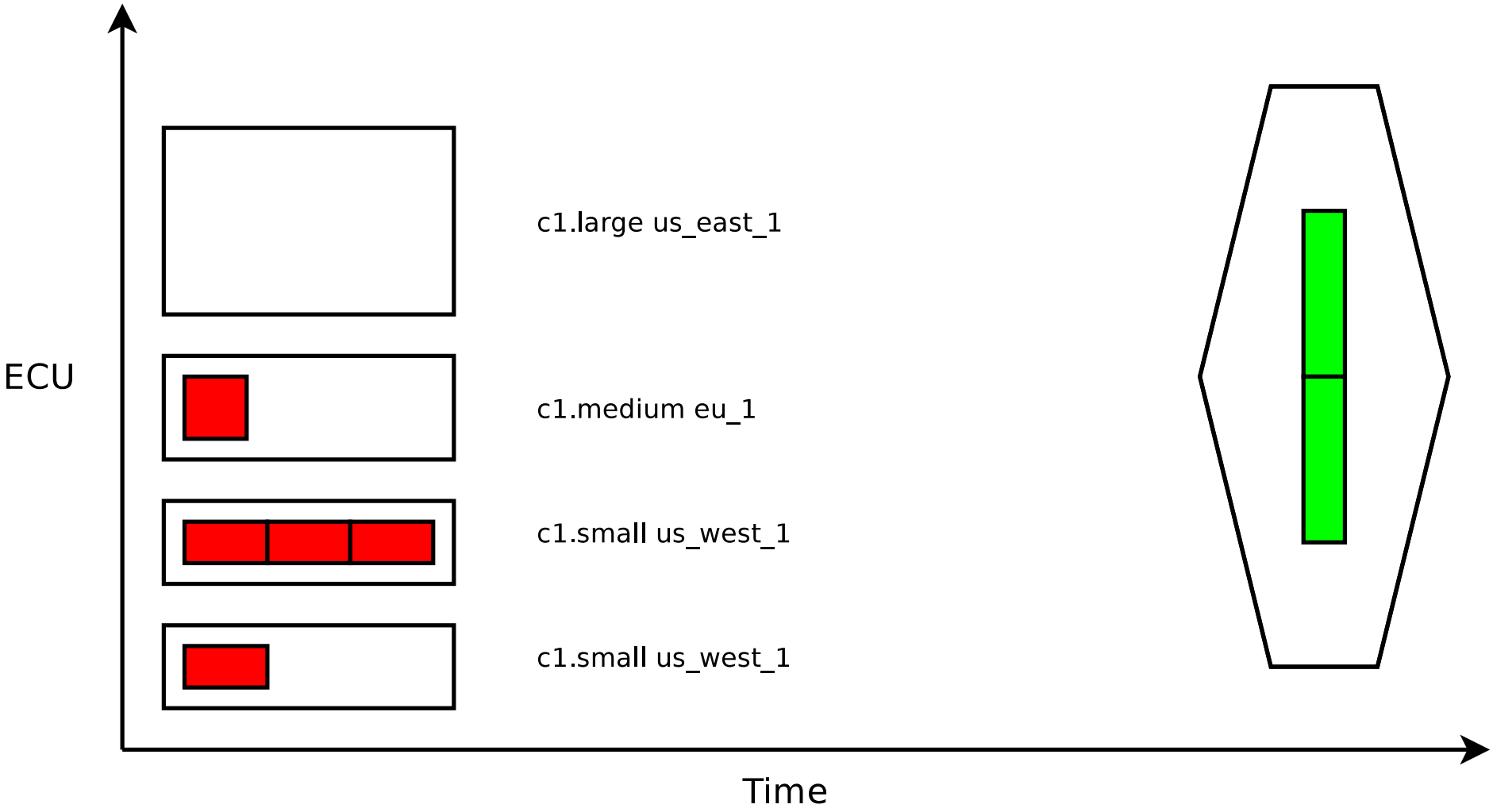
Bag of Tasks Example



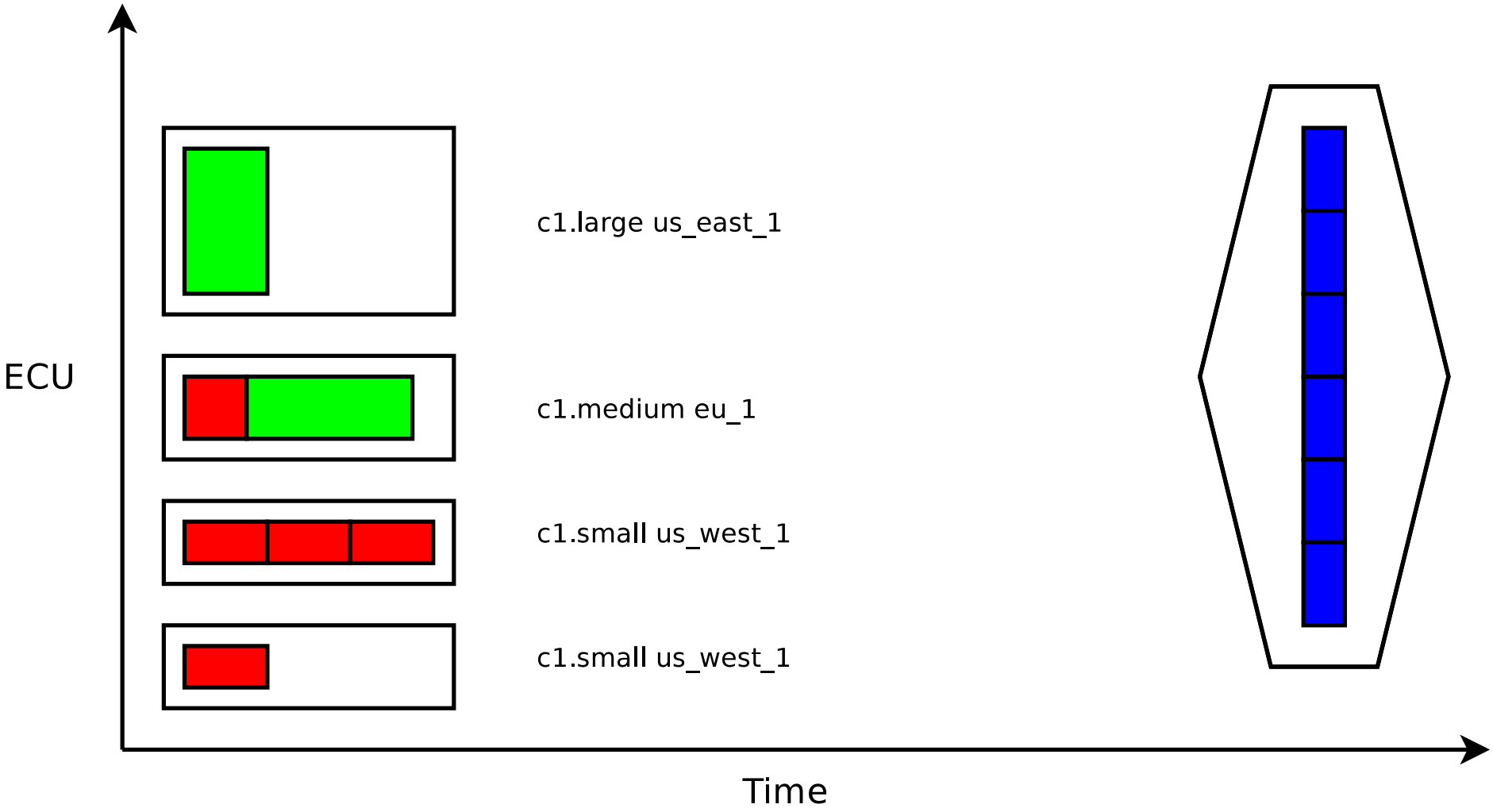
Bag of Tasks Example



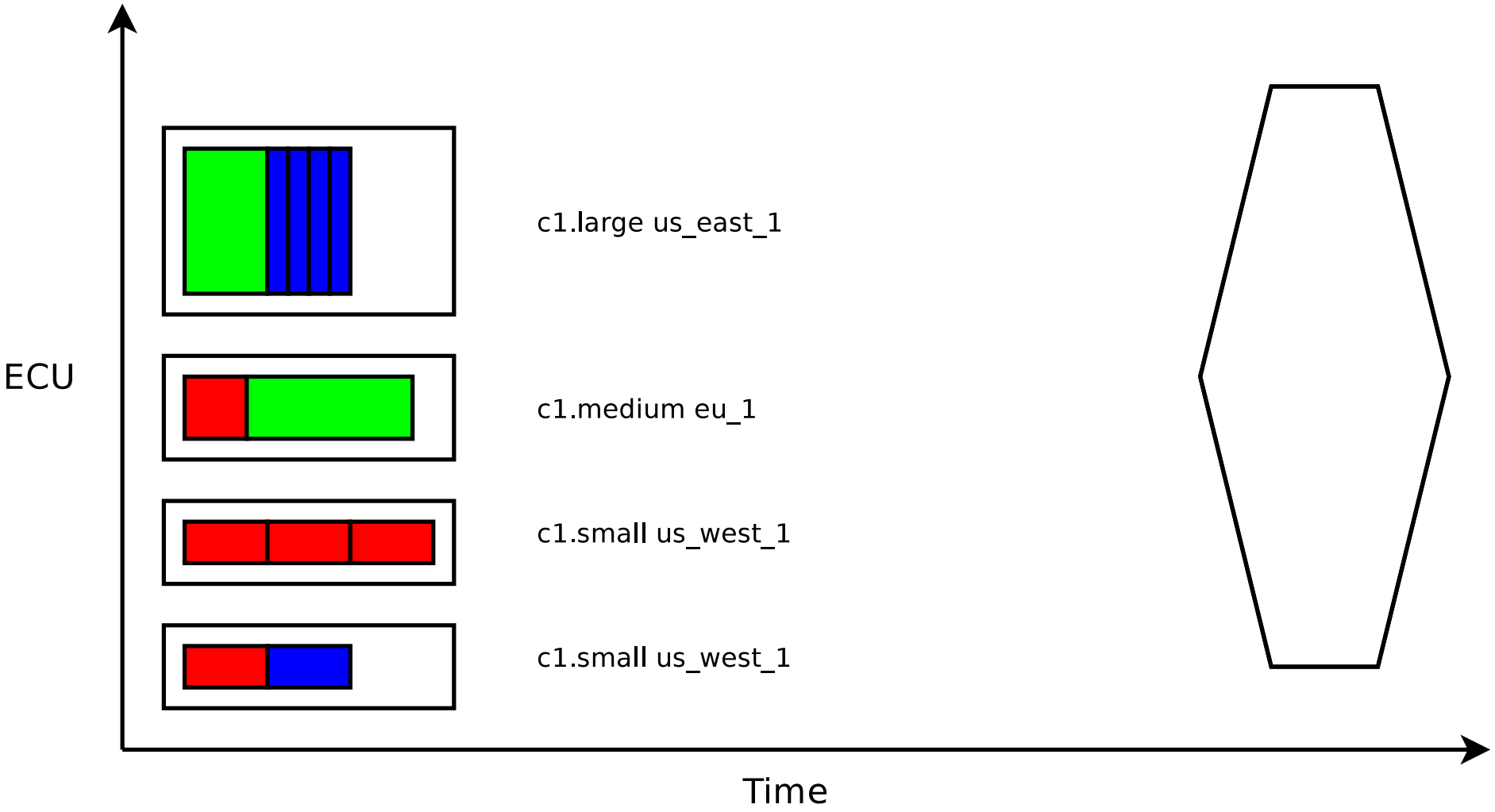
Bag of Tasks Example



