Hemera

B.1 Working Group

Transparent, safe and efficient large scale computing

Stéphane Genaud (ICPS) - Fabrice Huet (OASIS)

oct. 5, 2010

Scope

How can applications take advantage of large scale, non-uniform, sets of computing resources.

- Scalability: preserve speedups on large platforms
- Weak scalability: compute a larger dataset in a similar amount of time
- Weaker scalability: achieve to process datasets too large/long to fit previously.

- Hierarchical architecture: e.g probably client $\leftrightarrow$ cluster(s), (e.g clouds)
- High-latency: WAN network links, (possibly) unbalanced processors.

- Programming Models: combination of models ? new paradigms ?
- Middleware: which abstractions for runtime librairies or users ?

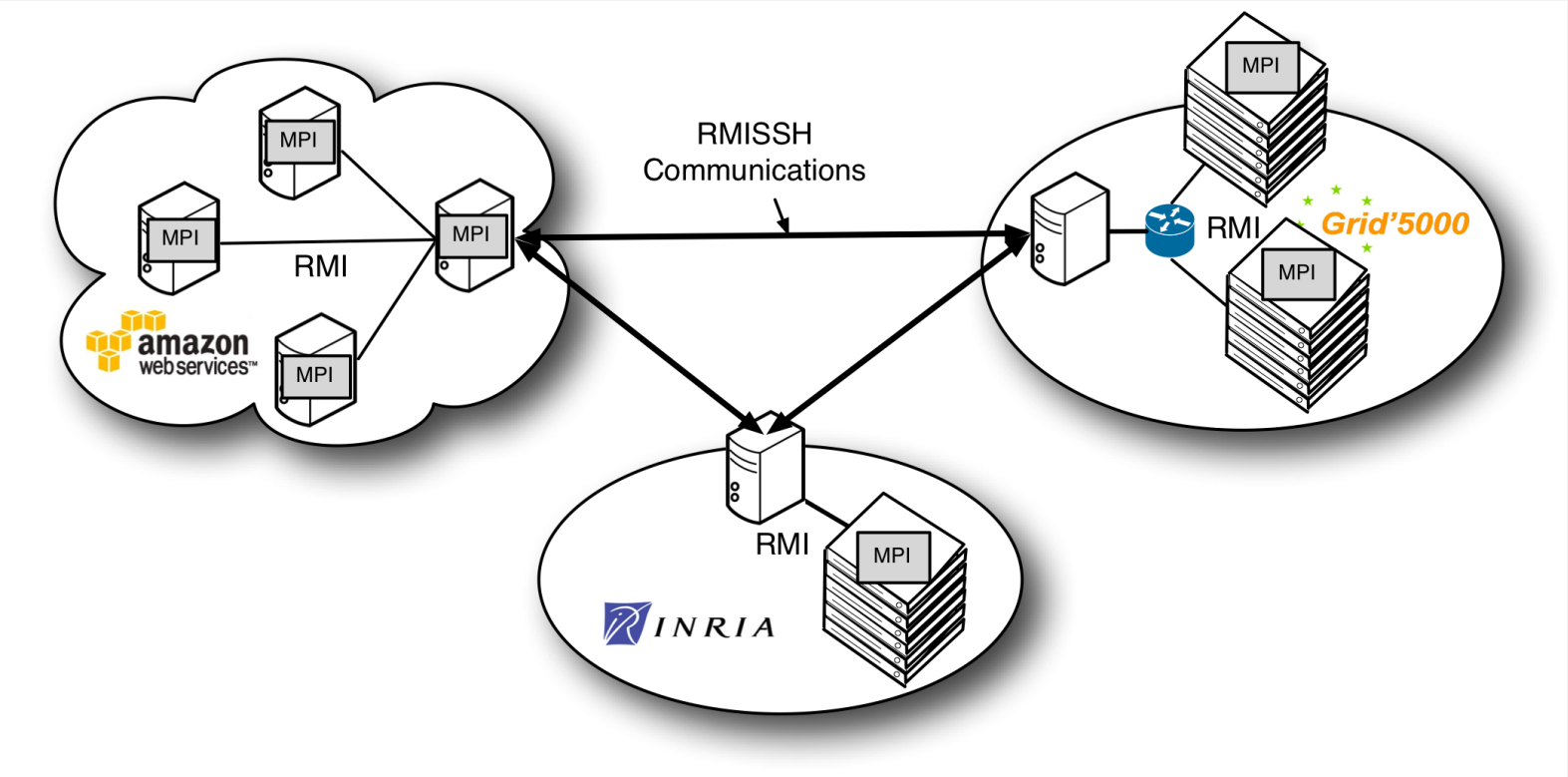
Example: ProActive – programming model

Objective 1: to improve the Active Object model for multicores

- An Active Objects (AO) forms a complete subsystem. It has
  - A queue of pending requests
  - A single thread of service
  - A set of passive (standard) objects
- Issue: an AO cannot share its private data whereas multicores benefit from shared memory.
- Options (ongoing work): data can :
  - be copied between AOs (consistency issues)
  - embedded into an AO (performance and other issues)

