Second International Workshop on Simulation and Modelling
in Emergent Computational Systems (SMECS-2009)

September 22-25, 2009, Vienna, Austria

New Trends in Large Scale Distributed Systems Simulation

Ciprian Dobre, Florin Pop, Valentin Cristea
ciprian.dobre@cs.pub.ro

Faculty of Automatics and Computer Science
University POLITEHNICA of Bucharest
Romania

Outline

- Taxonomy to analyze large scale distributed systems simulators
- A critical analysis of simulation tools for large scale systems
- Future trends
- Conclusions

The Grid influences on the design of the simulation framework

- The simulation model proposed by MONARC 2 need to include the necessary components for simulating various distributed systems technologies, with respect to their specific components and characteristics
- The model should include the necessary components to describe various actual distributed system technologies,
  - ➔ It already provides mechanisms to describe concurrent network traffic, evaluate different strategies in data replication, analyze job scheduling procedures, etc.
- An important characteristic of the proposed simulation model is its generality

- **http://monarc.cacr.caltech.edu**

A new taxonomy? Do we need it?

- Most comparisons are not objective
  - ➔ papers published by authors in to highlight the quality of their tools in comparison with different existing simulation frameworks → tend to be misleading and subjective

- (1) proposed a taxonomy for **any** parallel and distributed systems simulators
  - ➔ Presence of physical time → a “must-be” property in case of large scale distributed simulators (network traffic, job processing models)

(1) A. Sulistio, C. S. Yeo, R. Buyya, “A taxonomy of computer-based simulations and its mapping to parallel and distributed systems simulation tools”, Software – Practice and Experience, 2004.

A new taxonomy? Do we need it?

- (1) proposed a taxonomy for **any** parallel and distributed systems simulators

  - ➔ Entity-based vs. event-based frameworks → all simulators nowadays use both: entities to simulate components such as clusters, processing units, network elements, and they use events to trigger the evolution of the simulated scenario

  - ➔ Validation not considered

  - ➔ Scope not considered

  - ➔ Supported model not considered

(1) A. Sulistio, C. S. Yeo, R. Buyya, "A taxonomy of computer-based simulations and its mapping to parallel and distributed systems simulation tools", Software – Practice and Experience, 2004.

A taxonomy of large scale distributed systems simulators (1)

Simulators for large scale distributed systems

1. Simulation model
2. Design and implementation

1. Simulation model
   - 1.1 Scope
   - 1.2 Supported model

1.1 Scope

➔ Many simulators designed for particular classes of (small) problems:

- Intranet or Internet systems, Web, Grid, Cloud, Farm, or Cluster systems, P2P networks, etc.
- Scheduling, data transfer, data replication technologies

A taxonomy of large scale distributed systems simulators (2)

1.1 **Scope** → Capabilities to simulate the layers of the distributed architecture:

- Host
- Network
- Middleware
- User applications

- Host characteristics
  - → Computing, data storage, other resources; grouped into single or distributed systems
  - → Of interest are their characteristics and organization
    - "Central model" proposed by Bricks
    - "Tier model" proposed by MONARC

A taxonomy of large scale distributed systems simulators (3)

- Network characteristics
  - ➔ Routers, switches and other devices
  - ➔ Communication protocols modeling
  - ➔ packet-level or stream-level model
- Middleware characteristics
  - ➔ Schedulers, security enforcement components, fault-tolerance solutions, etc.
- User applications
  - ➔ Jobs and applications within the distributed applications

A taxonomy of large scale distributed systems simulators (4)

1.2 **Supported model**

- ➔ Behavior – the “randomness” of the simulation
  - Deterministic
  - Probabilistic
- ➔ Time base – values that the simulation can contain
  - Discrete simulation
  - Continuous simulation

2. **Design and Implementation**

- ➔ Simulation engine
  - Execution
  - Mechanics

  - Performance of the queuing structures
  - How the simulator maps jobs on physical threads or processes

A taxonomy of large scale distributed systems simulators (5)

- **Mechanics** – how the simulation advances
  - ➔ Continuous
  - ➔ Discrete-event (DES)
  - ➔ Hybrid

- Furthermore:
  - ➔ Trace-driven DES
    - Input: a set of events collected independently from another environment
    - Advantage: suitable for modeling a system that has executed before in another environment
  - ➔ Time-driven DES
    - Advances by fixed time increments
    - Advantage: useful for modeling events that occur at regular time intervals
  - ➔ Event-driven DES
    - Advances by irregular time increments
    - Advantage: useful for modeling events that may occur at any time

A taxonomy of large scale distributed systems simulators (6)

- **Execution**:
  - ➔ Centralized simulators
  - ➔ Distributed simulators

- **Model specification**: a simulator can incorporate
  - ➔ Specialized languages
  - ➔ General programming languages and specialized library routines
  - ➔ Some visual components (drag&drop for model construction)