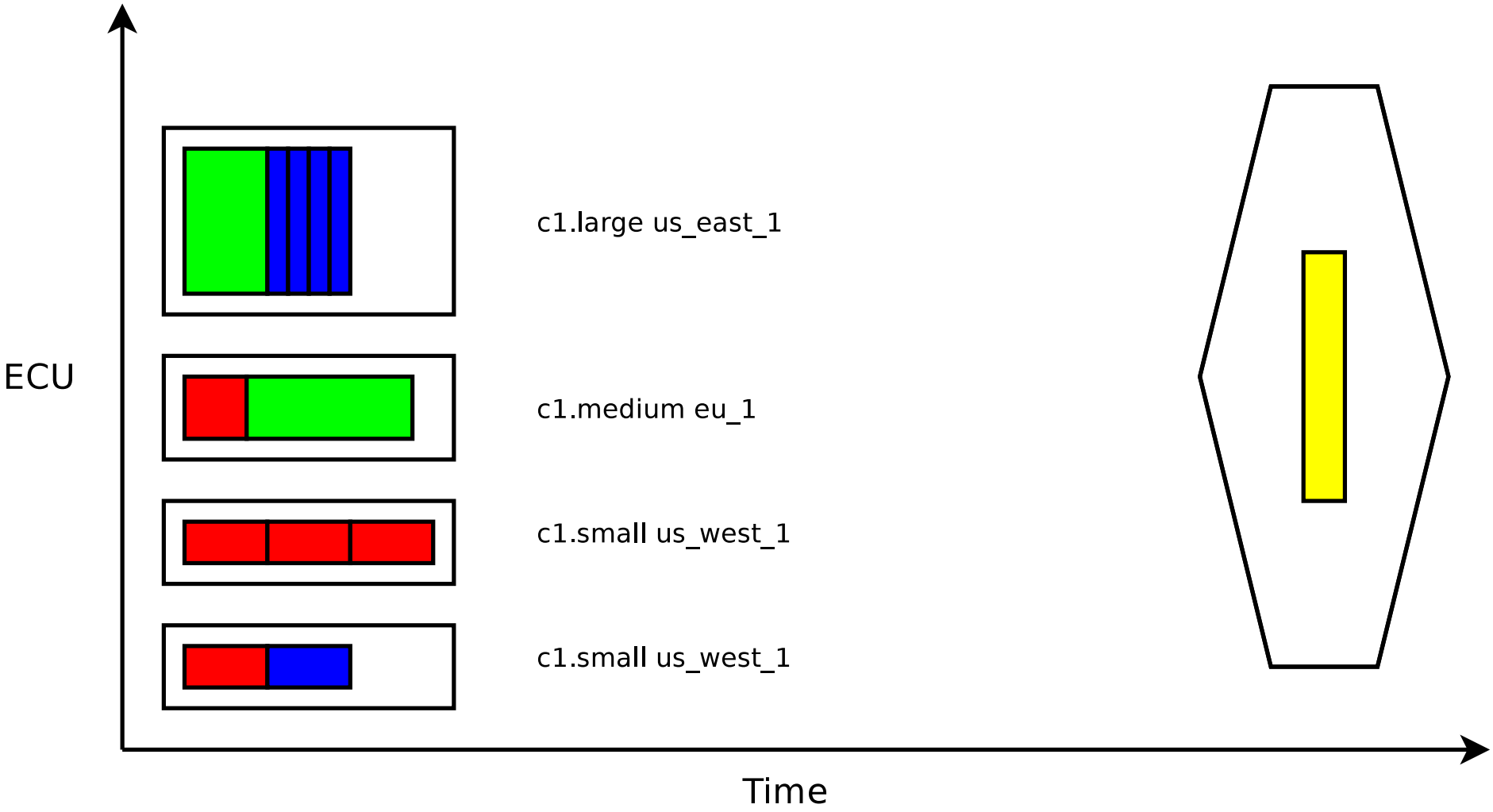
Bag of Tasks Example



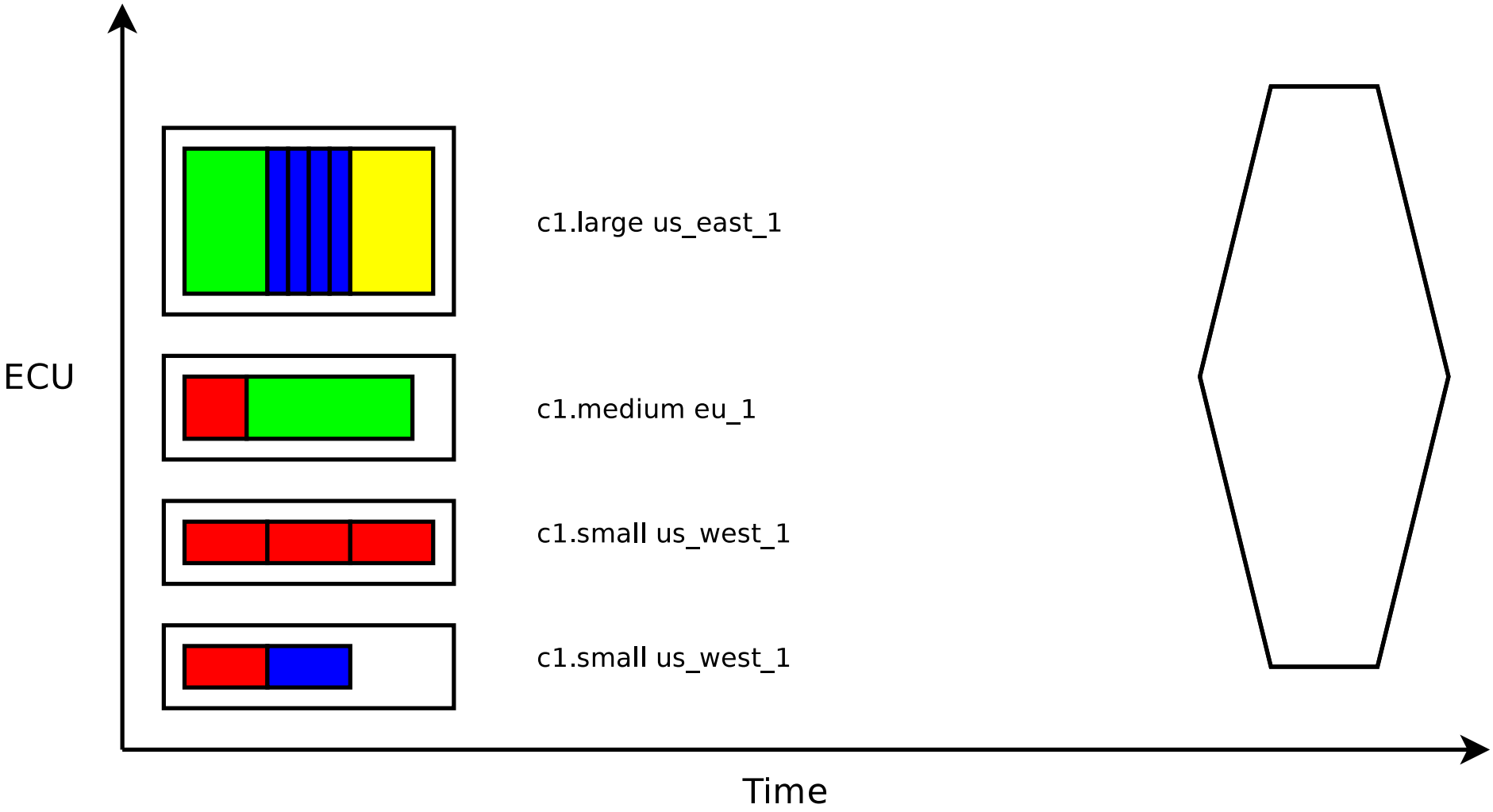
Bag of Tasks Example



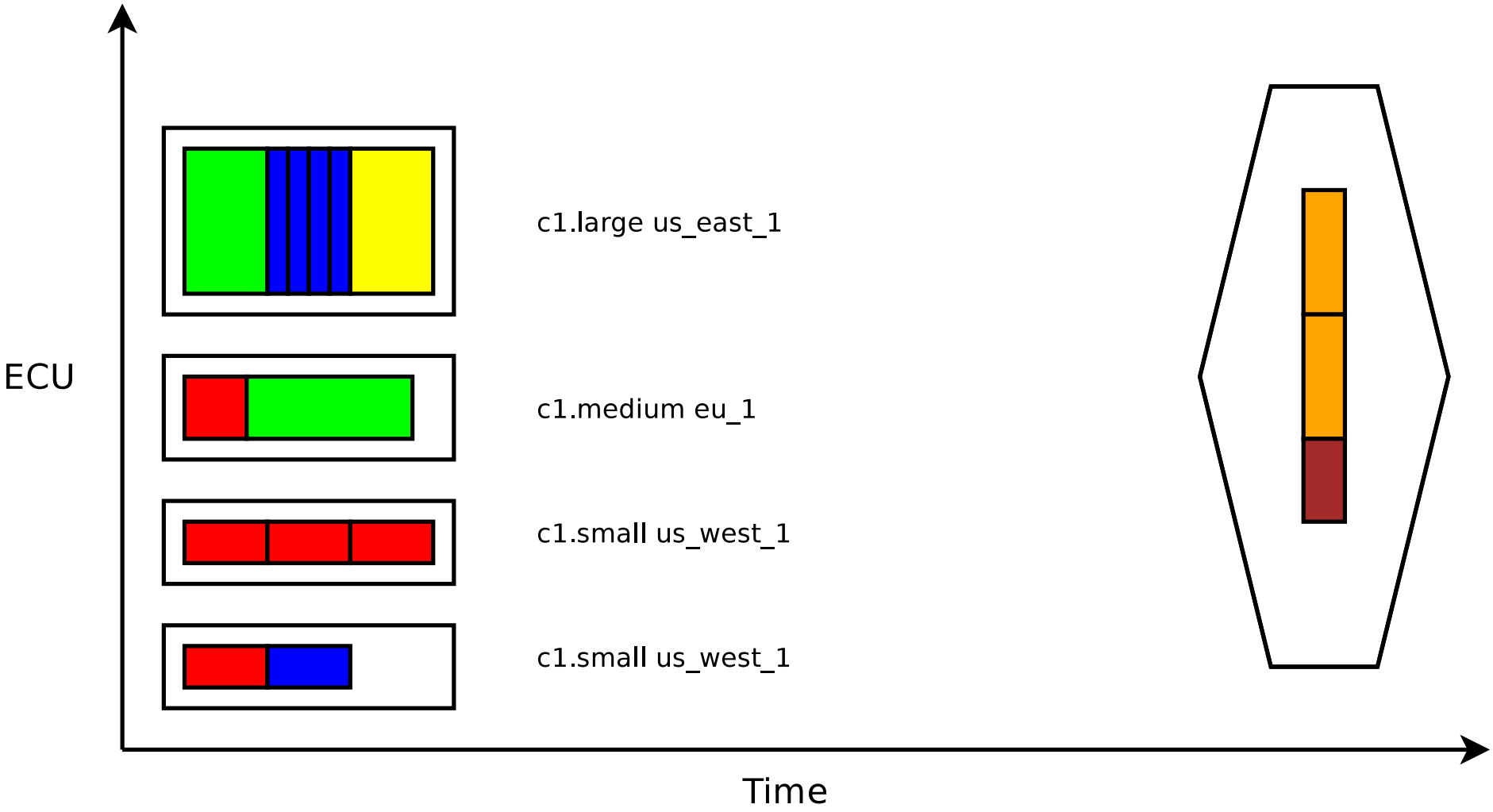
Bag of Tasks Example



