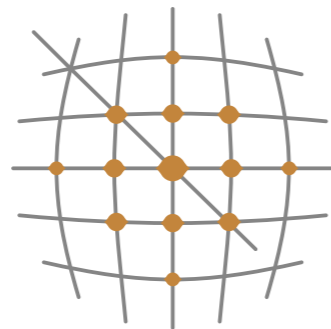
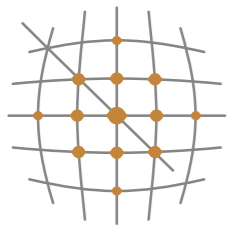


# Cloud infrastructure for the on demand provisioning of Worker Nodes

C.Triantafyllidis, P. Korosoglou, A.Andronidis, G. Fergadis, C. Kanellopoulos and P.Argyrakis

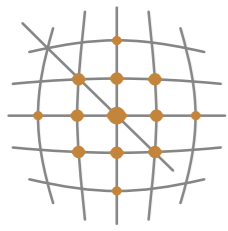


**Scientific Computing Center**  
Aristotle University of Thessaloniki



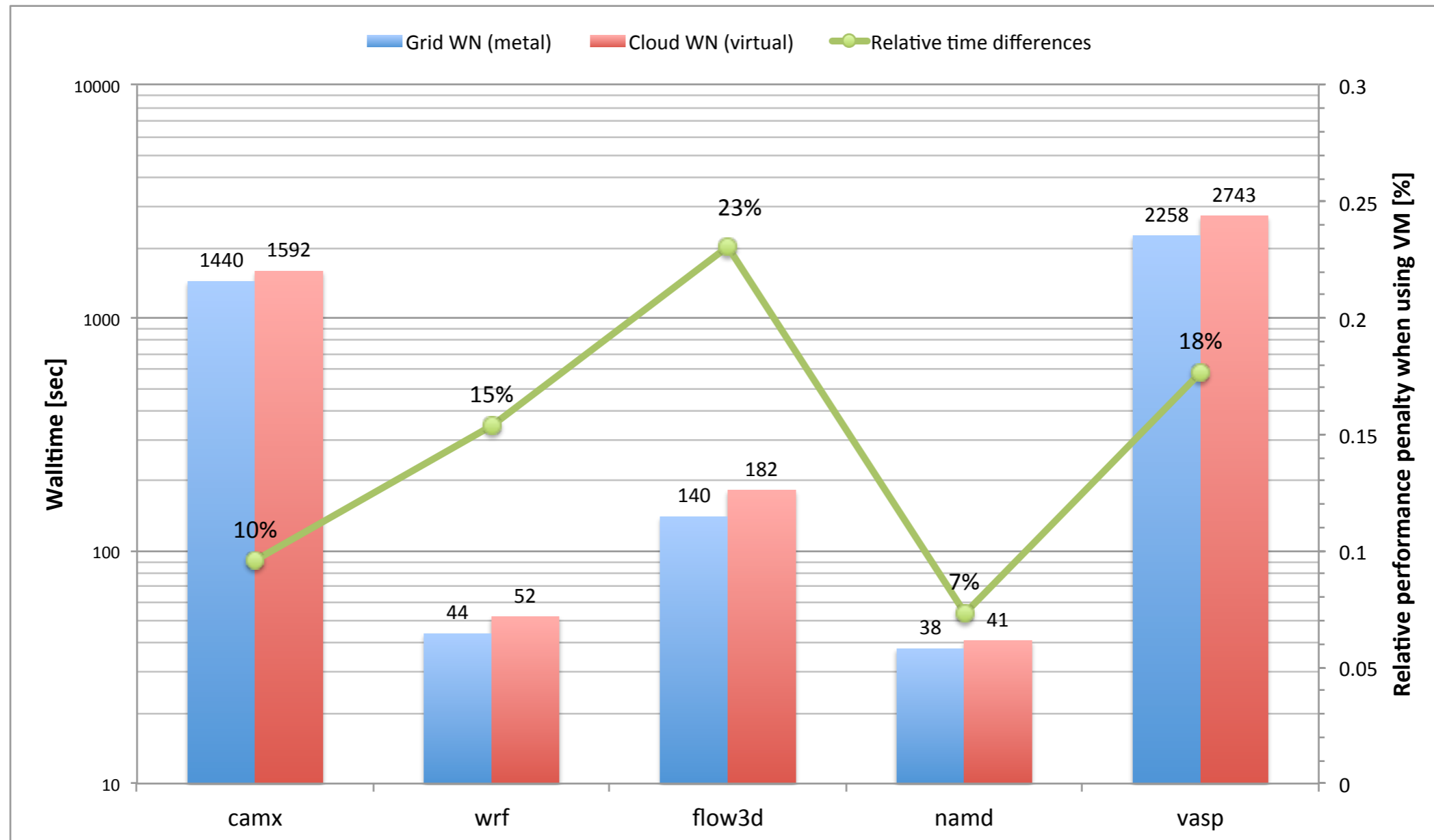
# Outline

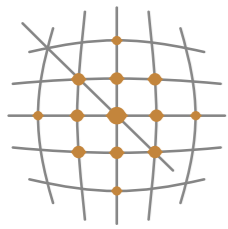
- Observation
- Idea
- Implementation
- Demo
- Source code



# Our observations (last Spring)

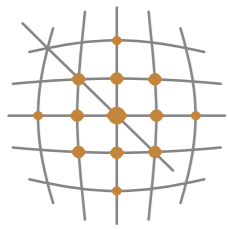
- Scientific applications performance penalty in the range 5%-25%





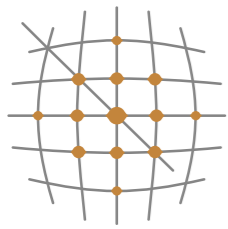
# Concept

- Create and save an image preconfigured to act as a WN
- Whenever a high demand on Computing Resources is at place:
  - Boot up a predefined number of Virtual WNs
  - Configure them appropriately under the CREAM service
- Once the demand is low again remove resources from the CREAM service and terminate instances



# Implications

- Which are the thresholds (i.e. when to boot up or terminate instances)
- Where to run the service (i.e. on the CREAM service)
- Image creation and templating
- Network connectivity & firewall configuration
- Instance sizes (wrt instance size options available)
- Yaim functions

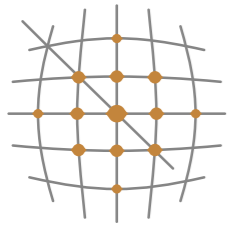