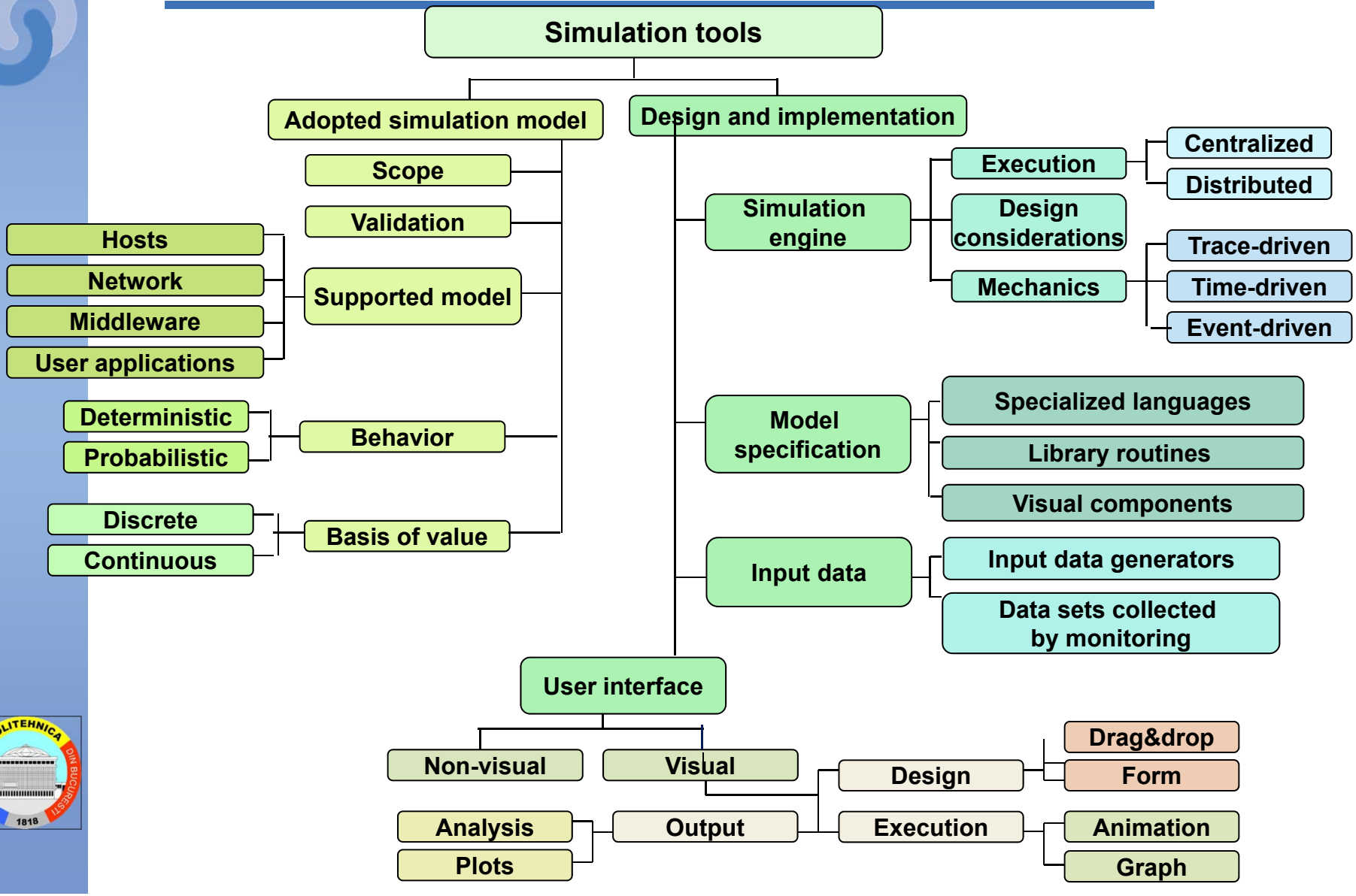
- **Input data**: a simulator can
  - ➔ include input data generators
    - ChicagoSim accepts input data generators
  - ➔ accept data sets collected by monitoring
    - MONARC 2 accepts both types of input (the monitoring data format is the one produced by MonALISA)

A taxonomy of large scale distributed systems simulators (7)

- **User interface**
  - ➔ determines how the user interacts with the simulator
  - ➔ Textual vs. graphical output
  - ➔ Visual design facility, visual execution interface, animations, graphs, visual output analyzers, etc.

- **Validation**
  - ➔ Simulators must present validation results in various forms:
  - ➔ Mathematical comparison vs. comparison between simulation model and real-world testbed systems
    - Only several simulators present validation studies (e.g. Bricks, MONARC and SimGrid)