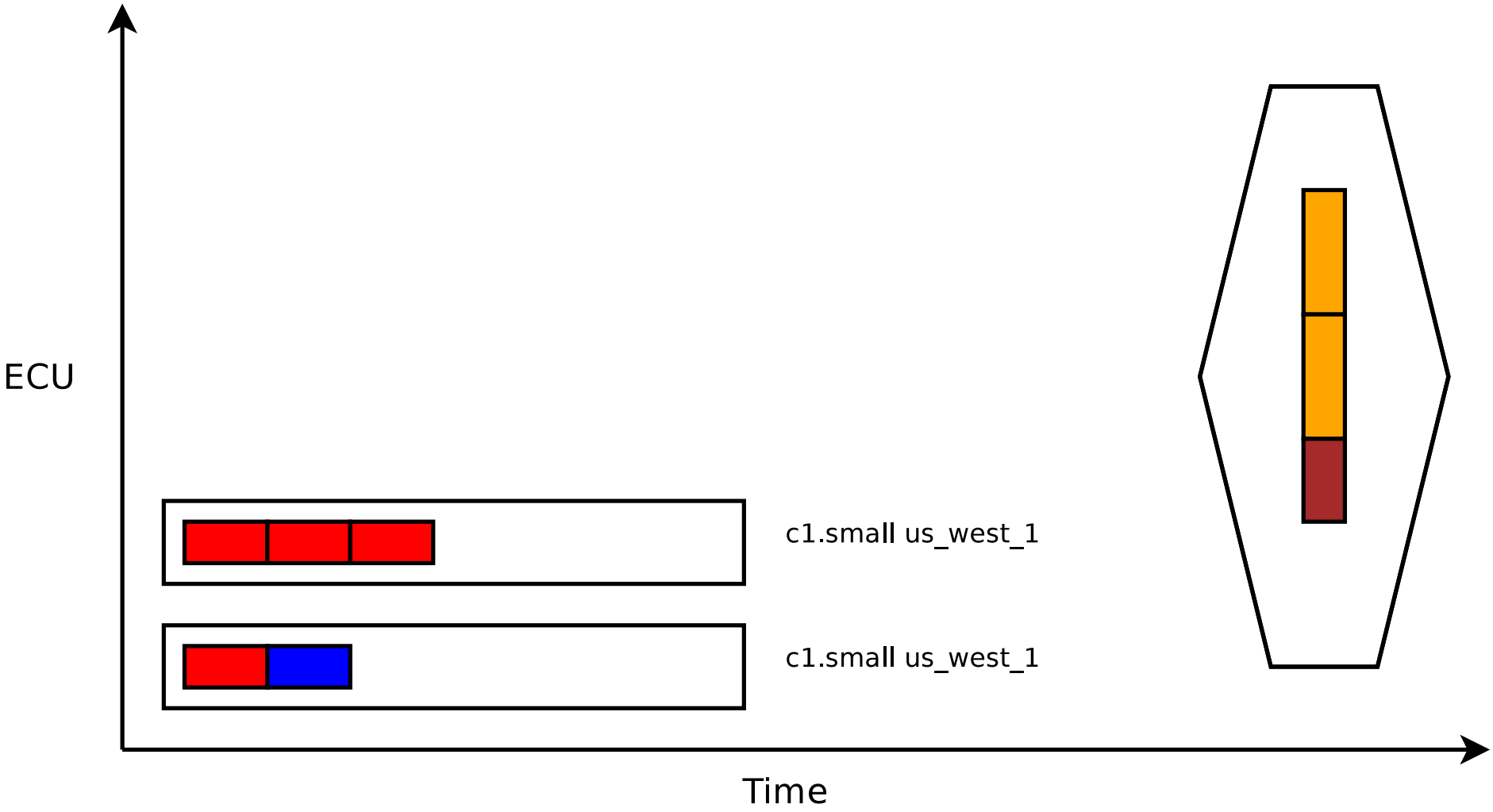
Bag of Tasks Example



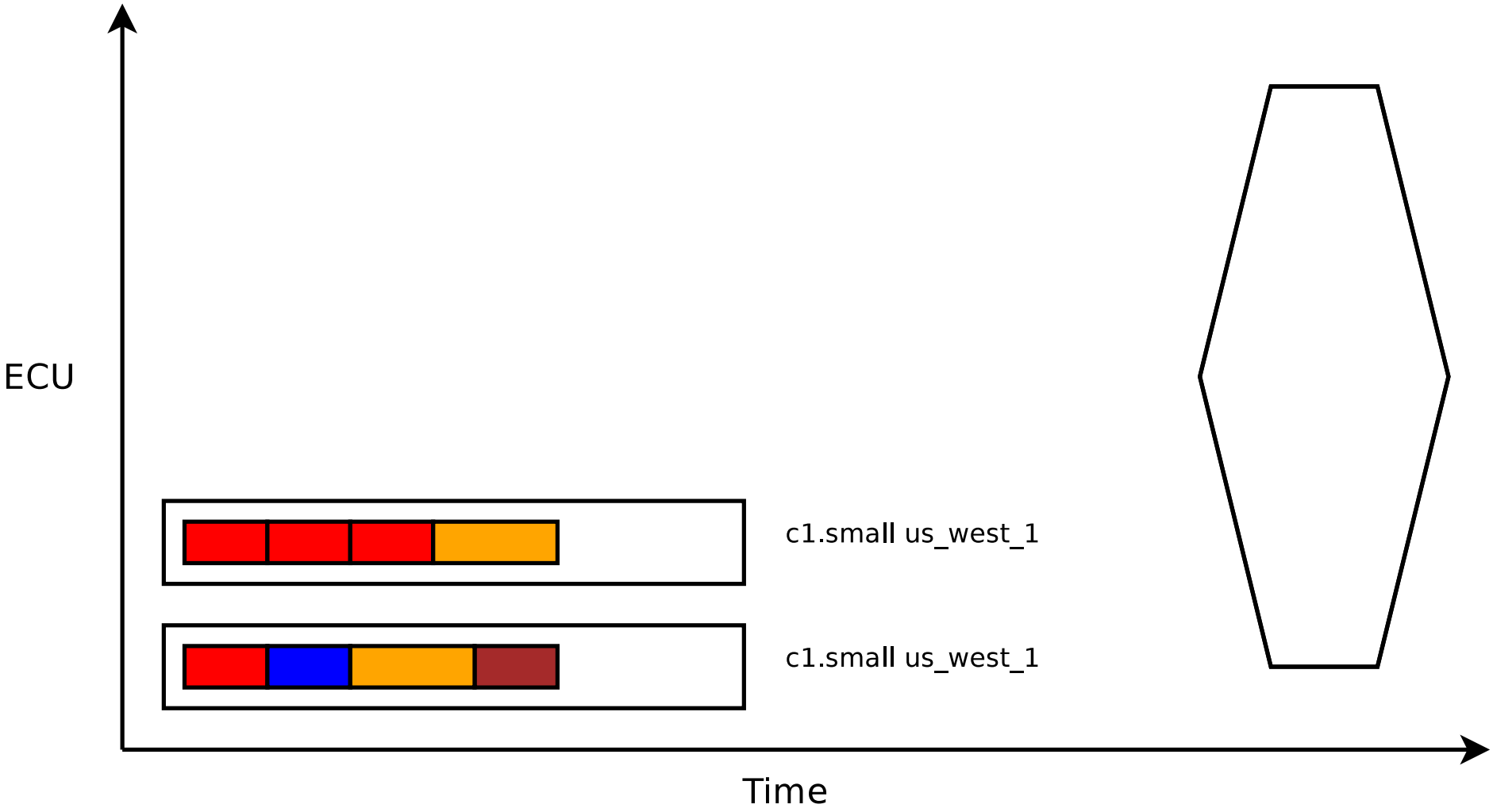
Bag of Tasks Example



Bag of Tasks Example



Bag of Tasks Example



Work in Progress

- Work in Progress;
- 7 Provisioning Algorithms;
- 18 Task Allocation Algorithms;
- Best combination of algorithms for each type of Bag Of Tasks;
- Storage policy impact;