# Thresholds

- Simplistic approach
  - When jobs on queue exceed a pre-fixed number fire up Virtual WNs, else terminate them if any exist

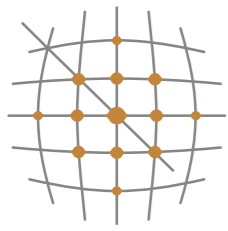
$(w(t) > a(t=0)) ? add(n(t)) : check\_n\_remove(n(t)) ;$

- $w(t)$ : waiting jobs at time  $t$
- $a(t=0)$ : available job slots at  $t=0$  (fixed number)
- $n(t)$ : Number of Virtual WNs to boot/terminate



# Image creation and templating

- EMI-2 based WNs
  - OS: CentOS 6.3 base image with small (EC2) size is sufficient
  - Additional repositories: EMI, EGI-Trustanchors
  - Additional packages:
    - gluster-fuse
    - emi-wn
    - emi-torque-client
  - Additional configuration:
    - munge key & service
    - Supported VO files & yaim configuration files
    - Post-boot scripts to configure instances via yaim (in `/etc/rc.local`)



# Networking and firewalling

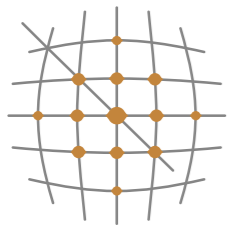
The image shows two screenshots from the OpenStack dashboard. The top screenshot displays the 'Security Groups' page with a table listing 'Torque-WN' and 'default'. A green button 'Create Security Group' is visible. The bottom screenshot shows the 'Launch Instance' form with the 'Torque-WN' security group selected in the 'Security Groups' dropdown. An orange arrow points from the 'Create Security Group' button in the top screenshot to the 'Torque-WN' selection in the bottom screenshot.