A taxonomy of large scale distributed systems simulators



Characteristics of the distributed systems

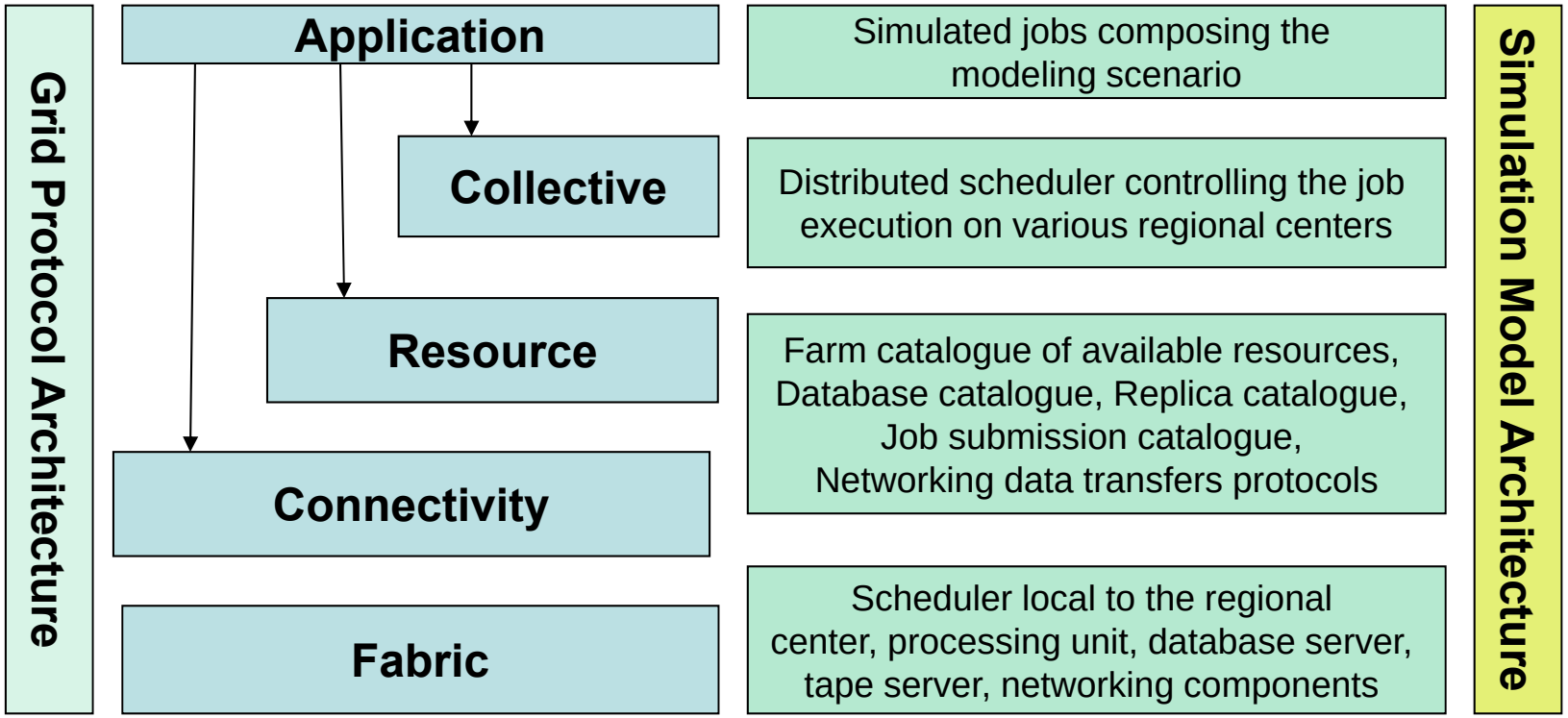
- The characteristics* of a distributed system should be mapped on the simulation model

Characteristic	Influence on the simulation model
Resource sharing	Use of networking components and data sharing entities
Openness	Inclusion of easily extendable object-oriented modeling infrastructure and standard interfaces that allow access to the fabric components inside a running simulation experiment
Concurrency	Inclusion of mechanisms to model concurrent processes and networking transfers (possible based on some interrupt mechanisms)
Scalability	The adoption of an object-oriented simulation model and the use of advanced internal structures to make better use of available physical resources of the underlying stations
Fault-tolerance	Mechanisms to model the occurrence of faults and the possibility to include mechanisms to detect and recover from occurring faults
Transparency	Use of advanced routing algorithms, data replication algorithms, and scheduling algorithms to consider failure-transparency

***Coulouris, G., Dollimore, J. & Kindberg, T. (1994). *Distributed Systems – Concepts and Design*. Addison-Wesley, second edition.**

Grid architecture and its influence on the simulation model

- A simulation model for Grid systems should incorporate the components and characteristics specific to the Grid layered architecture



The influence of the Grid characteristics on a simulator

Grid characteristic*	Influence on the simulation framework
Large scale	Careful design consideration for the simulation model: the use of advanced internal structure could allow the modeling of experiments with many incorporated resources.
Geographical distribution	The inclusion of sites, geographically distributed, in the simulation model. The sites should be connected by special WAN modeled links.
Heterogeneity	Use of various models for hardware components; software architectures captured using probability distributions.
Resource sharing	Represented in the network model.
Multiple administration	Inclusion of a distributed scheduler.
Resource coordination	Resource coordination mechanisms.
Dependable access	Implementation of DAG scheduling algorithms.
Consistent access	Use of standard methods to access the resources.
Pervasive access	The scheduling framework detecting faults and taking appropriate actions.

***Bote-Lorenzo, M., Dimitriadis, Y., & Gomez-Sanchez, E. (2002). *Grid characteristics and uses: a grid definition*. Tehnical Report CICYT, Univ. of Valladolid, Spain.**

A critical analysis of simulation tools for large scale systems

- Evaluation using the proposed taxonomy
  - ➔ Bricks – resource scheduling in Grids
  - ➔ OptorSim – replica management and optimization
  - ➔ SimGrid – scheduling algorithms in heterogeneous, computational distributed environments
  - ➔ GridSim – effective resource allocation techniques based on computational economy
  - ➔ ChicagoSim – designed to investigate scheduling strategies in conjunction with data location
  - ➔ MONARC 2 – generic simulator for large scale distributed systems modeling and simulation

A critical analysis of simulation tools for large scale systems

No.	Simulation tool	Scope	Time base	Simulated components
1	Bricks	Resource scheduling in Grid systems	Discrete	• Client-Server components organized in a central model; • Servers and networking elements modeled as queuing systems; • Scheduling Unit as the central simulation component.
2	OptorSim	Resource scheduling; Data replication strategies	Discrete	• Grid sites composed of Computing Elements and Storage Elements; • Computing Elements run one job at a time; • Complex network model but lack routing, data transport, packetization; • GSs with modeled Resource Broker, Replica Manager and Replica Optimiser.
3	SimGrid	Resource scheduling	Discrete	• Scheduling tasks; • Resource objects: modeled hosts and network links; • Grid model can be obtained from traces (ENV and NWS are supported).
4	GridSim	Resource scheduling; Simplistic data replication	Discrete	• The modeling systems is composed of users, brokers and resources; • Both Computational and Data Grids are supported by the simulation model; • Networking model takes into consideration QoS, background traffic; • Well suited for algorithms designed for Nimrod-G.