SimGrid Cloud Broker

- A new SimGrid project: Started 1 year ago;
  - 12 years old; Open Source
  - Collaboration Loria / Inria Rhone-Alpes / CCIN2P3 / U. Hawaii
  - Allows studies of Grid, P2P, HPC, Volunteer Computing and others
  - Validated, Scalable, Usable; Modular; Portable
  - Grounded +100 papers; 100 members on simgrid-user@; Open Source

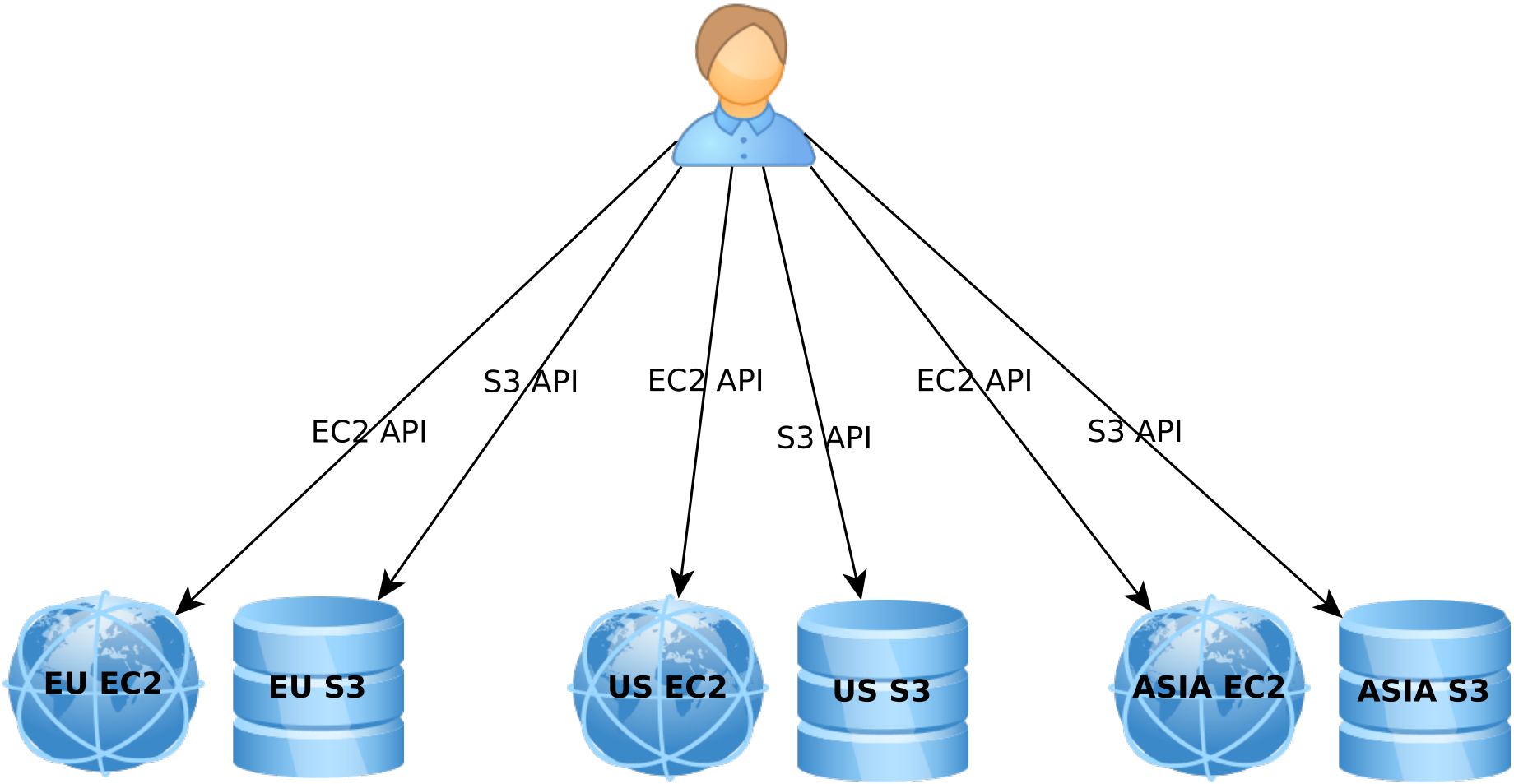
SimGrid Cloud Broker

- A new SimGrid project: Started 1 year ago;
- Not yet another Cloud simulator: Multi-clouds environment based EC2/S3 API;

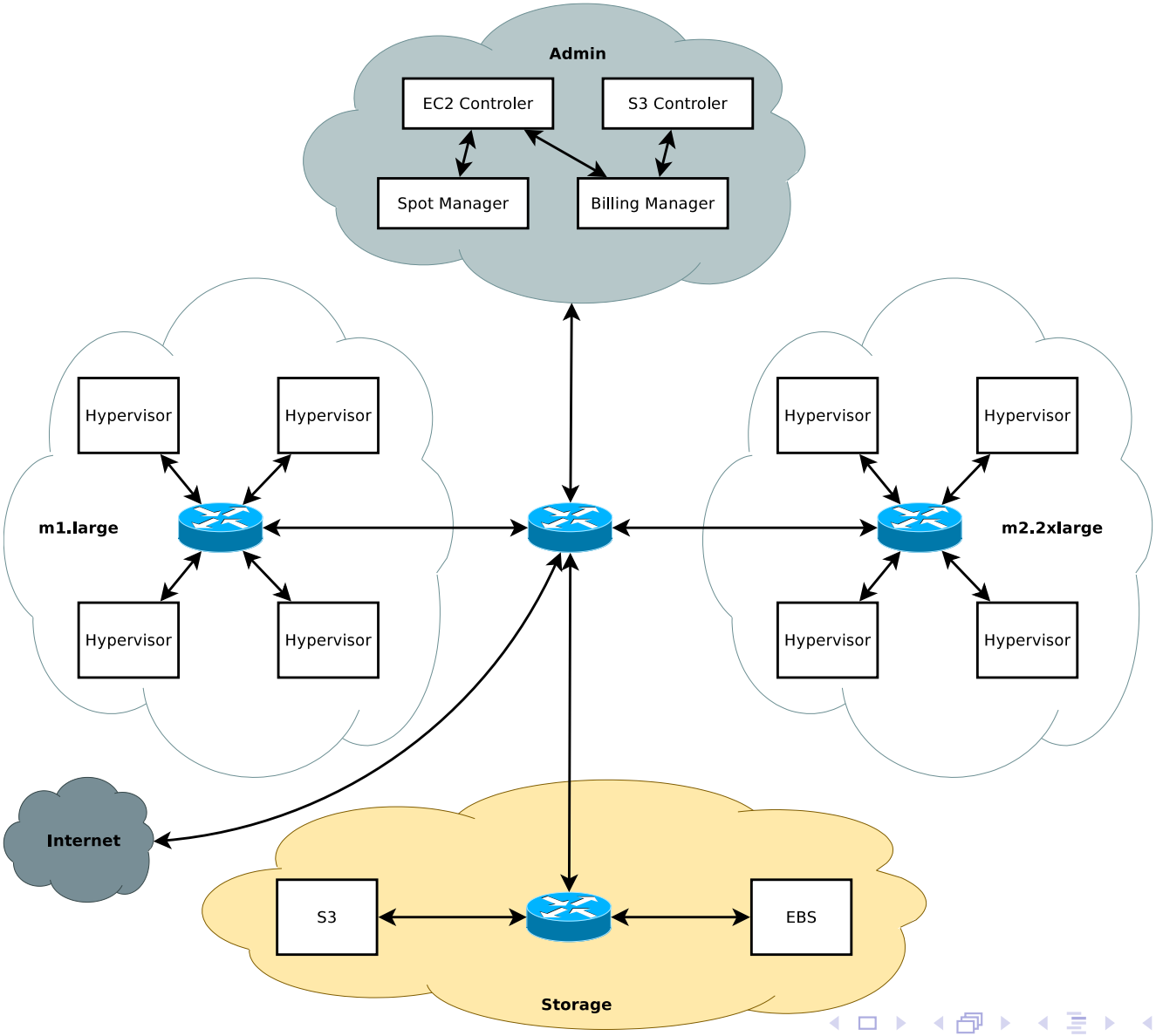
SimGrid Cloud Broker

- A new SimGrid project: Started 1 year ago;
- Work in progress but already features:
  - All AWS regions;
  - All instance types (resources and prices);
  - On-demand and spot instances;
  - S3 and EBS Storage;
  - Accounting of resources usage (Network, Compute, Storage);
  - Resources performance models based on information given by Amazon and extracted from scientific papers;
  - Spot Instances: 3 dynamic price policies (random, file, model);

