COSSIM

An Integrated Solution to Address the Simulator Gap for Parallel Heterogeneous Systems

Andreas Brokalakis
Synelixis Solutions Ltd, Greece
brokalakis@synelixis.com

Nikolaos Tampouratzis
Telecommunication Systems Institute, Greece
ntampouratzis@mhl.tuc.gr

gem5 Workshop
ARM Research Summit 2017
September 11, 2017

Motivation

- Microprocessor and Systems designers cannot ignore **network interactions** when designing new system architectures, since networks are **an integral part** of all parallel systems
  - Too many **interactions**, too **complicated**
  - Major **optimization** opportunities

- Majority of the **existing simulation tools** can **handle efficiently only parts** of a system
  - either **only the processing part** or **the network**

- **Existing simulators** are typically single-threaded
  - A parallel approach is required for networked systems simulation if realistic simulation times have to expected

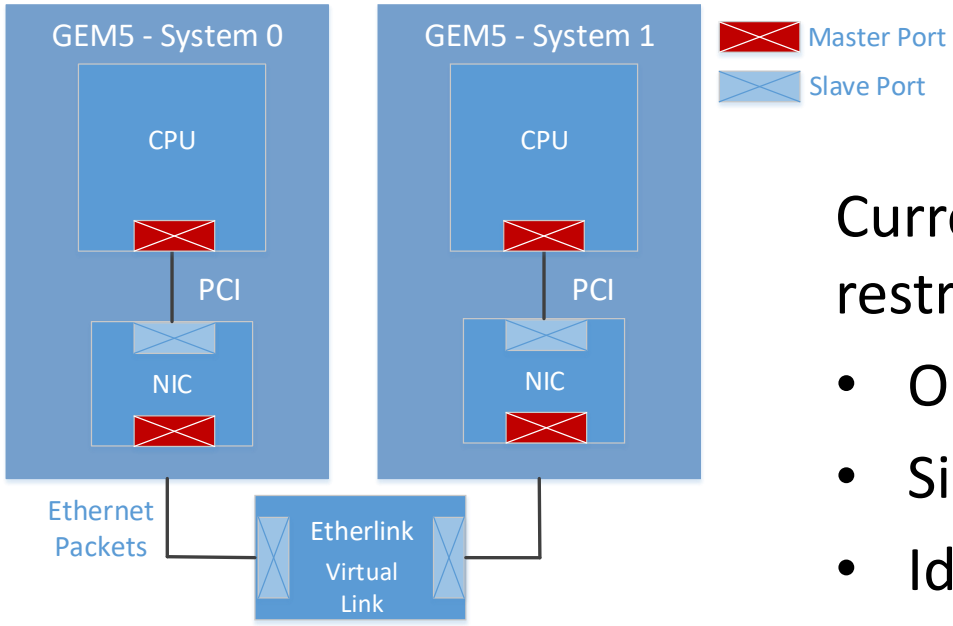
What is COSSIM?

- **COSSIM** is an **open-source** framework that can simulate **Highly Parallel Systems** or more generally **Systems of Systems**
  - Networking and the processing simulators are integrated into **a single framework**
    - **Single notion of time**, **accurate** processing and network **interactions**
  - Easy simulation set-up, execution and visualization of results through a **Graphical User Interface (Eclipse-based)**
  - **IP-based** so that simulation can be **distributed**
  - **Power estimation tools** are also integrated in order to account for the real processing - network interactions
  - **Fully functional** version **open & free**, **commercial support** and support for **add-on packages** by providing proper extensions

COSSIM Framework

- **Key concept :**
  - use **well-established processing and network simulators**
  - Retain **compatibility** to be able to readily take advantage of all related research and development work
- **COSSIM** is built on top of
  - **GEM5**, to simulate the components of each processing node in the simulated system
  - **OMNET++**, to simulate the real networking infrastructure
  - **McPAT / OMNET++ addons** to provide energy and power consumption estimations
  - **CERTI (IEEE1516 HLA)** to integrate all simulator packages together