A critical analysis of simulation tools for large scale systems

No.	Simulation tool	Scope	Time base	Simulated components
5	EDGSIM	Resource Scheduling	Continuous	• Jobs submitted using appropriate User Interface; • Resource Broker programmed with various Scheduling algorithms; • Replica Catalog mapping logical and physical file names; • Compute Element models the computation Resource of the Grid; • The network model is simple, without considering low-level functionality.
6	ChicagoSim	Resource scheduling; Data replication strategies	Discrete	• Three modeled components: the site, the network and the driver; • Replica management is carried out at local level; • The modeled Grid includes any number of external schedulers, whilst the managing of the files is done locally by a dataset scheduler; • Support for modeling various scheduling algorithms and various replica methods.
7	MONARC	Generic Grid simulator	Discrete	• Processing units, Data Storage, farms, networking, HEP components; • Support for the modeling of scheduling algorithms, replica management, networking procedures, etc. • Strong support for modeling generic Grid architectures.

A critical analysis of simulation tools for large scale systems

No.	Simulation tool	Execution	User interface	Validation
1	Bricks	Centralized, event driven	Textual output, designed to be used with external tools	Performed by replacing the Predictor with NWS
2	OptorSim	Centralized, event and time driven	Graphical user interface, animations	Degenerate tests, fixed values, internal validity
3	SimGrid	Centralized, event driven	Graphical user interface, result analysis capabilities	Fixed values
4	GridSim	Centralized and distributed, event driven	Graphical user interface built on top of the simulator	N/A
5	EDGSIM	Centralized	Graphical user interface, drag-drop capability to construct scenarios	N/A
6	ChicagoSim	Centralized, event driven	Textual output, designed to be used with external tools	N/A
7	MONARC 2	Centralized, event driven	Graphical user interface, animations	Model Validation Monitored resource utilization vs. simulated observations Queuing theory tests

Future trends

- The development of solutions designed for large scale distributed systems is facilitated by the use of adequate simulation instruments
  - ➔ Existing simulators are too focused on specific technologies
- The lack of generality in simulation model will be increasingly reduced
  - ➔ MONARC 2
  - ➔ ChicagoSim
- Lack of evaluation results
  - ➔ A simulator must present comparisons between experiments modeling small distributed systems against equivalent real-world testbeds
  - ➔ Use of queuing theory
- The need to model very large distributed systems, with a great number of resources
  - ➔ Optimizations of the simulation engine
    - Advanced priority queuing structures for the simulation events
    - Optimizing the way in which simulated entities are being scheduled in simulation for execution
    - Using various simplifications mechanisms
    - Using the underlying physical distributed resources of clusters of nodes

Conclusions

- A taxonomy for comparing simulators for large scale distributed systems
  - ➔ Existing work is either too generic or do not consider important characteristics.
  - ➔ The taxonomy is particularly focused on the simulation of distributed systems.
- Comparison study of the most important simulation projects involved in the modeling of distributed systems
  - ➔ How they relate and differ
  - ➔ Their advantages and disadvantages
- They all cover important aspects of distributed systems
  - ➔ Exploration of different areas of parameter space

Questions?

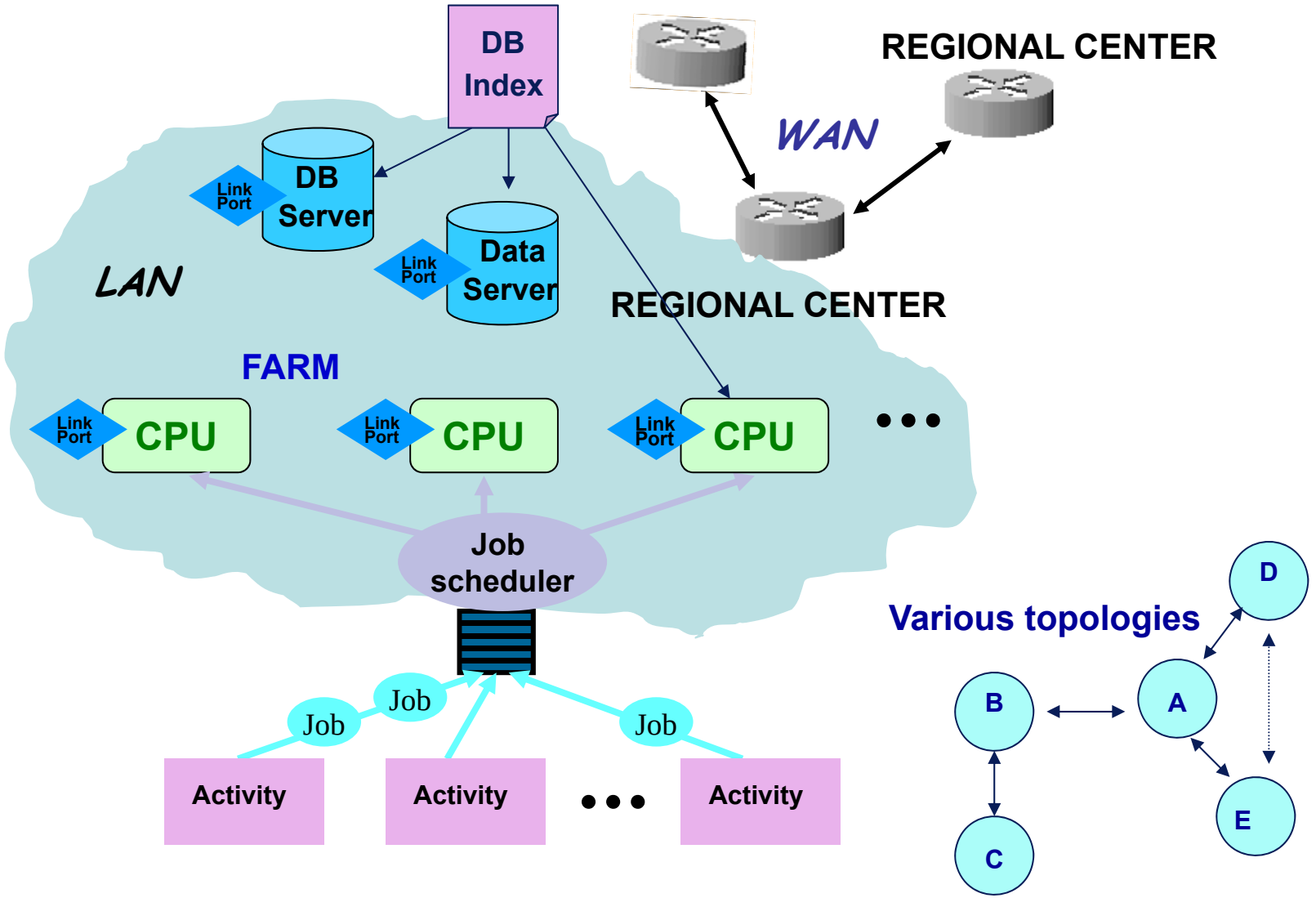
Thank You!

