



SORMSYS: Towards to a Resource Management Platform for Self-Organizing Large Scale Distributed Systems

Florin Pop

Computer Science Department, Faculty of Automatic Control and Computers,

University POLITEHNICA of Bucharest, Romania

Email: florin.pop@cs.pub.ro

Research Motivation

- Optimization of resource management in large scale distributed systems (LSDS).
- Self-organization capability => adaptability, fault tolerance assurance, QoS.
- Decentralized architecture
 - Resources from different systems: Grid, P2P, and Cloud.
- Resources management in LSDS
- Heterogeneity of LSDS
 - Grid and Massive Parallel Machine
 - P2P and Mobile Systems
 - Cloud

Project main goal

The SORMSYS main goal is to optimize resource management in large scale distributed systems with the capability of self-organization.

- Decentralized architecture based on existing middleware solutions for distributed systems.
- Design of algorithms and methods inspired by natural models.
- The important components considered for the architecture are:
 - resources allocation
 - task scheduling
 - resources monitoring
 - resource discovery

Outline

- Overview of Large Scale Distributed Systems
- Resource Management in LSDS
- SORMSYS proposed architecture
- Monitoring process for resources involved in SORMSYS
- System evaluation
- Conclusions and Future Work



Overview of Large Scale Distributed Systems

- **Grid Computing:** distributed paradigm dedicated to high performance computing that brings together resources from different organizations in order to facilitate the collaboration of a group of people.
 - Data Grid
 - eScience Grid
 - Enterprise Grid
 - Autonomic Grid
 - Knowledge Grid
 - Economy Grid

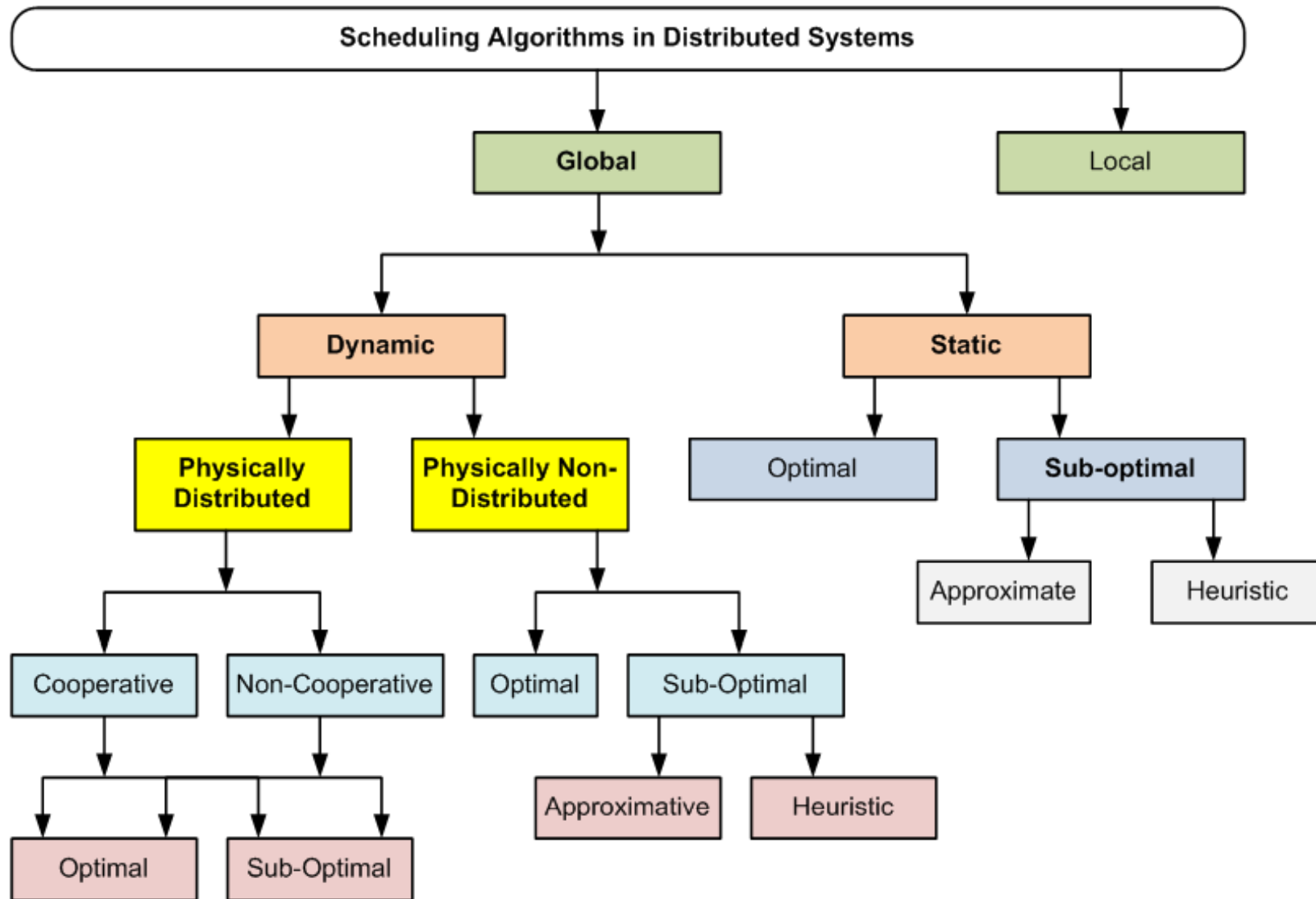
Overview of Large Scale Distributed Systems