Name	Description	Actions
Torque-WN	Open ports for Torque server-client communication	Edit Rules Delete
default	default	

Quota Name	Limit
RAM (MB)	512MB
Floating IPs	10
Instances	10
Volumes	10
Gigabytes	1000GB

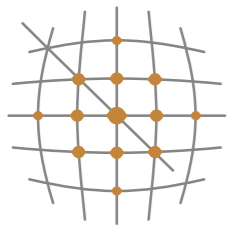
- Additional security group
- Add new instances to appropriate security groups



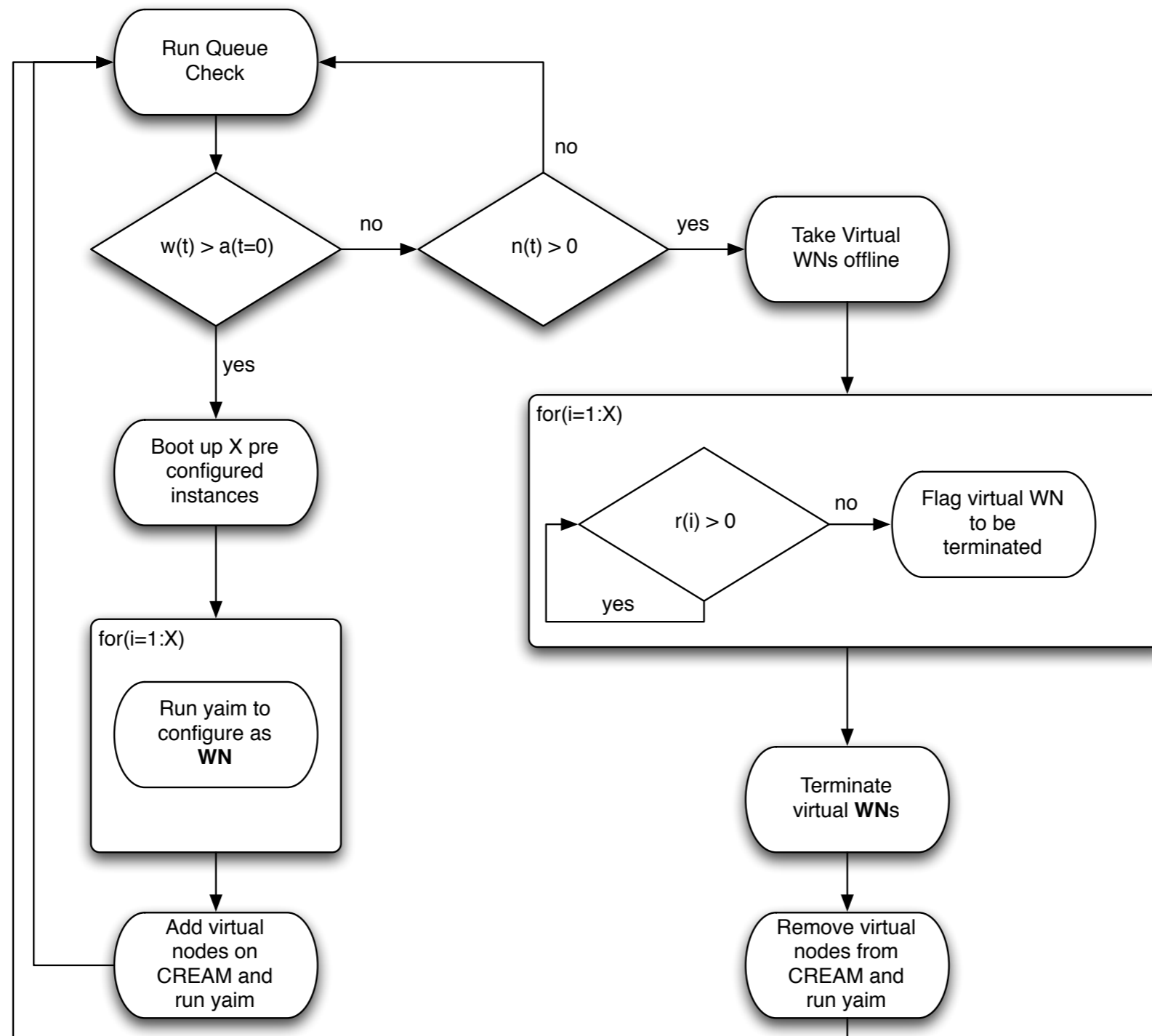


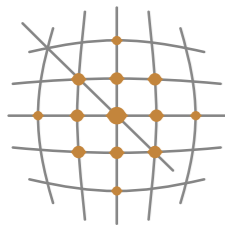
# Where to run the service

- No need to induce further load on CREAM instance
- Service may run as a cron job on any other machine that can access
  - using ssh the
    - CREAM service and the
    - Virtual WNs (instances)
  - using REST API the
    - Nova compute node




# How it all comes together








# More on...


## <https://github.com/auth-scc/openstack-scaler>


**PUBLIC**  **auth-scc / openstack-scaler** Pull Request Unwatch Star 0 Fork 0

**Code** Network Pull Requests 0 Issues 0 Wiki Graphs Admin



This project initialize VMs from Openstack to be added as worker nodes to Grid. It configures Cream and restarts YAIM. — [Read more](#)

