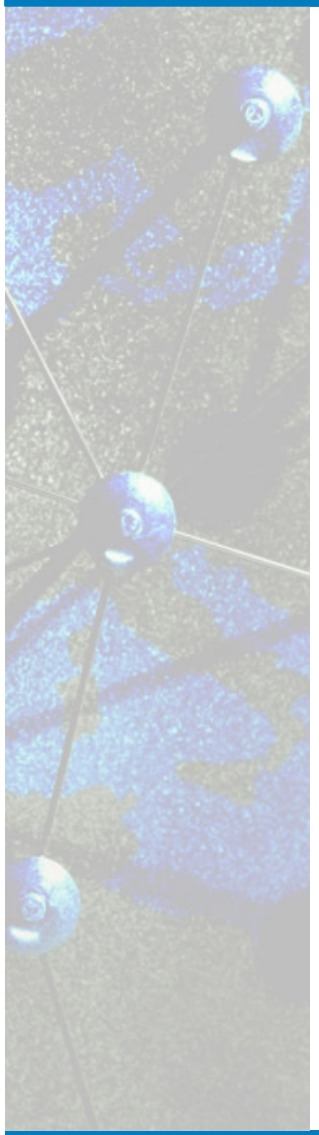


Quality Evaluation on the Grid and Requirements

Developing Quality Metrics aimed at enhancing Virtual Community Sustainability and Resources Ranking

COMPCHEM Virtual Organization

Dr. Carlo Manuali <carlo@unipg.it>
Department of Chemistry, Biology and Biotechnologies
University of Perugia (IT)

e-infrastructure

Introduction

The **COMPCHEM** VO has designed a **Grid Credit System named GCreS** to be adopted as a tool to enhance **Sustainability** for general Grid Communities, as in the case of Virtual Research Communities (VRC)s.

GCreS provides the **Evaluation of both the work performed by Users and the Services offered by VRCs** in Grid by implementing **new global Quality Parameters** (**QoS, QoU, QoP and QoC**) **and related metrics**.

GCreS runs **on top of a Service-oriented Grid Framework** (*like for example either GriF or WS-PGRADE*) **and converts the information provided** (by it) **into Credits and Costs**.

Quality of Service (QoS)

To estimate **Costs**, **Quality of Service** (**QoS**) **parameters have been applied to Grid Services.**

At present, we use the following **QoS Parameters**:

- **Accessibility** (S_{acc})
- **Integrity** (S_{int})
- **Performance** (S_{per})
- **Reliability** (S_{rel})
- **Security** (S_{sec})

Also adopted for implementing the RANKING feature

$$\text{QoS Metrics} = w_0 S_{acc} + w_1 S_{int} + w_2 S_{per} + w_3 S_{rel} + w_4 S_{sec}$$

Quality of User (QoU)

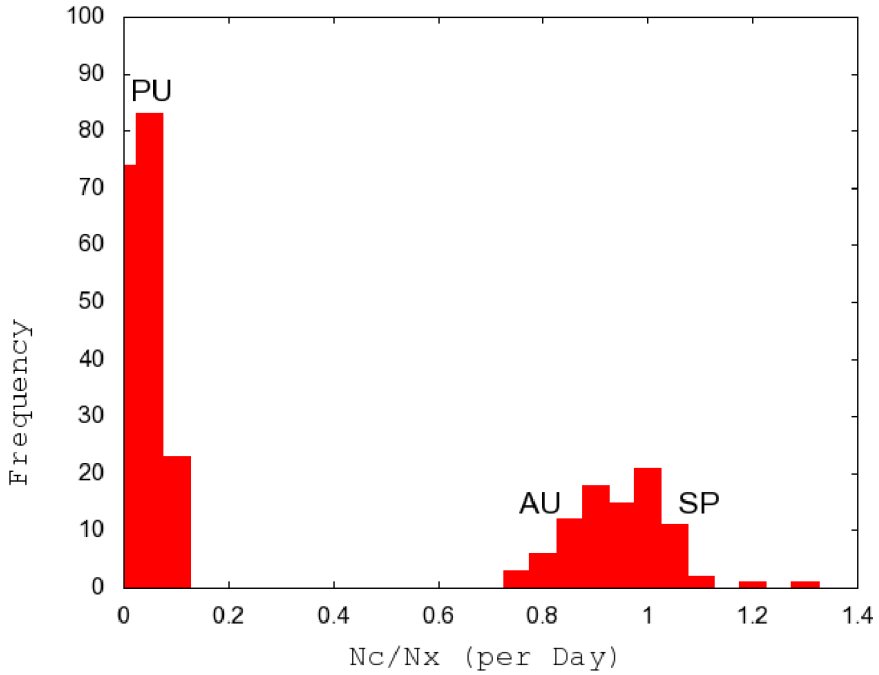
To estimate **Credits**, **Quality of User** (**QoU**) **parameters have been applied to the Grid Users**. At present, we use the following **QoU Parameters**:

- U_{cx} the ratio between the number of successful compilations and the number of related executions (**Profile Information**)
- U_{cu} the percentage of custom runs multiplied the sum of the number of custom runs and the number of runs derived from the already available applications weighted respectively by the ratio between the number of related results retrieved and the number of related results available (**Qualitative Information**)
- U_{cm} the number of results retrieved multiplied the weighted sum of the average cpu time elapsed and the average virtual memory amount consumed for all the related jobs (**Quantitative Information**)
- U_{ge} the ratio between both the sum of cpu time and wall time for all the 'Done' jobs belonging to a given user (**Grid Efficiency**)
- U_{fb} the number of feedbacks produce by users (**Active Filtering**)

$$\text{QoU Metrics} = p_0 w_5 U_{cx} + p_1 w_6 U_{cu} + p_2 w_7 U_{cm} + w_8 U_{ge} + w_9 U_{fb}$$

Results

1. **Services** provided by a given Grid Community **can be ranked and costs associated with their use** drafted on the basis of the **Quality of Service (QoS)**



2. **Users** belonging to a given Grid Community **can be rewarded for their work by credits** on the basis of the (*profile-dependent*) **Quality of User (QoU)**

3. **This fosters the development of a Grid Economy Model** enhancing **Sustainability**

QoU Requirements

For each User, the underlying Framework has to gather (on a MySQL DBMS, for example) **the following qualitative and quantitative information on which the QoU metrics (adopting *Collaborative – passive and active - Filtering*) is based:**

1. **Number of runs corresponding to those applications already made available** by the Framework;
2. **Number of runs corresponding to new programs provided** by the user;
3. **Number of results generated** from 1) and 2), respectively (**or failures**);
4. **Number of results accessed** from 1) and 2), respectively;
5. **Average cpu & wall time elapsed** for runs generating results and having results accessed, respectively;
6. **Average memory consumed** for runs generating results and having results accessed, respectively;
7. **User Feedbacks** (*active filtering*).

QoU Use Case

For each grid user, or a group of them, the underlying grid layer has to offer the possibility to retrieve (e.g. by portlets and command line)**, for a given period, various qualitative and quantitative information on which the QoU metrics is based.** For example:

- **The number of results generated and accessed from those runs corresponding to applications already made available by the portal accessed by users**.

***Example**:*

- In the year *2013* the user *Bob* performed in total 20 runs, 15 corresponding to program A, 5 to B. A failed 7 times, B never failed. Then *Bob* has accessed all the available results from A, just one from B.
- In *2013* for *Bob* we have:

  - for Prog. A: 8 out of 8 successes;

  - for Prog. B: 1 out of 5 successes.

QoS Requirements (1/2)

For Calculating QoS Metrics & Ranking

- Registering for each Grid Job/Subjob

1. **Username** owning the job
2. **Grid service name** and **Grid site name** used
3. **Job Type** (e.g. Single, Parameter Study or Workflow)
4. **Submission/Ending date**
5. **Program name** and **Input** provided (if any)
6. **Results address** (e.g. Storage Element reference)

(specifically for Ranking Computing Elements)

7. **CE address** (e.g. queue name)
8. **CE Exit status** (e.g. 0 -> done with success; 1 -> CE error)
9. **Cpu/Wall time**
10. **Mem consumed**

QoS Requirements (2/2)

For Calculating QoS Metrics & Ranking

- ***Registering for each Grid Service***

1. **Name**
2. **Description** and other information (optional)
3. **Grid site name** location
4. **Maintainer / Manager**
5. **Number of Functions** (e.g. Web Services composing the Grid Service)
6. **Number of Errors**
7. **Elapsed (averaged) time for satisfying (initiating) a call**
8. **Elapsed (averaged) TTR (Time-to-Repair) for repairing an error**
9. **QoS** (calculated)
10. **Cost** (assigned)

Q/A

Thank you for your attention.