ciprian.dobre@cs.pub.ro

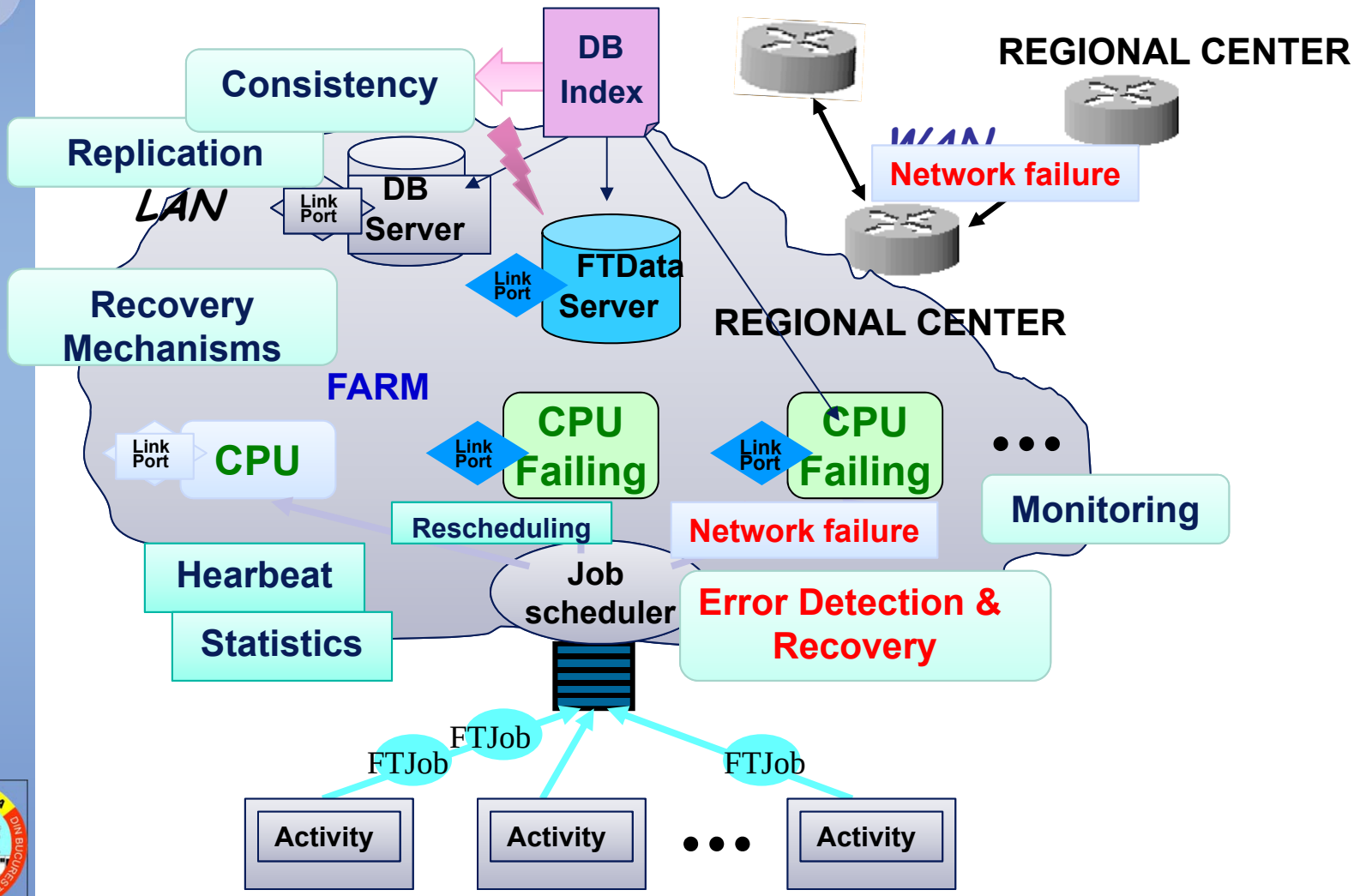
http://monarc.cacr.caltech.edu

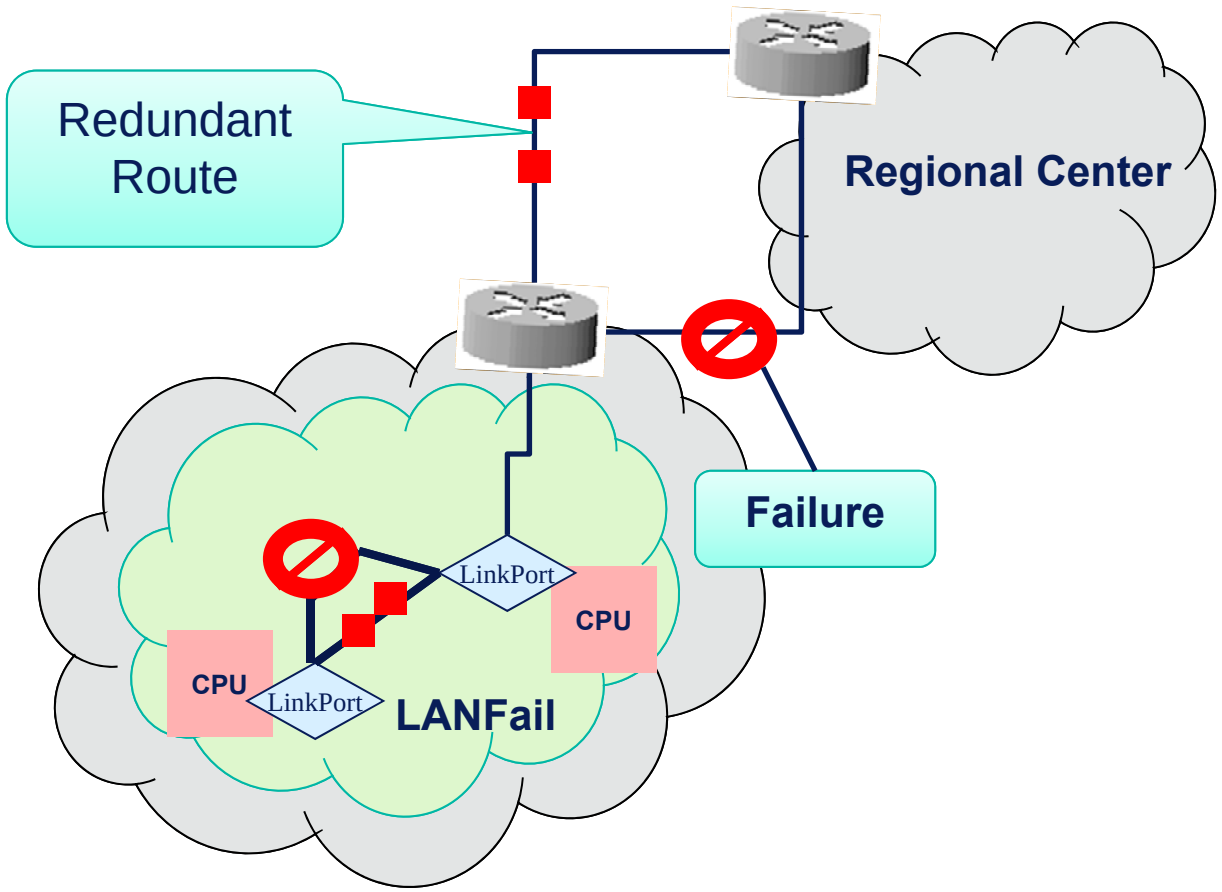
Supplementary slides

Components of MONARC 2



Components of MONARC 2



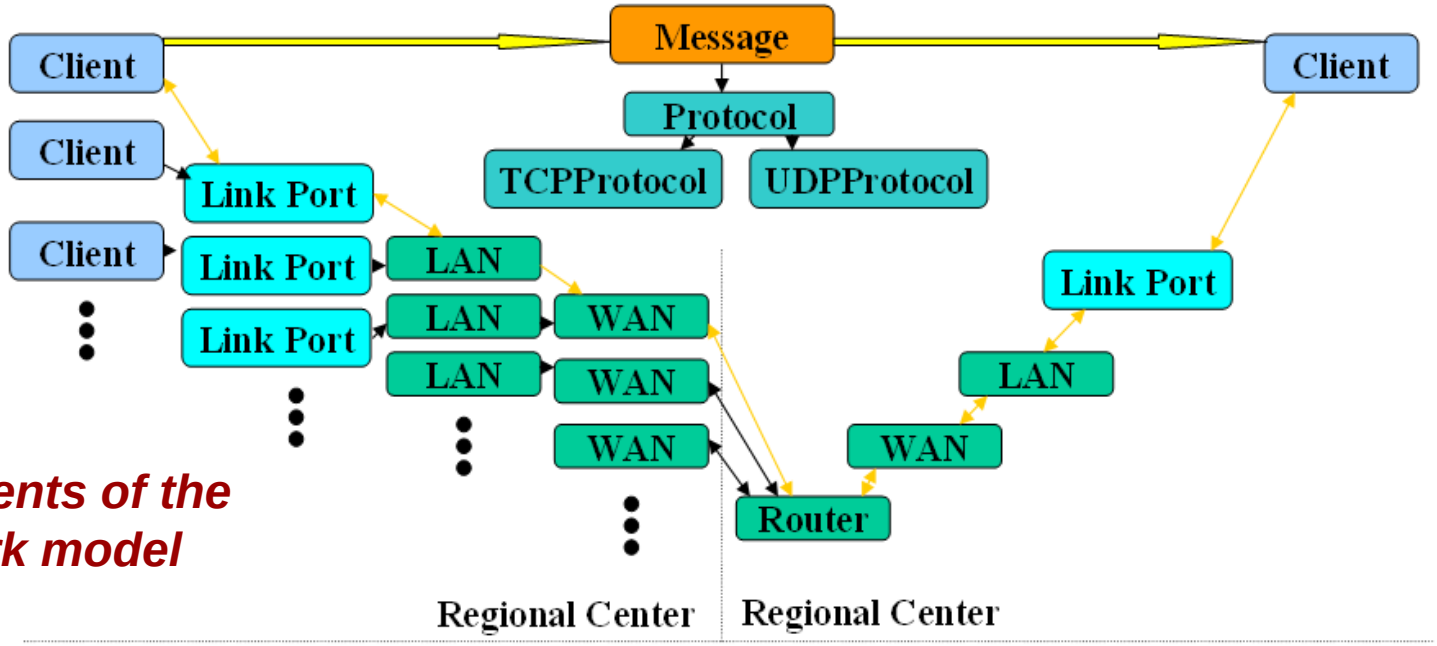