- **Cloud Computing Systems:** a set of network enabled services, providing scalable, QoS guaranteed, normally personalized, inexpensive computing platforms on demand, which could be accessed in a simple and pervasive way.
 - Hardware as a Service
 - Software as a Service
 - Data as a Service
- The main features that distinguish cloud computing:
 - User-centric interfaces
 - On-demand service provisioning
 - QoS guaranteed
 - Autonomous system
 - Scalability and flexibility

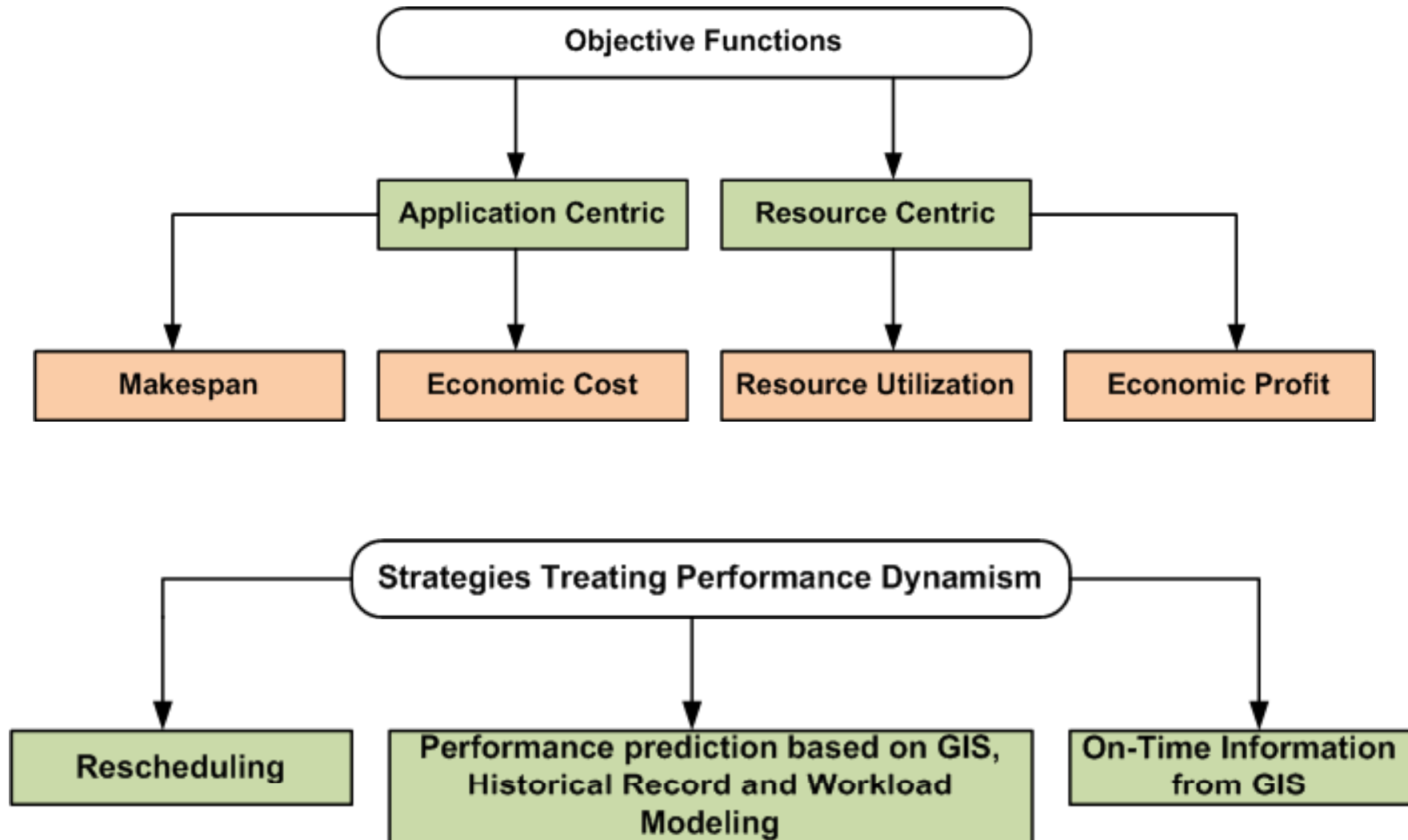
Overview of Large Scale Distributed Systems