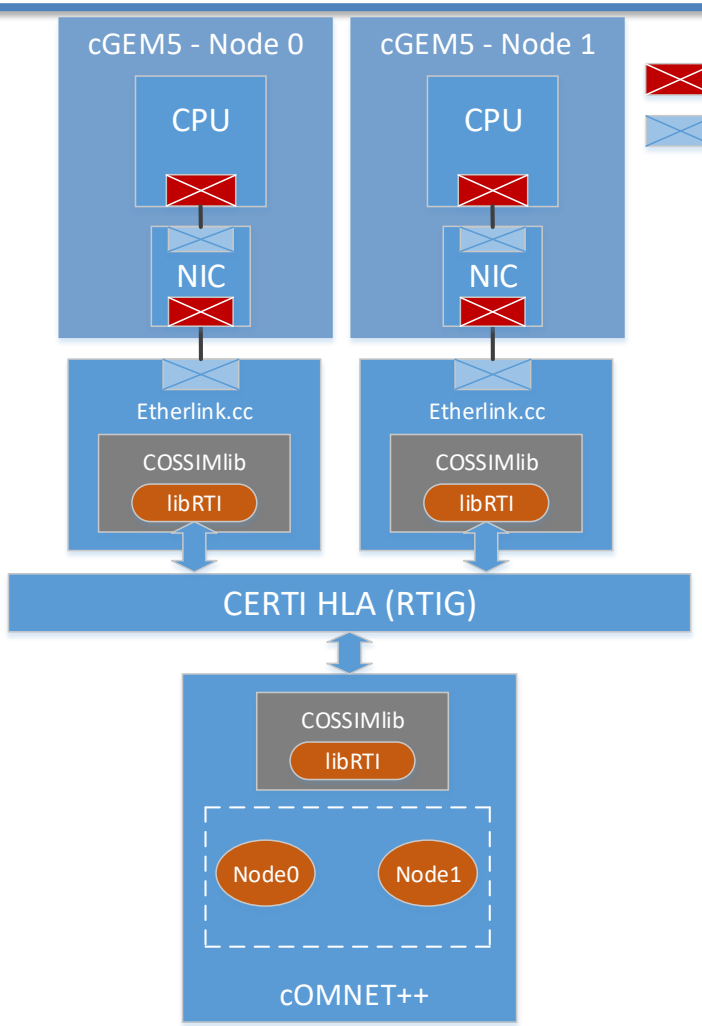
Network Support on gem5



Current network model of gem5 is restrictive:

- Only two nodes
- Simple wire (no switch / router)
- Identical CPU systems
- Single-thread / simplistic synchronization

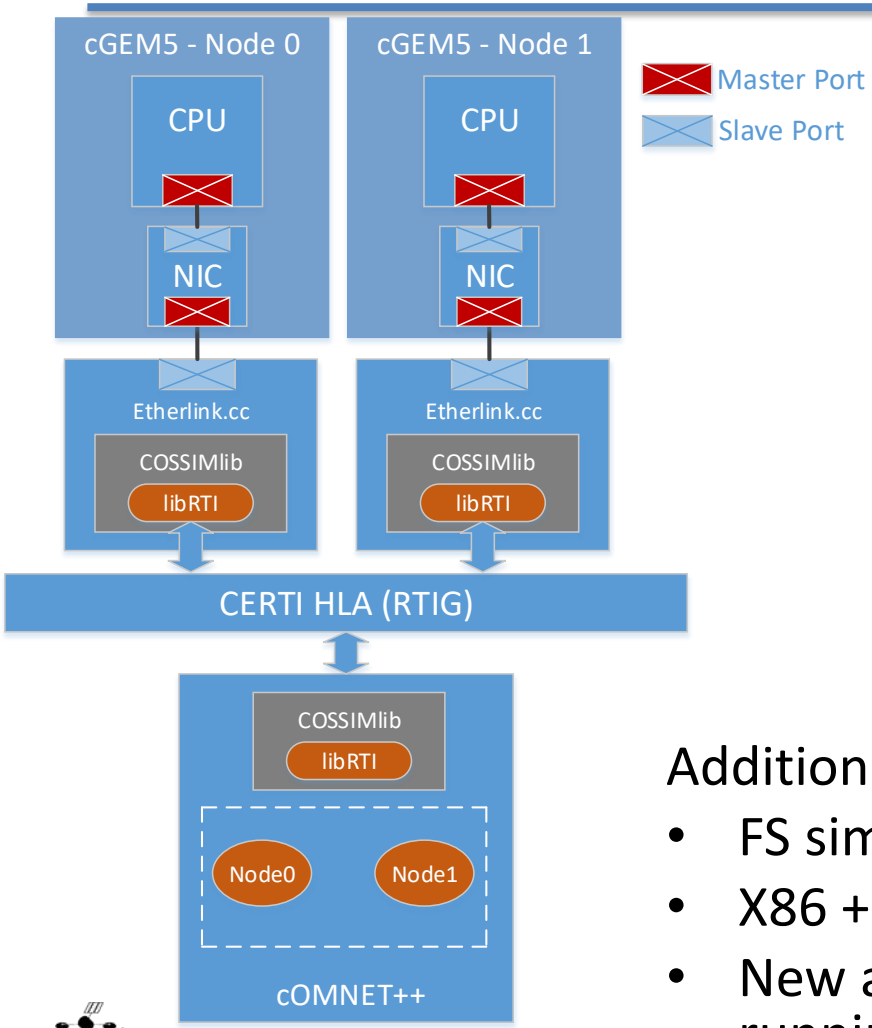
Extending gem5 for COSSIM (1)