 **Clone in Mac**  **ZIP** **HTTP** **SSH** **Git Read-Only** <https://github.com/auth-scc/openstack-scaler.git>  **Read+Write access**








 branch: **master** **Files** **Commits** **Branches 1** **Tags** **Downloads**

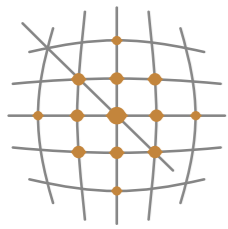
 Latest commit to the **master** branch

typo

 **andronat** authored 16 hours ago  commit cdea6b1a3d

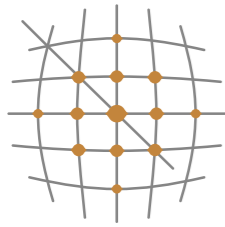
**openstack-scaler /**

name	age	message	history
 <a href="#">README.md</a>	3 days ago	Initial commit [andronat]	
 <a href="#">cream_handler.rb</a>	19 hours ago	error handling [andronat]	
 <a href="#">demo_scaler.rb</a>	3 days ago	Publishing first code. [andronat]	
 <a href="#">openstack_handler.rb</a>	16 hours ago	semi-final version [andronat]	
 <a href="#">scaler.rb</a>	16 hours ago	typo [andronat]	
 <a href="#">scaler_config.rb</a>	21 hours ago	fix in config [andronat]	
 <a href="#">vm_handler.rb</a>	21 hours ago	time tunning [andronat]	

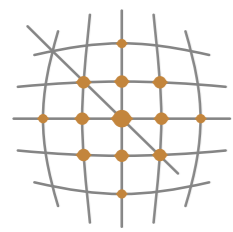


# Future work

- Develop threaded version of check mechanism that checks if a virtual WN has been emptied or not
- Enable re-usage of offline virtual nodes
- Re-work on thresholds based on production usage of resources



# Thank you!



# Short Demo