A Trust-based Approach for a Competitive Cloud/Grid Computing Scenario

F.Messina[1], G.Pappalardo[1], D.Rosaci[2],
C.Santoro[1], G.M.L.Sarné[2]

[1]Dipartimento di Matematica ed Informatica
Università di Catania, ITALY

[2]DIMET, Università "Mediterranea" of Reggio Calabria, ITALY

IDC 2012 - 6^{th} International Symposium on Intelligent Distributed Computing
Calabria, Italy - Sept. 24-26, 2012

Outline

Background and motivation

Context: a competitive grid/cloud computing scenario.

- Resource are assigned on-demand
- A price is payed for the services offered
- The computing nodes compete for the assignment of a certain job/request

To obtain a job, a node *may cheat on* in announcing its resource availability and then request the collaboration of other nodes.

To select the nodes to collaborate with, the evaluation of their *trust degree* must be performed.

System Model and Interaction Protocol (1)

1. Very large-scale cloud/grid computing environment (in the order of 10K - 100K nodes).
2. Some special nodes, called *Task Allocators* (*TA*), have the responsibility of receiving client's requests and selecting the most appropriate node (say A) on the basis of a *trust model*.
3. Node A could not possess adequate resources but, to obtain the job, may ask the collaboration of/buy resources from other nodes.
4. ...

System Model and Interaction Protocol (2)

3. ...
4. Node A starts a *resource finding protocol* in order to obtain the set of candidate nodes for collaboration.
5. For each node of the set, say B, node A asks a third node, say C, the information about the *reputation* of B; on this basis, A selects the nodes to collaborate with.
6. Selection of a node n is performed by A with a *preference coefficient* $P_A(n)$ computed using the reputation protocol.
7. When the service is provided, the TA asks the client a *feedback* which is, in turn, forwarded to A so that it can use the information to update the *trust degree* of collaborating nodes.

Reliability-Reputation Model (1)

- $SR_i(j)$, service reliability that node i assigns to j.
- $RR_i(j)$, recommendation reliability that node i assigns to j.
- $R_i(j)$, reputation that node i assigns to j.
- $RECC_i(j,k)$, recommendation provided to i by j about k.
- $FEED_i(s,j)$, quality of collaboration (feedback) provided to i by j about service s.

These functions/maps are in the range $[0,1]$ (0=minimum, 1=maximum).

Reliability-Reputation Model (2)

Phase 1: Recommendation reception. $RECC_i(j,k)$ is updated accordingly.

Phase 2.1: Computation of SR (service reliability)

- Let $Services_i(j)$ the set of services provided to i by j at previous step;
- Compute $SR_i(j) = \alpha \cdot SR_i(j) + (1-\alpha) \cdot \frac{\sum_{s \in Services_i(j)} FEED_i(s,j)}{||Services_i(j)||}$, with $\alpha \in [0,1]$ the *update rate* of $SR_i(j)$.

Reliability-Reputation Model (3)

Phase 2.2: Computation of RR (recommendation reliability)

- Compute
  $rr_i(j) = \frac{1}{||Nodes(j)||} \sum_{k \in Nodes(j)} \frac{\sum_{s \in Services(k)} (1 - |RECC(j,k) - FEED(s,k)|)}{||Services(k)||}$
- Update $RR_i(j) = \alpha \cdot RR_i(j) + (1 - \alpha) \cdot rr_i(j)$

Phase 3: Computation of R (reputation)

- $R_i(j) = \frac{\sum_{k \in NODES, k \neq i} RECC_i(k,j) \cdot RR_i(k,j)}{\sum_{k \in NODES, k \neq i} RR_i(k,j)}$

Phase 4: Computation of P (preference)

- $P_i(j) = \beta \cdot SR_i(j) + (1 - \beta) \cdot R_i(j)$ with $\beta \in [0,1]$ the *service reliability vs. recommendation reliability weight*.

$P_i(j)$ is then used by node i to decide whether selecting node j for collaboration.

Resource Finding (1)

Very large environment (100K nodes): Resource finding may be hard!

SW-HYGRA: *Small-World HYperspace Grid Resource Allocation*

- Decentralised self-organising approach
- Resource are coordinates of a virtual *hyperspace*;
- Each node, on the basis of resource amount availability, is a *point* in the hyperspace;
- A *metric*, based on Euclidean distance, is defined;
- An *overlay network* is constructed;
- *Resource Finding* is preformed by surfing the overlay network.

SW-HYGRA: Overlay Construction

- **Algorithm**
  1. Each node contacts its linked nodes in order to obtain the set of *2-hop linked* nodes;
  2. The set is ordered on the basis of the Euclidean distance;
  3. Each node, on the basis of resource amount availability, is a *point* in the hyperspace;
  4. Connections of n are rearranged in order to create **short links** with probability p_s and **long links** with probability p_l;
  5. As a result, nodes featuring similar resource availability form *clusters* interconnected by short links, while long links interconnects clusters between them.
- The resulting structure resembles a *small-world network*.
- Since a node may change its resource availability, the algorithm runs continuously.