Example: ProActive – middleware

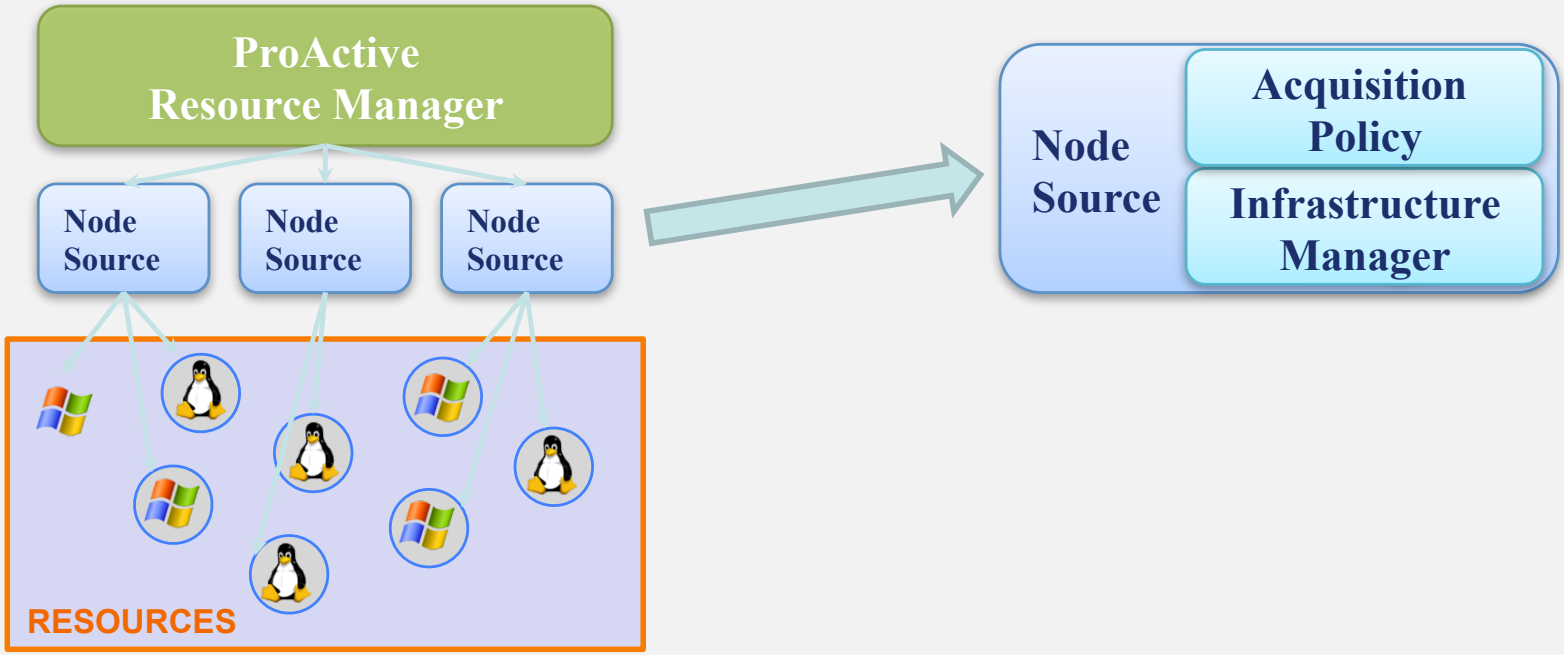
Objective 2: enable independent applications coupling



- MPI applications:
  - Handle hierarchical MPI process ranks
  - ProActive tunneling and message forwarding for multi-domain execution
- Legacy codes coupling

Example: ProActive – middleware

Objective 3: federate resources



- Define a **node source**: a pluggable entity
  - tell *how* and *when* resources may be acquired
  - can be used to acquire resourcesi from clouds (tested with Amazon EC2)

Workshop

- Forum : runtime/middleware designers $\Leftrightarrow$ application designers
- Benchmarks on real use case of
  - alternative programming models: *JavaSpace*, *Active Objects*, *MapReduce*, *HOCL*, *BIOOM*, ...
  - complex deployments: workflows, code coupling, webservices
- Related issues:
  - Fault-tolerance
  - Load-balancing / Scheduling