In the modified version of gem5 (cGEM5):

- Ethernet packets are captured and through a custom library are sent to an HLA server
- The packets are sent to a representation of the nodes in the network simulator implementing the communication topology

Extending gem5 for COSSIM (2)



By this process:

- An *arbitrary number of nodes* can be connected
- *Sophisticated network topologies* can be implemented
- Each node is independent and therefore *heterogeneous systems* can be composed
- IP-based (HLA) interconnection allows for *parallel / distributed execution* of each cGEM5 instance

Additionally:

- FS simulation only (NIC + OS drivers rq.)
- X86 + ARM fully supported
- New and most current linux kernel versions running

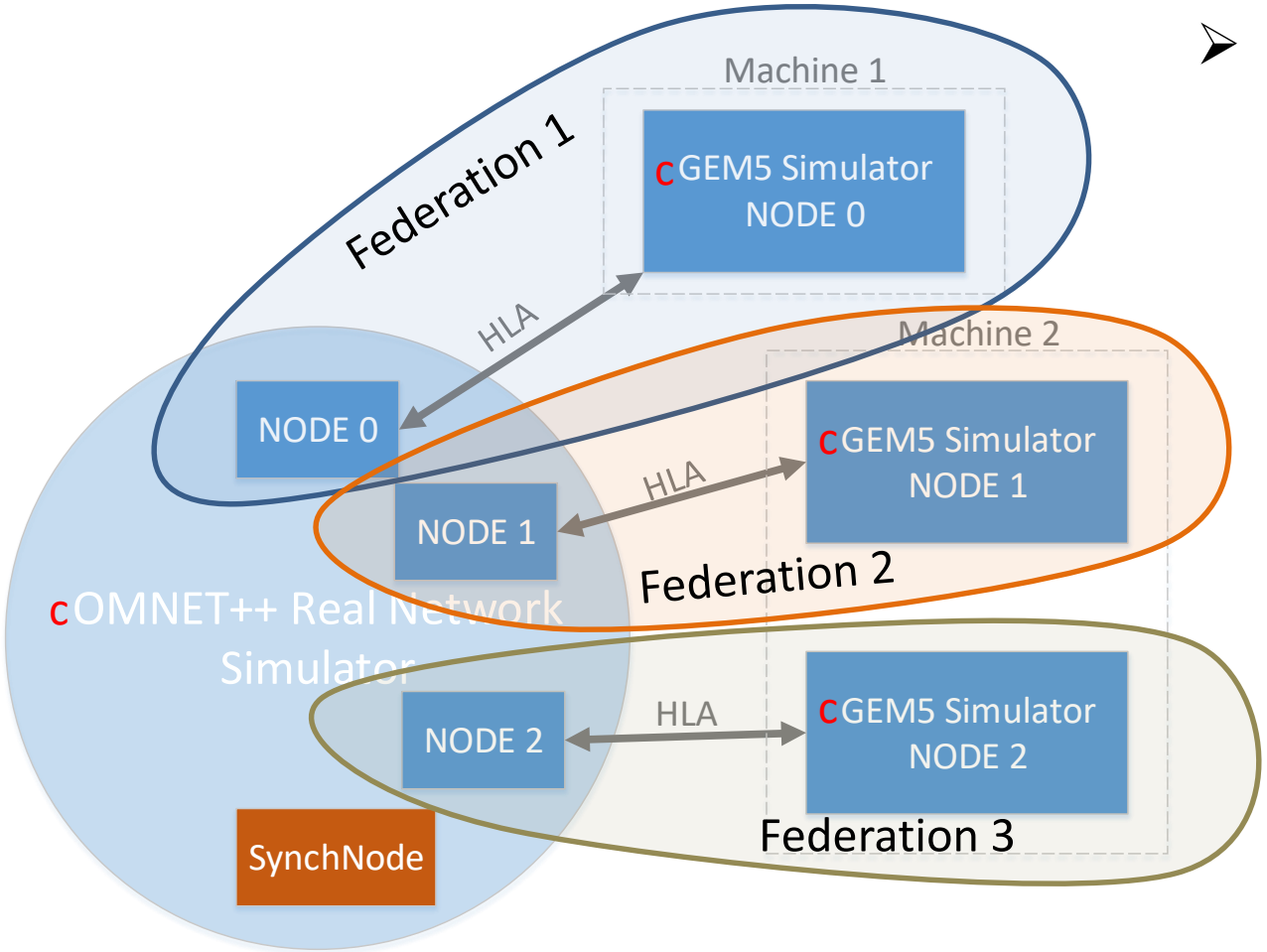
The Network Simulator Sub System

- We employ OMNeT++ network simulator
- Each cGEM5 system is reflected in an OMNET++ HLA-enabled node
- **Challenge**: Incompatible network stacks between OMNeT++ and cGEM5