- P2P Computing Systems: distributed systems with no centralized control, where each node accomplishes the same functionality.
- New forms of paradigms have appeared:
 - B2B (Business to Business)
 - B2C (Business to Consumer)
 - B2G (Business to Government)
 - B2E (Business to Employee)
 - ...

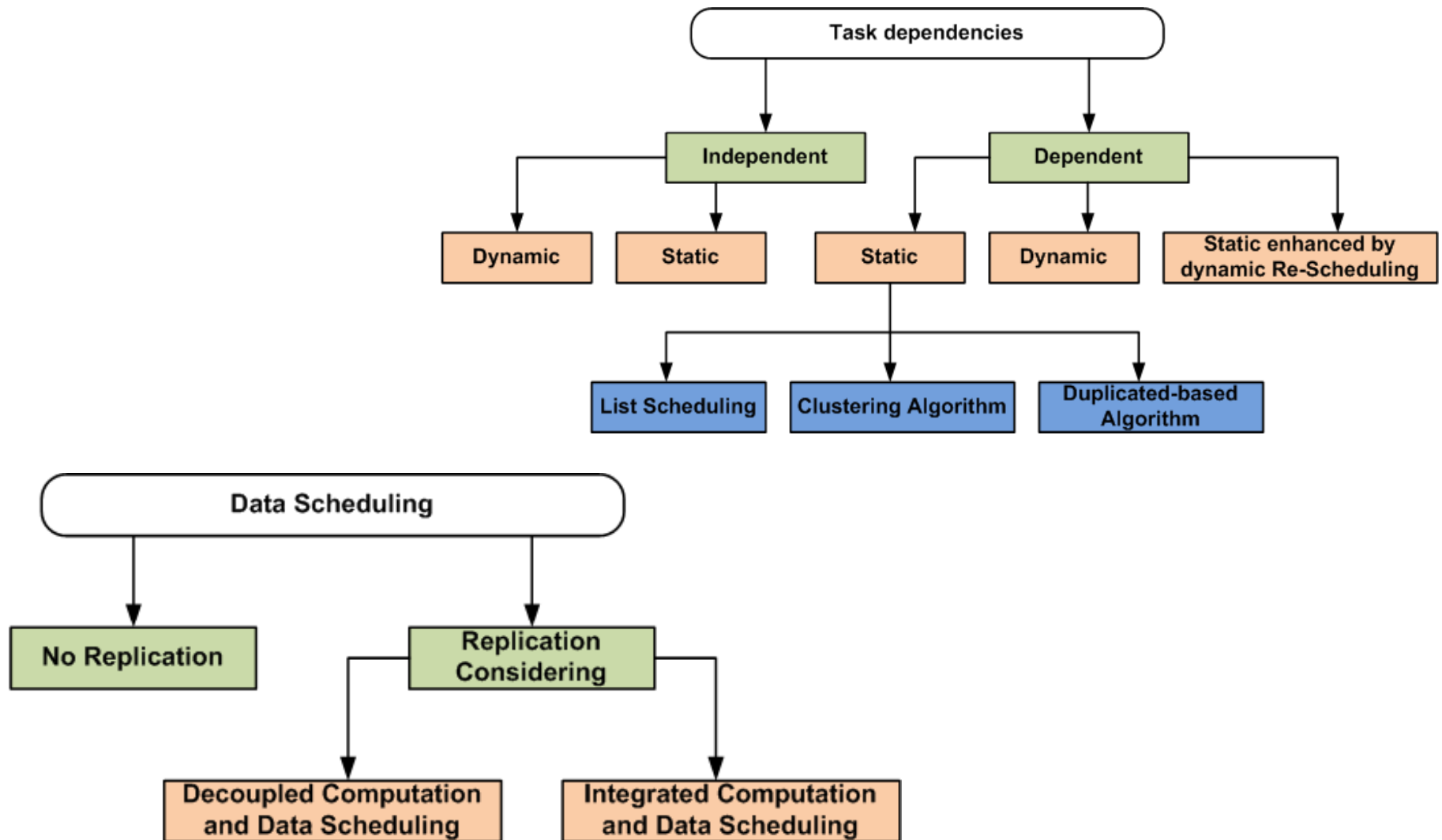
Resource Management in LSDS



Resource Management in LSDS



Task Management in LSDS



SORMSYS Components

- Virtual Controller Entity (**VCE**)
 - software running on each machine of the system
 - interacts with other similar entities in order to ensure an adaptive and self-organizing environment.
 - implementation is based on bio-inspired and natural models.
- Virtual Middleware (**VM**)
 - guarantees the collaboration and interoperability between VCEs
 - provides communication channels and protocols
- Virtual Connectors (**VC**)
 - provides the proper solutions for different types of distributed systems, according to their architectural structure and requirements: Grid, P2P, Massive Parallel Machine (MPM) or Cloud.

Virtual Middleware in SORMSYS

Ensure communication and interoperability between high level entities defined as VCEs that manage and control the whole system.

- Responsible for providing a self-organizing environment.
- It can be viewed as an extension of the actual middleware.
- Provides connectors to the current available distributed systems technology: Cloud, Grid, P2P and MPM.
- Offered services:
 - decentralized gossip-based algorithms for resource monitoring
 - bio-inspired algorithms for resource management optimization

Resources Management in SORMSYS

- It is impossible to gather and maintain a detailed and up-to-date list of all nodes participating in any large computation.
- => The solution comes from gossip-based algorithms