SGCB: Client View



SGCB: Inside a Cloud



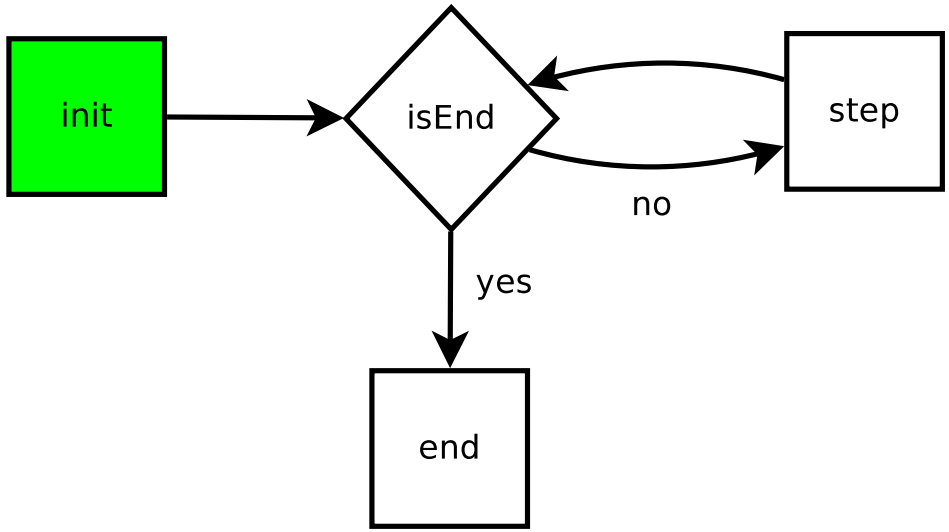
SGCB: Client View

- Purpose: Test an application and its provisioning mechanisms before deploying it;
- A SGCB user will just have to do call to S3 and EC2 API inside the simulator;
- Using SimGrid processes and tasks, he can also simulate his applications;
- A set of examples exists to demonstrate the different use cases;
- A complete SimGrid trace is available to enable post execution analysis: bill, network usage, etc.;

SGCB: Application Scenario (1)

- Application Scenario is a way in SGCB to ease the simulation of an application;
- The life cycle in Clouds is composed of 3 basic steps:
  - **init** Provisioning of the VM and start the application;
  - **step** Adapts the number and type of VMs to the current load (while not *isEnd*);
  - **end** Stop the application and release the resources;
- **isEnd** function checks if the application is over;

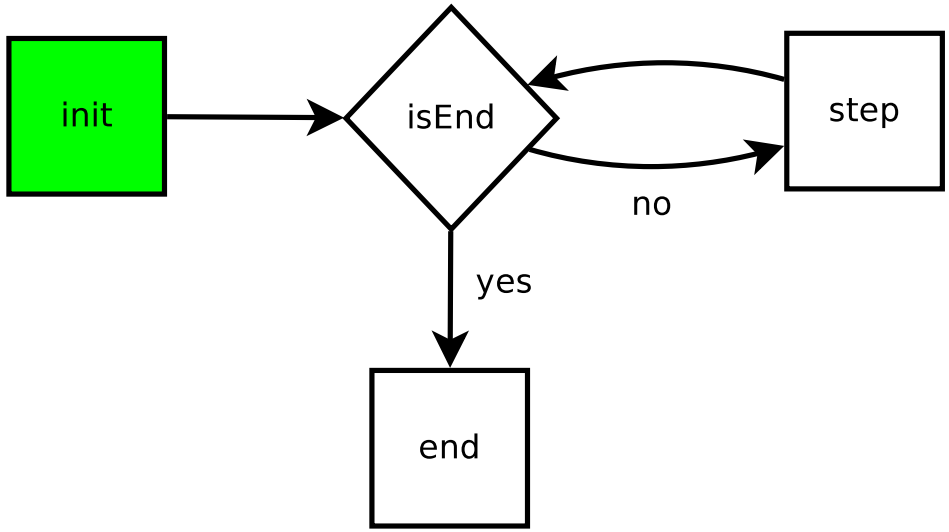
SGCB: Application Scenario (Example)



init

1. Select the best template and region where to start the master;

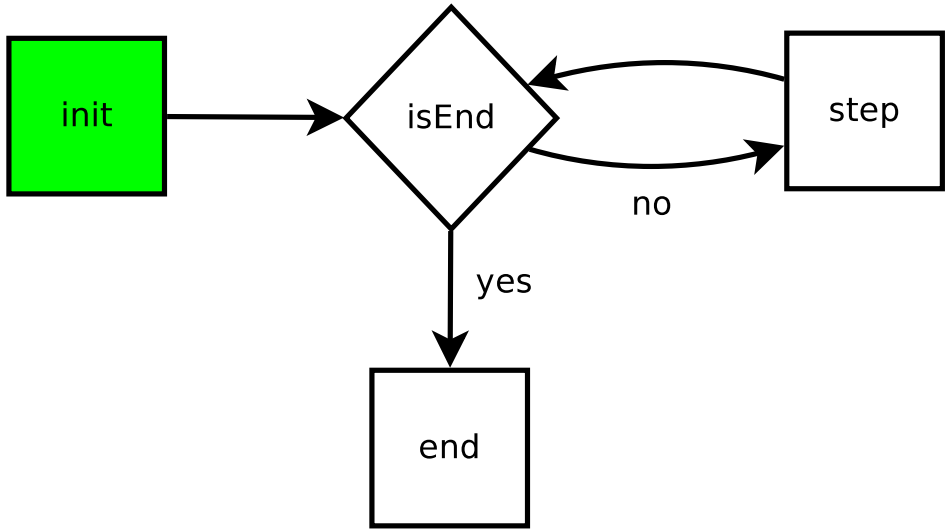
SGCB: Application Scenario (Example)



init

1. Select the best template and region where to start the master;
   1. Upload to S3 the VM image of the master VM;
   2. Register the uploaded image as an S3-backed AMI;
   3. Run one instance of template in the selected region;
   4. When the VM is started (describeInstances), start the master application;

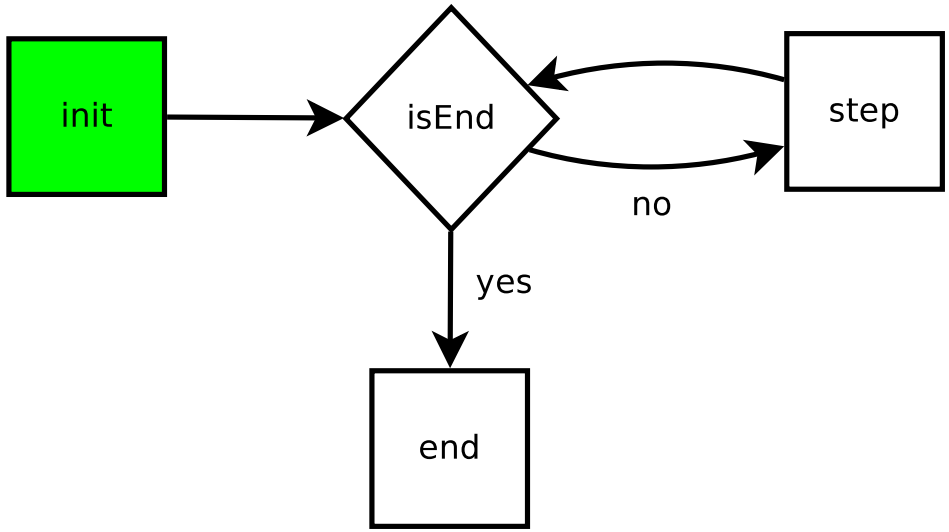
SGCB: Application Scenario (Example)