Redundant Route
Regional Center
Failure
LinkPort
CPU
CPU
LinkPort
LANFail

Network model

- **data traffic simulated for both local and wide area networks**
- **a simulation at the packet level is practically impossible**
- **we adopted a larger scale aproach, based on an "interrupt" mechanism**

Network Entity:

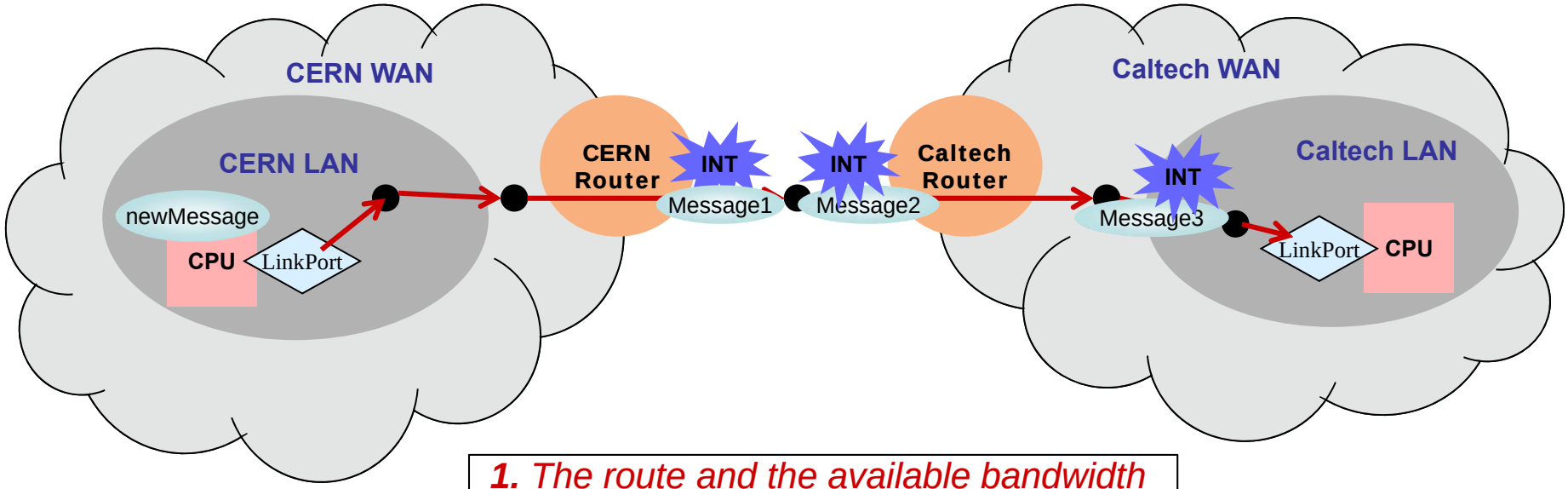
- LAN, WAN, LinkPort
- main attribute: bandwidth
- keeps the evidence of the messages that traverse it