- **Developed**: custom-fit functionality developed at user space
  - No modification to the OMNeT++ -INET code
  - Fully compatible with the OMNeT++ legacy
- Preserving OMNET++ functionality, any supported network topology can be used, including network devices and custom nodes (in the OMNET++ space – not simulated through cGEM5)

The Elephant in the Room
Integration and Synchronization (1)

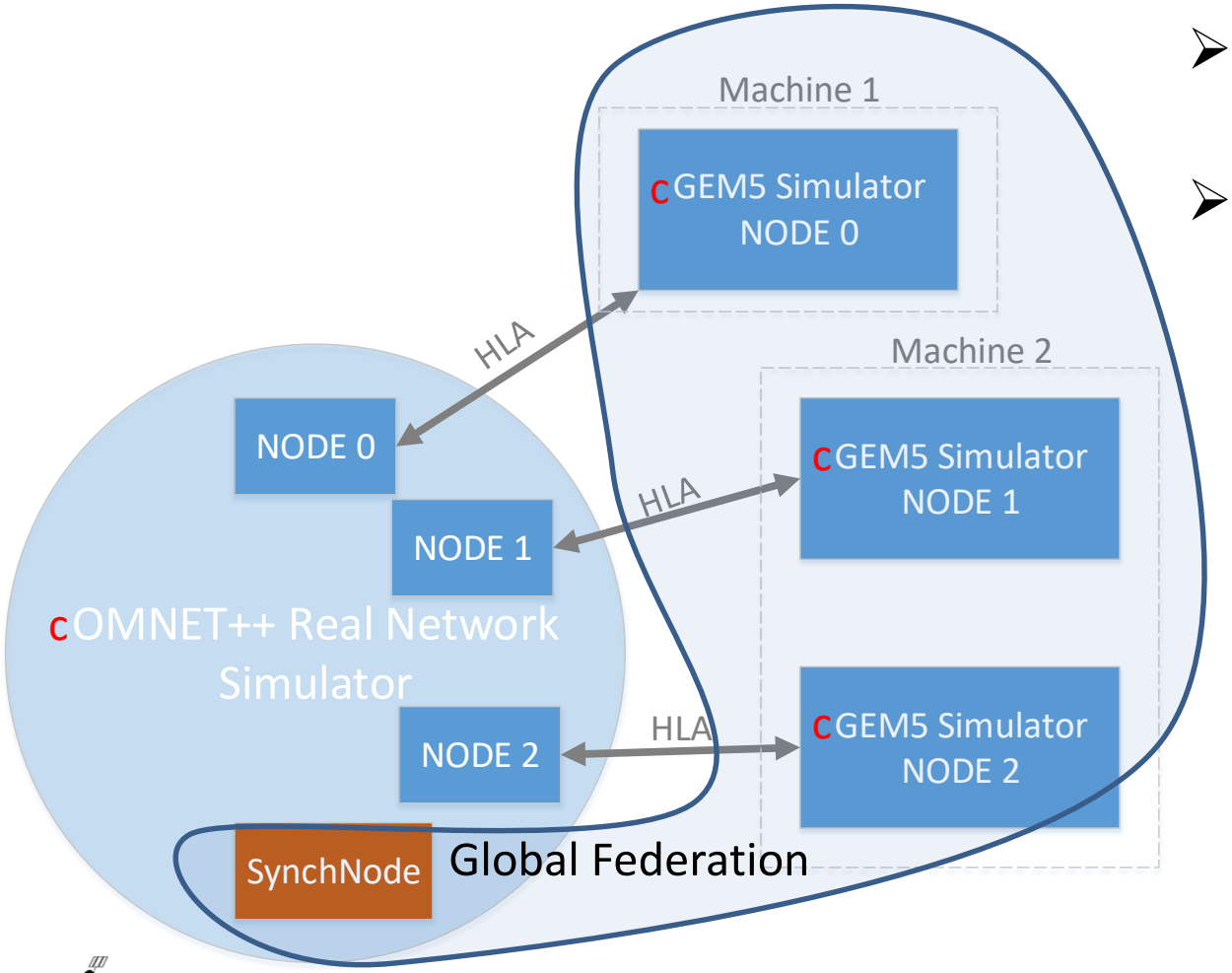
- Synchronization issues arise
  - differences between gem5 and OMNET++ (event-based)
  - communication between different gem5 systems running independently
  - requirement of a common notion of time throughout the whole simulated system
- Two-stage solution through CERTI HLA
  - Synchronization per node (each cGEM5 node needs to synchronize with its counterpart network node)
  - Global Synchronization (sync all nodes simultaneously periodically as different types of CPUs with different clock cycles → Different simulation time)