 - excellent behavior
 - provide very good convergence and resilience properties
 - react to continuous changes across the system
- First achievement in SORMSYS research:

Gossip-based algorithm capable to monitor the nodes in terms of their performance, availability and reliability.

SORMSYS MONITORING

- Nodes Evaluation => ARP-property
 - Availability (A) - a node is considered available if it is not executing a task or if its current load is not too intensive ($0 \leq A \leq 1$).
 - Reliability (R) - it is critical to have an evaluation of the nodes regarding their level of dependability, because unexpected failures must be avoided during a computation task
 - Alive ($R=1$)
 - Dead ($R=0$)
 - Transit ($0 < R < 1$)
 - Performance (P) - this parameter reflect the node's physical resources (CPU, memory, channels) and has three thresholds: High, Medium and Low ($0 \leq P \leq 1$).
- The node's evaluation -> $f(A, R, P)$

Monitoring Steps in SORMSYS

- **Step 1.** First, a certain node, randomly selects a gossiping partner, packs its knowledge existing in the gossiping table into a message labeled with *Gossip Request Message* and sends it to the selected peer.
- **Step 2.** For a predefined timeout, the node waits the reply from its partner. Based on the reply time, the initiator will evaluate the availability according to a certain pre-established threshold. In the mean time, the node can also receive other gossiping requests from its neighbors’.
- **Step 3.** When the timeout of current round expires, the node will have to handle the gossip tables received from its neighbors’ during gossiping exchanges. It will solve the occurred conflicts, by storing in its local gossip table only the last modified records. After finalizing these updates, a new round can be performed.