Components of the network model

Simulating the network transfers

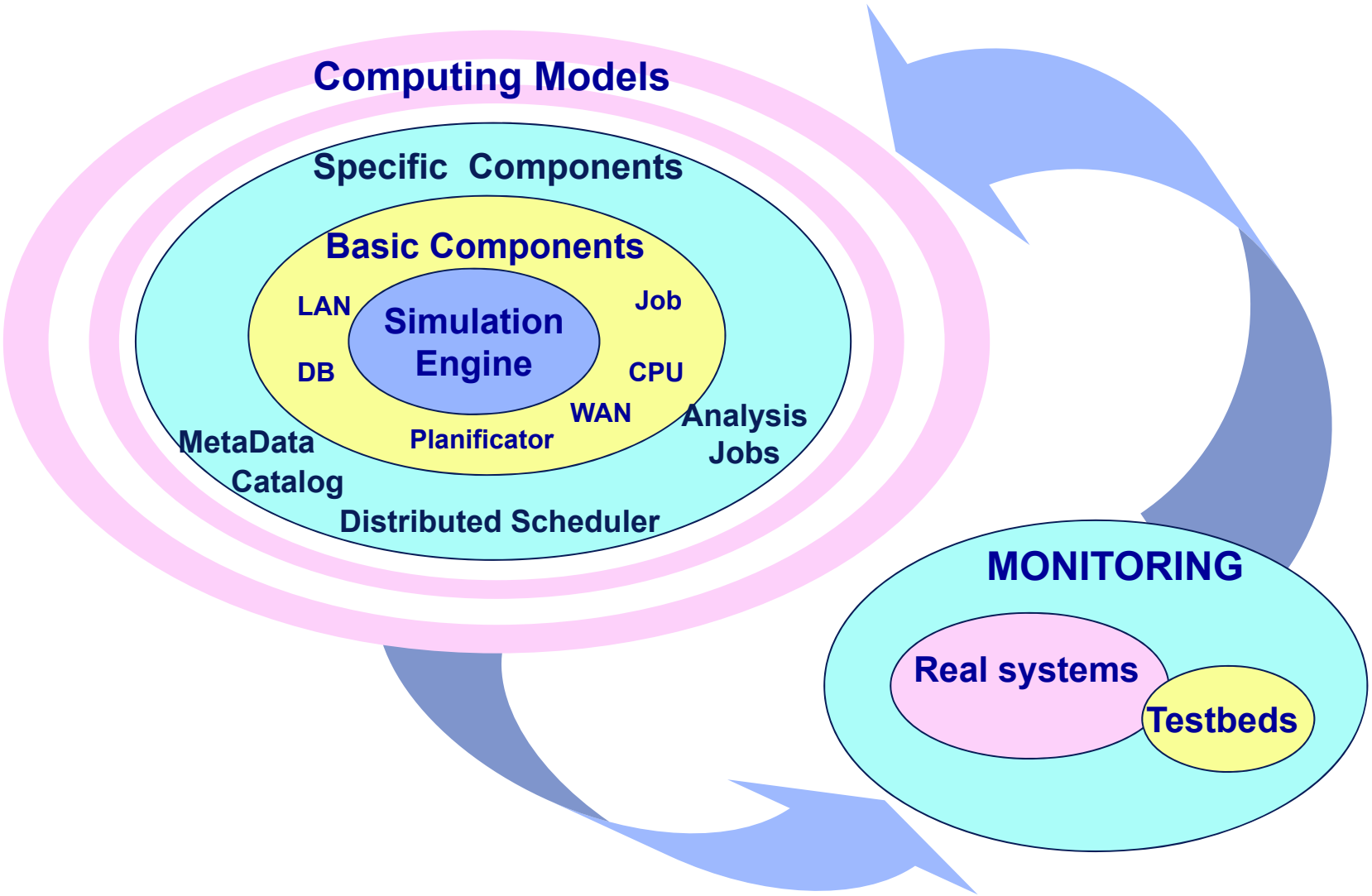
- interrupt mechanism similar with the one used for job execution simulation
- the initial speed of a message is determined by evaluating the bandwidth that each entity on the route can offer
- different network protocols can be modelled



1. *The route and the available bandwidth for the new message are determined.*

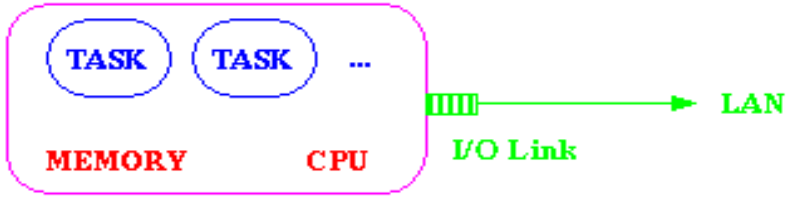
1. *The messages on the route are interrupted and their speeds are recalculated.*

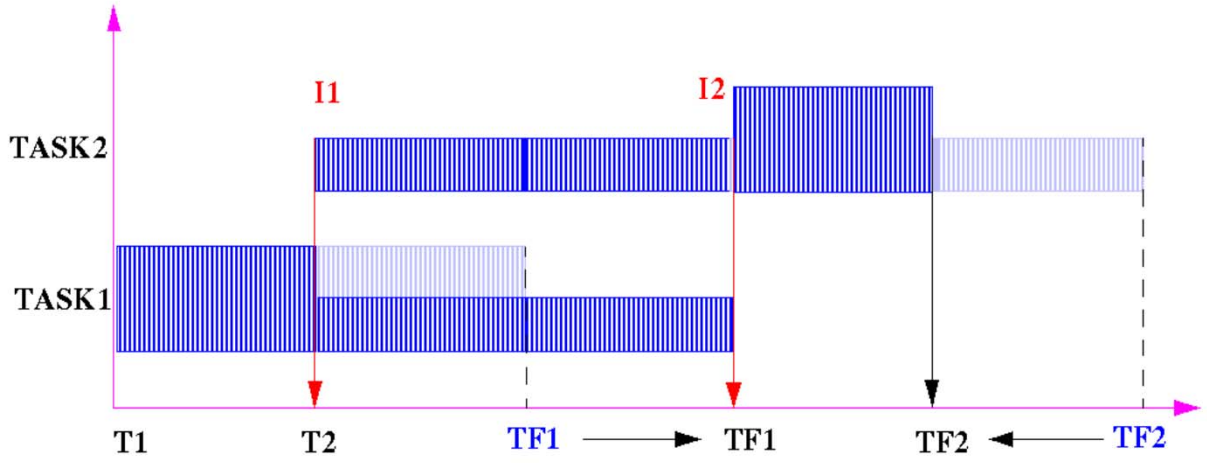
The architecture of MONARC 2



Multitasking Processing Model

- The interrupt mechanism is useful for describing concurrent running tasks sharing resources, memories, I/O operations
- It is based on the simulation events





Running performance