init

1. Select the best template and region where to start the master;
2. Do the same for the Slaves;

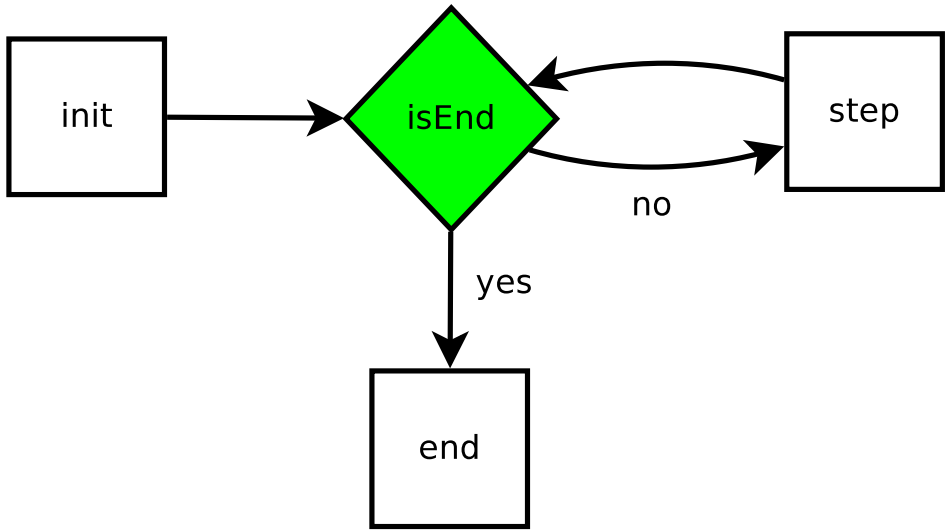
SGCB: Application Scenario (Example)



init

1. Select the best template and region where to start the master;
2. Do the same for the Slaves;
3. Send 10 Computing Tasks out of N to the master;

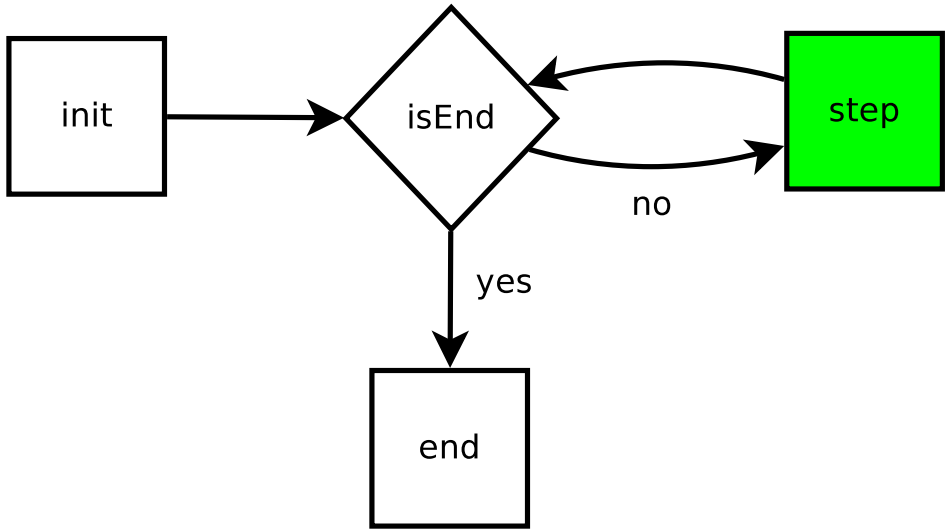
SGCB: Application Scenario (Example)



isEnd

1. Check if all N Computing Tasks have been sent;
2. Check if all results of the computing tasks have been fetched;

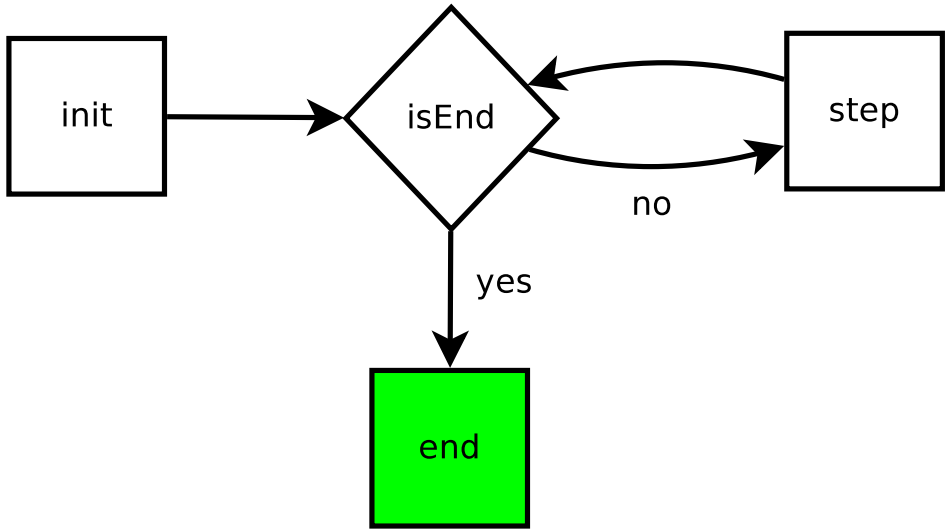
SGCB: Application Scenario (Example)



step

1. Send 10 more Computing Tasks to the master;
2. Fetch the results of already computed tasks;

SGCB: Application Scenario (Example)



end

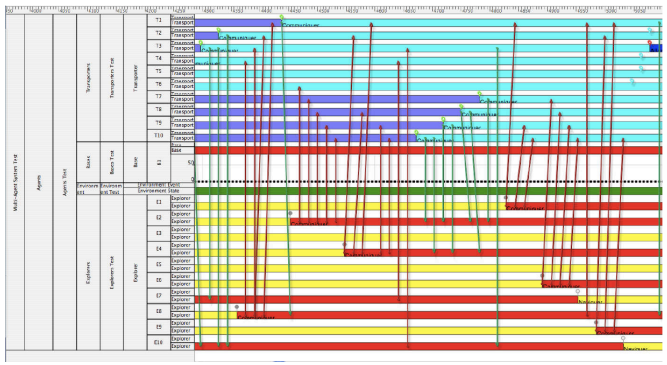
1. Stop the master and slaves;
2. Release the VM;
3. Remove the VM Image;

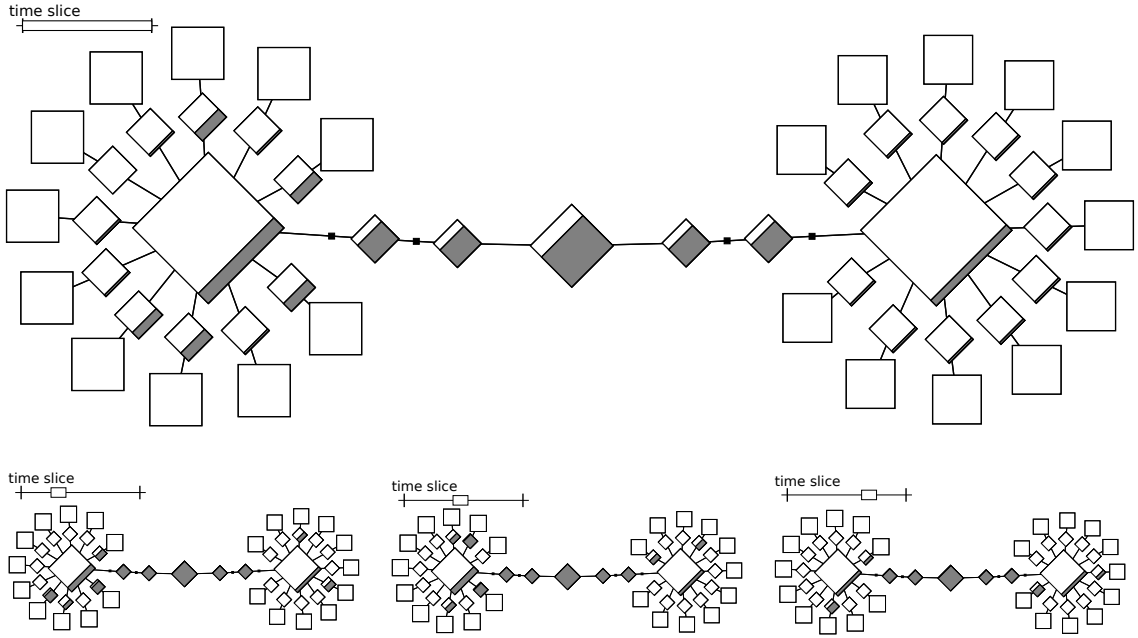
SGCB: Application Scenario (Example)