The Elephant in the Room
Integration and Synchronization (2)



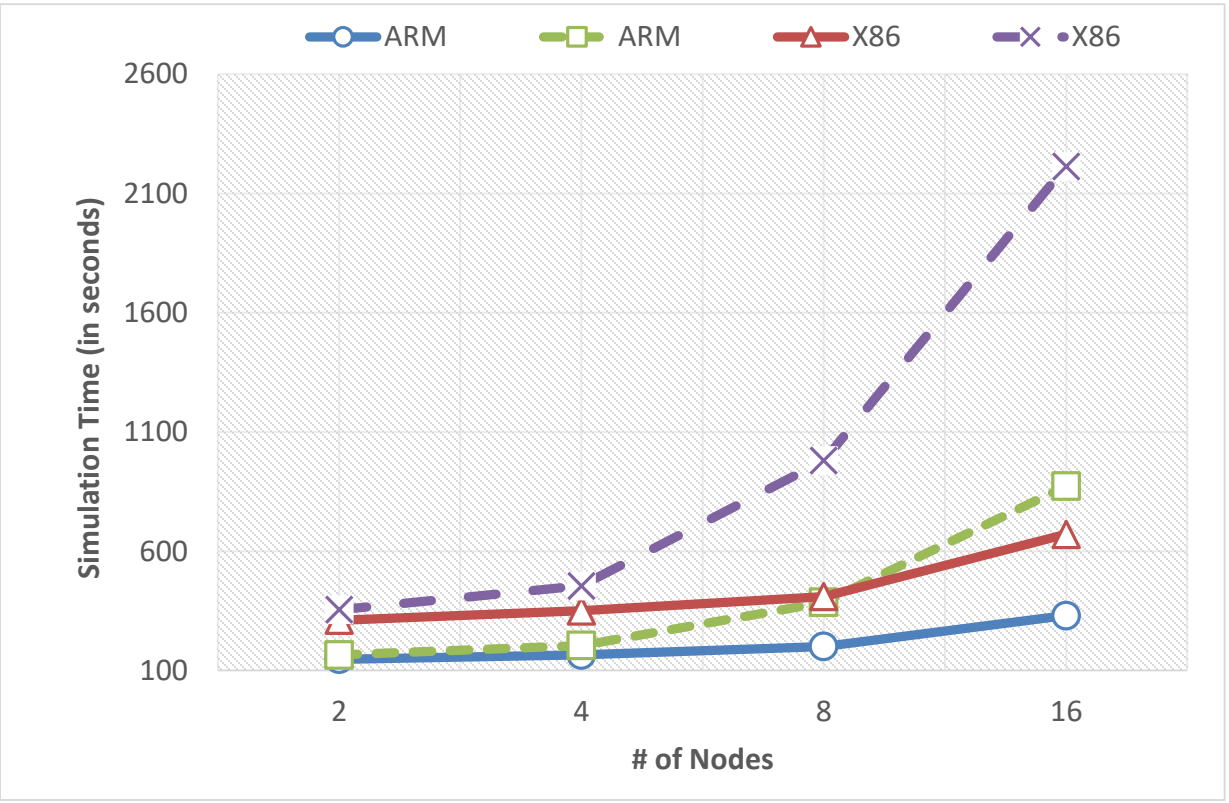
- *Synchronization per node*
  - *Exchange Data Packets*