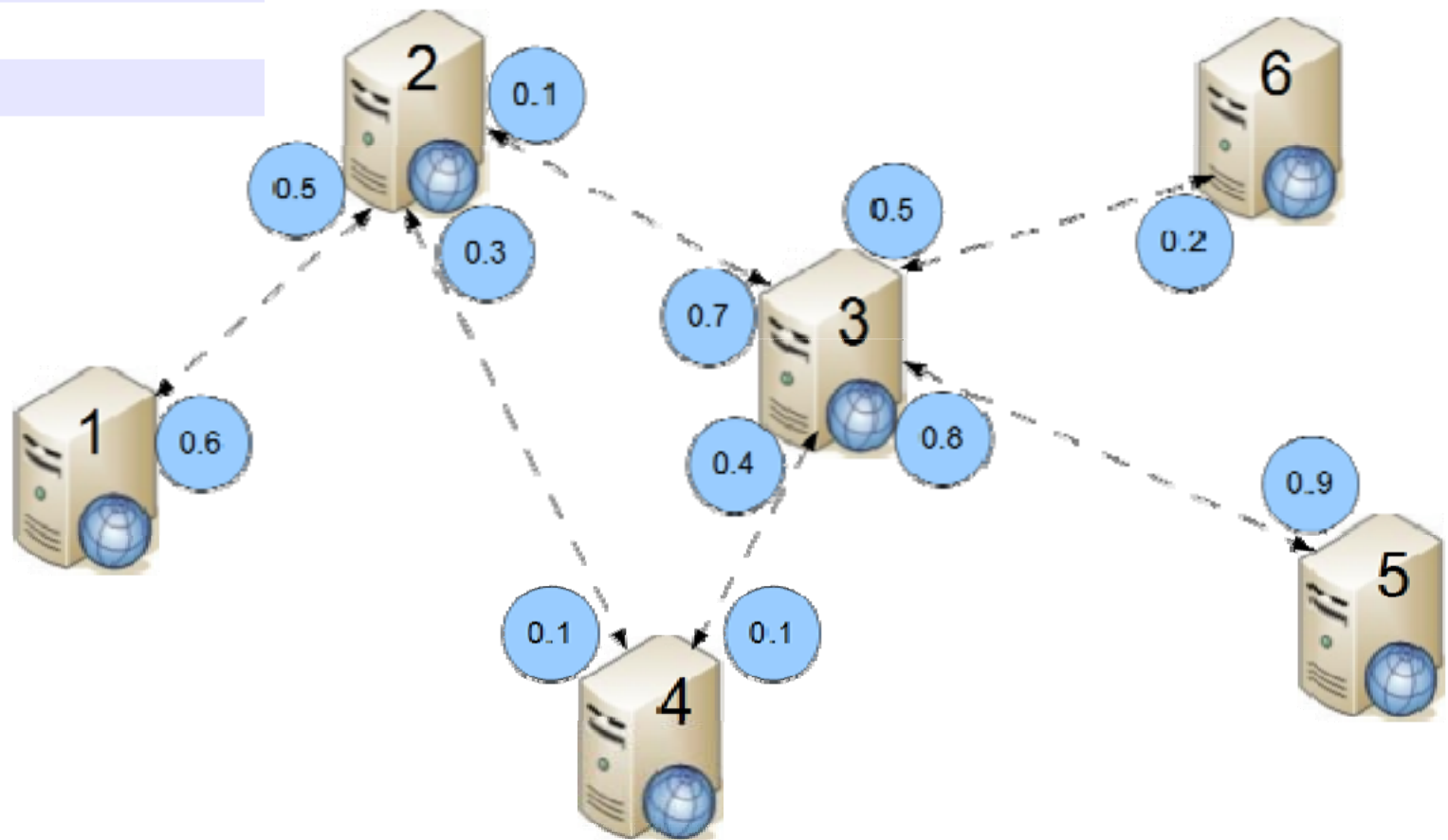
Local ARP Estimation

- Eliminates arbitrary selection of nodes to become gossiping partners.
- Each node will associate to its neighbors a confidence parameter expressing the capacity of each subset of nodes dominated by each neighbor.
- A node will locally store its confidence values associated for each connection it has with its neighbors.

$$\mathit{confidence}(P, N) = \frac{1}{n} \sum_{\mathit{child} \neq P} \mathit{confidence}(\mathit{child})$$

Example of ARP Evaluation of Nodes

PEER	ARP
1.	0.5
2.	0.6
3.	0.7
4.	0.3
5.	0.2
6.	0.2



Conclusion and future work

- Developing the Virtual Middleware proposed by SORMSYS project highlight a new approach over the infrastructure of Large Scale Distributed Systems.
- SORMSYS project uses bio-inspired methods that are meant to ensure resource management and system monitoring.
- Implementation and testing in a real environment the components of SORMSYS architecture:
 - VCE – Virtual Controller Entity
 - VC – Virtual Connector
 - VM – Virtual Middleware

Q&A

florin.pop@cs.pub.ro