Post-processing

Analyze the trace:

- Pajé
- Viva
- R





Bag Of Tasks on Federated Clouds

- Analyze the trace generated by the simulator;
- Propose new provisioning and task allocation algorithms;
- Work on storage policy;
- See the impact of amount of input and output data on the bill;
- Take into account new users' requirements e.g. storage location;
- Work on real-world traces to simulate the Bag of Tasks;

SimGrid Cloud Broker

- Improve performance models (network, storage, etc.);
- Verify the SGCB results with real-world execution;
- Add basic multi-core simulation;

- Available on demand (soon on github);

Potential Collaboration: HPC on Clouds

Tightly coupled applications are the weakness of Clouds:

- Work on VM scheduling algorithms to improve their performances;
- Work on VM scheduling algorithms to reduce noise due to neightboors;
- Co-scheduling with Network as a Service to improve network for HPC;
- Co-scheduling with Storage as a Service to improve network for HPC;
- How to bill the new resources related to HPC;

Potential Collaboration: Scientific Workflow on Clouds

- Extend current Cloud Middleware to support scientific workflow;
- Propose a Broker for Cloud Middleware;
- Work on VM scheduling algorithms for cloud workflow inside a cloud;
- Work on provisioning mechanisms for cloud workflow in a federated cloud;

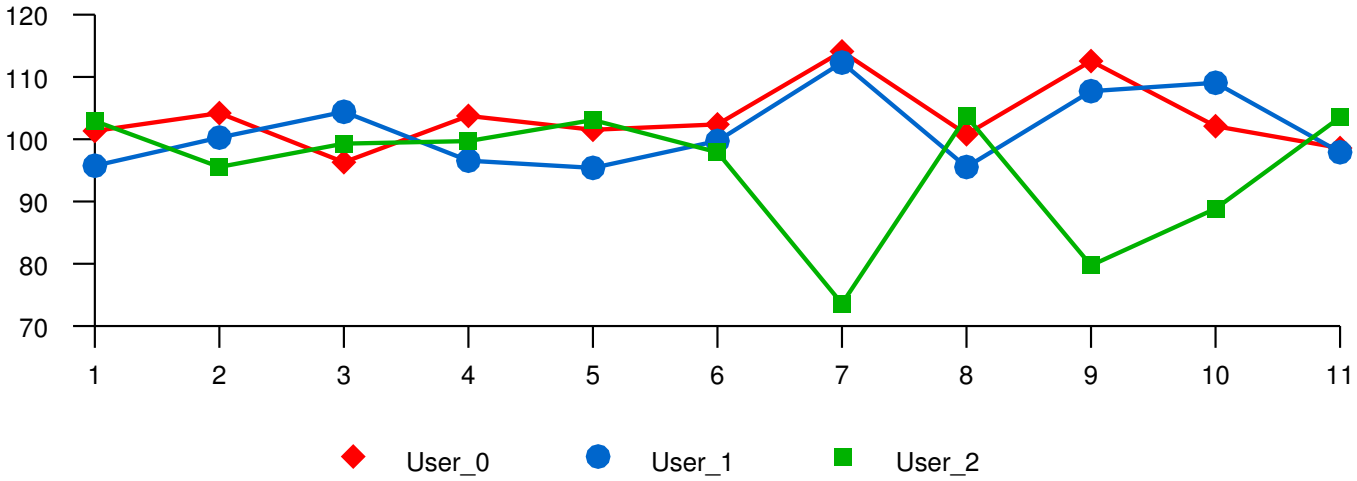
Potential Collaboration: Cloud Security

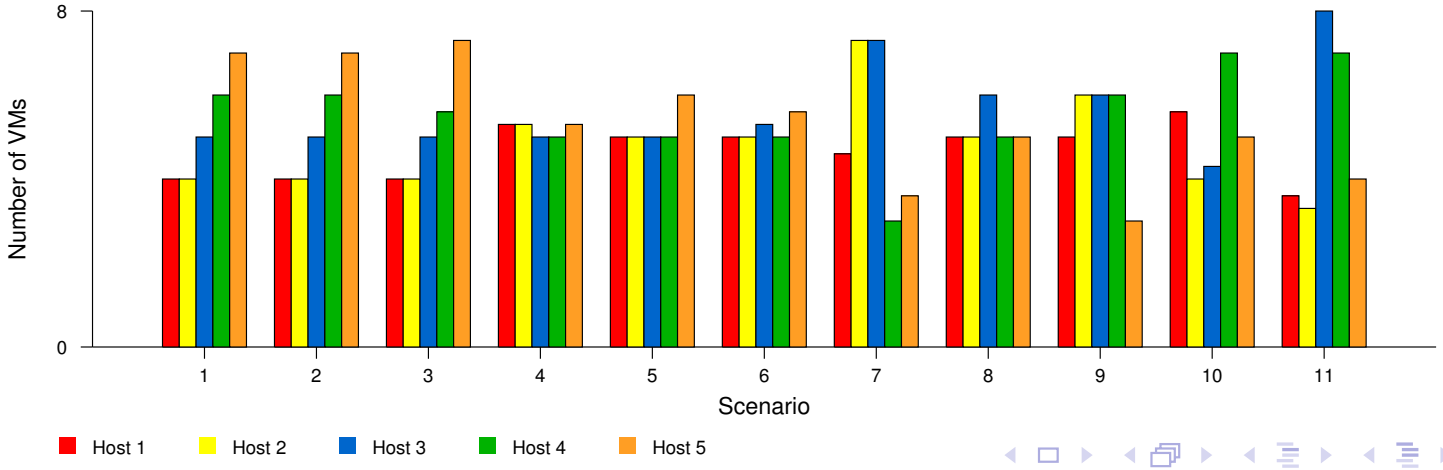
- European Project Celtic+ Seed4C;
- Create a highly secured Cloud;
- Leader of the demonstrator;
- Work on the HPC and Big Data use cases;
- Security on HPC, Big Data and Clouds: smart provisioning, self-configuration of security devices, access and information flow controls;

Potential Collaboration: Big Data on Clouds

- Work on VM provisioning algorithms to improve MapReduce performance in federated cloud;
- Work on VM placement algorithms to improve MapReduce performance inside a cloud;
- Efficiently used storage service to provide Big Data services in cloud;

Potential Collaboration: Big Data on Clouds





Potential Collaboration: Big Data on Clouds

	User 0		User 1		User 2	
	Isolation	# of VMs	Isolation	# of VMs	Isolation	# of VMs
1	1	21	1	21	1	21
2	1	14	1	14	2	7
3	2	7	2	7	2	7
4	1	21	1	21	1	21
5	1	14	1	14	2	7
6	2	7	2	7	2	7
7	1	14	1	14	3	7
8	2 with user 1 3 with user 2	7	2	7	2 with user 1 3 with user 0	7
9	1	14	1	14	3	7
10	2	7	2	7	3	7
11	3	7	3	7	3	7