The Elephant in the Room
Integration and Synchronization (2)



- *Synchronization per node*
  - *Exchange Data Packets*
- *Global Synchronization*
  - *All cGEM5 nodes*
  - *cOMNET++ SynchNode*

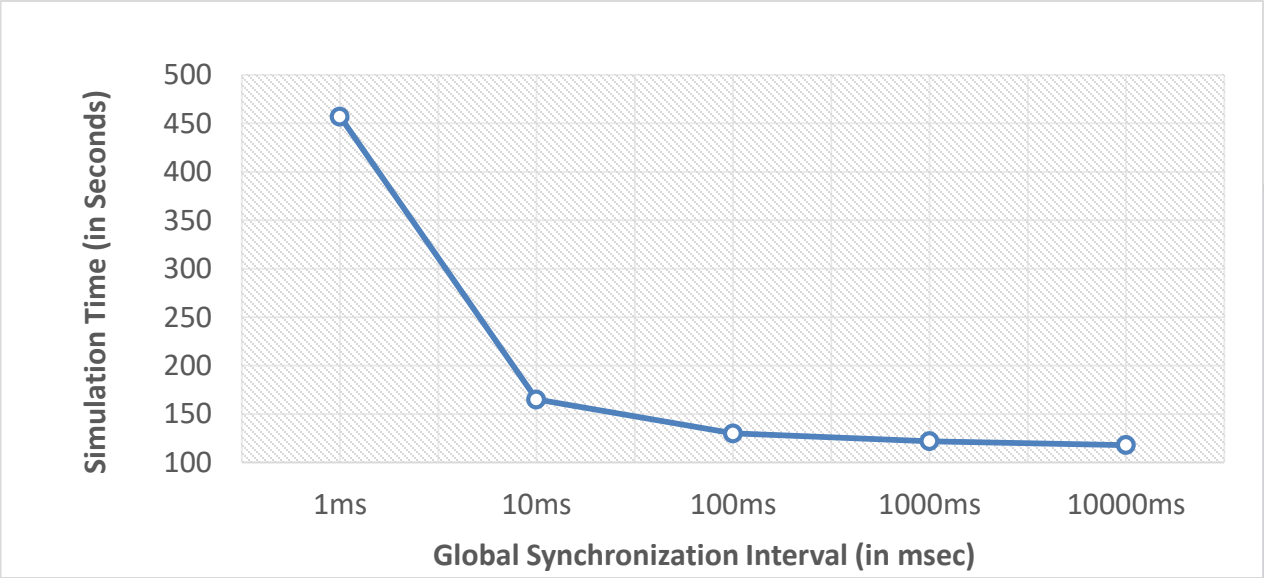
Performance



- ARM / x86 systems while booting Linux OS
- One cGEM5 instance per physical core
- Machine 1 (Quad-core)
- Machine 2 (12-core connected to Machine 1 through LAN)

Performance vs Accuracy

- Global Synchronization Interval introduces a potential performance bottleneck
  - Restricts how long a cGEM5 instance can run freely before pausing for synchronization
  - Affects accuracy of results (application dependent – user settable)
- Example: 4-node ARM systems while booting Linux OS



Thank you!