SW-HYGRA: Resource Finding

- **Algorithm**
  1. A resource request q is submitted to a node n of the network;
  2. If n is able to fulfill q the algorithm terminates;
  3. Otherwise, n contacts its linked nodes and chooses the one with the smallest euclidean distance to q.

- The request "surfs" the network through long links to reach the cluster where the target node resides.
- Then it surfs the net into the cluster, by using short links, in order to find the target node.

F. Messina, G. Pappalardo, C. Santoro. *Exploiting the Small-World Effect for Resource Finding in P2P Grids/Clouds*, Proc. of the 20th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2011).

F. Messina, G. Pappalardo, C. Santoro. *Decentralised Resource Finding in Cloud/Grid Computing Environments: a Performance Evaluation*, Proc. of the 21th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2012).

Preliminary Experiments

We used a C-based simulator (ComplexSim) developed by the authors to simulate the proposed schema.

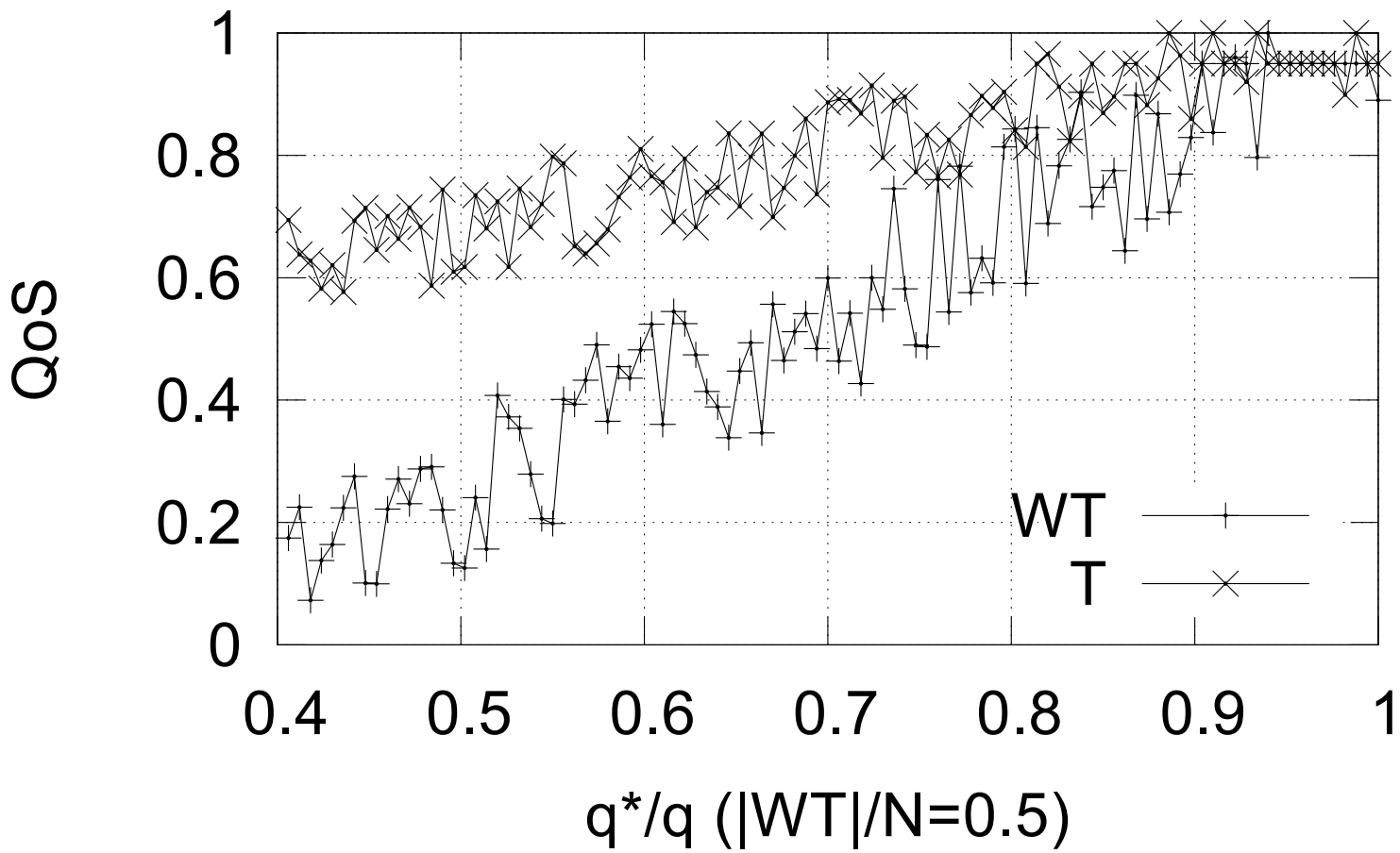
Parameters:

- q the number of resources *declared* by the single node
- q^* the number of resources *actually offered* by the single node
- N number of nodes $= 10^5$
- T set of nodes **with** trust model
- WT set of nodes **without** trust model

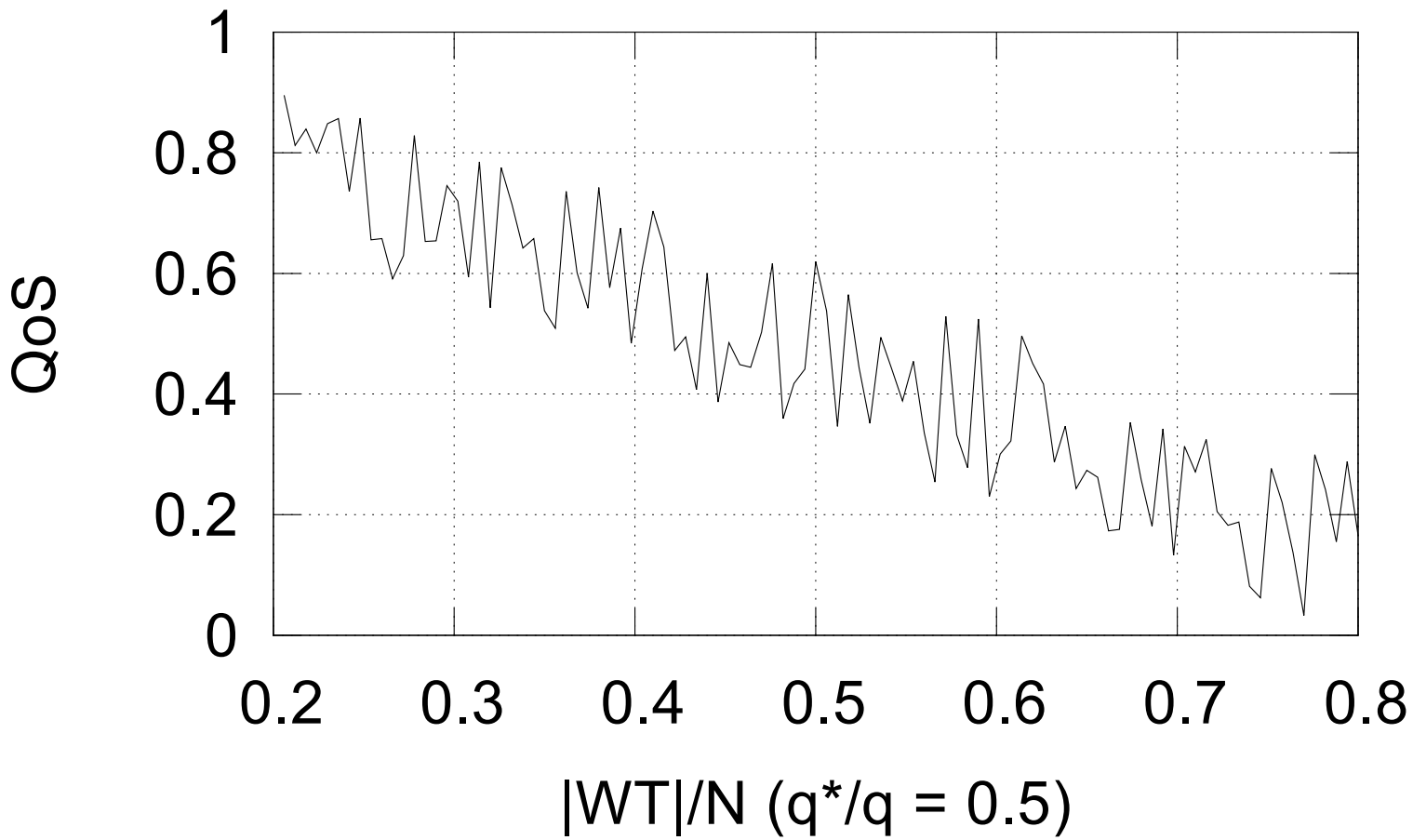
Measure:

- QoS perceived by the client

Preliminary Experiments: QoS vs. Node capacity



Preliminary Experiments: QoS vs. Number of Nodes without Trust Model



Conclusions and Future Work

- We proposed a trust model for large scale competitive grids/clouds.
- The use of SW-HYGRA allows a fast finding of interlocutors.
- Preliminary studies prove the effectiveness of the model.
- Further work is needed to simulate more realistic scenarios to better understand the trend of trust dimensions, role of coefficients, etc.

A Trust-based Approach for a Competitive Cloud/Grid Computing Scenario

F.Messina[1], G.Pappalardo[1], D.Rosaci[2],
C.Santoro[1], G.M.L.Sarné[2]

[1]Dipartimento di Matematica ed Informatica
Università di Catania, ITALY

[2]DIMET, Università "Mediterranea" of Reggio Calabria, ITALY

IDC 2012 - 6^{th} International Symposium on Intelligent Distributed Computing
Calabria, Italy - Sept. 24-26, 2012