XtreemOS

Enabling Linux for the Grid

Grid Checkpointing Service

Heinrich-Heine University Duesseldorf

XtreemOS IP project
is funded by the European Commission under contract IST-FP6-033576

Information Society Technologies

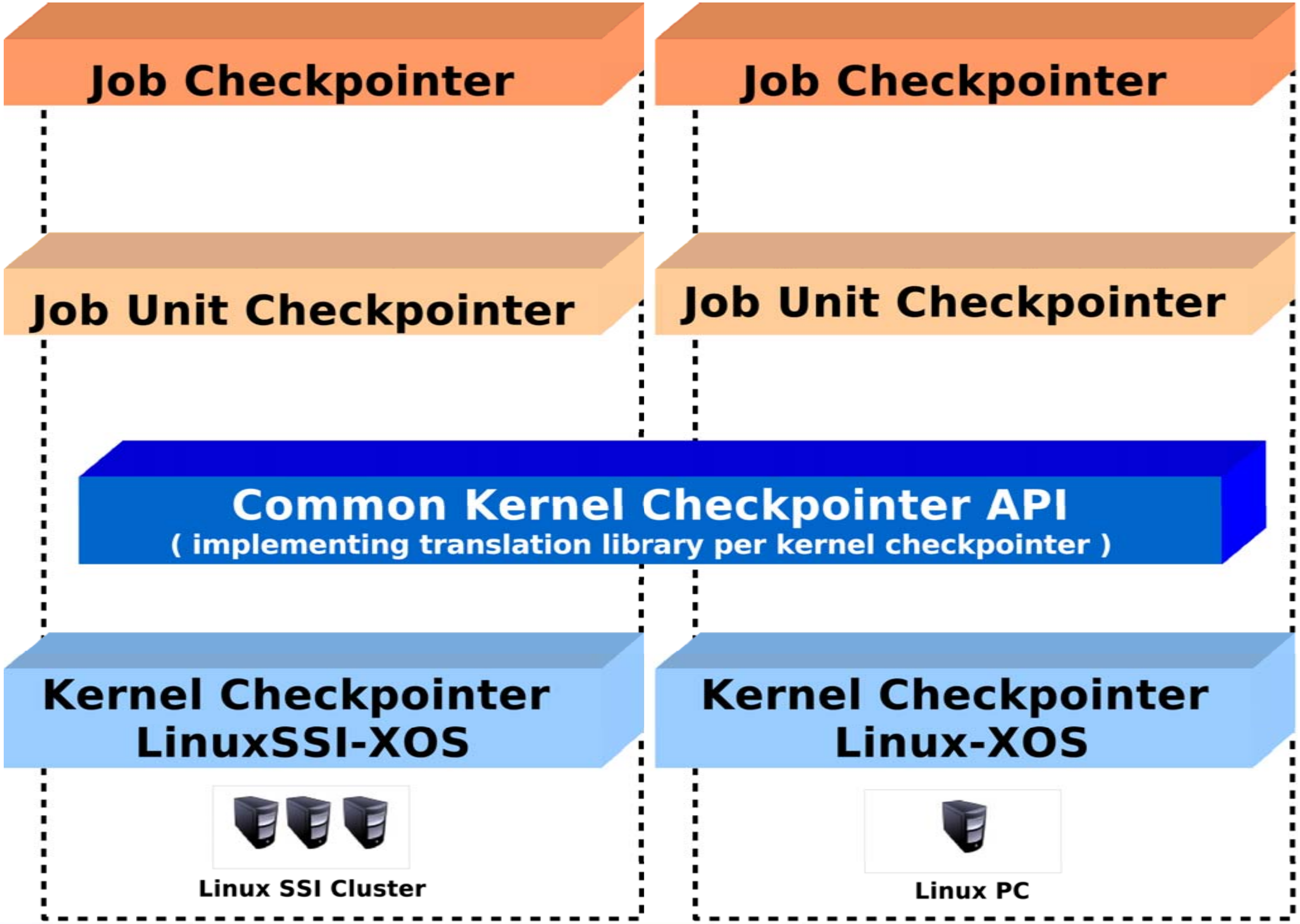
What is XtreemGCP?

- **A grid service integrated within AEM (WP3.3) providing job migration and job fault tolerance for grids.**
- **Service aims at integrating existing kernel checkpointers.**
- **By defining a common kernel checkpointer API implemented by translation libraries.**

Terminology

- **Job** = collection of job units.
- **Job unit** = collection of processes running on a grid node
- **Process** = managed by the LinuxXOS and LinuxSSI kernels of a grid node
- **Grid node** = PC or LinuxSSI cluster

XtreemGCP Architecture



Job Checkpointer

Job Checkpointer

Job Unit Checkpointer

Job Unit Checkpointer

Common Kernel Checkpointer API

(implementing translation library per kernel checkpointer)

Kernel Checkpointer LinuxSSI-XOS

Kernel Checkpointer Linux-XOS

XtreemGCP Architecture (2)

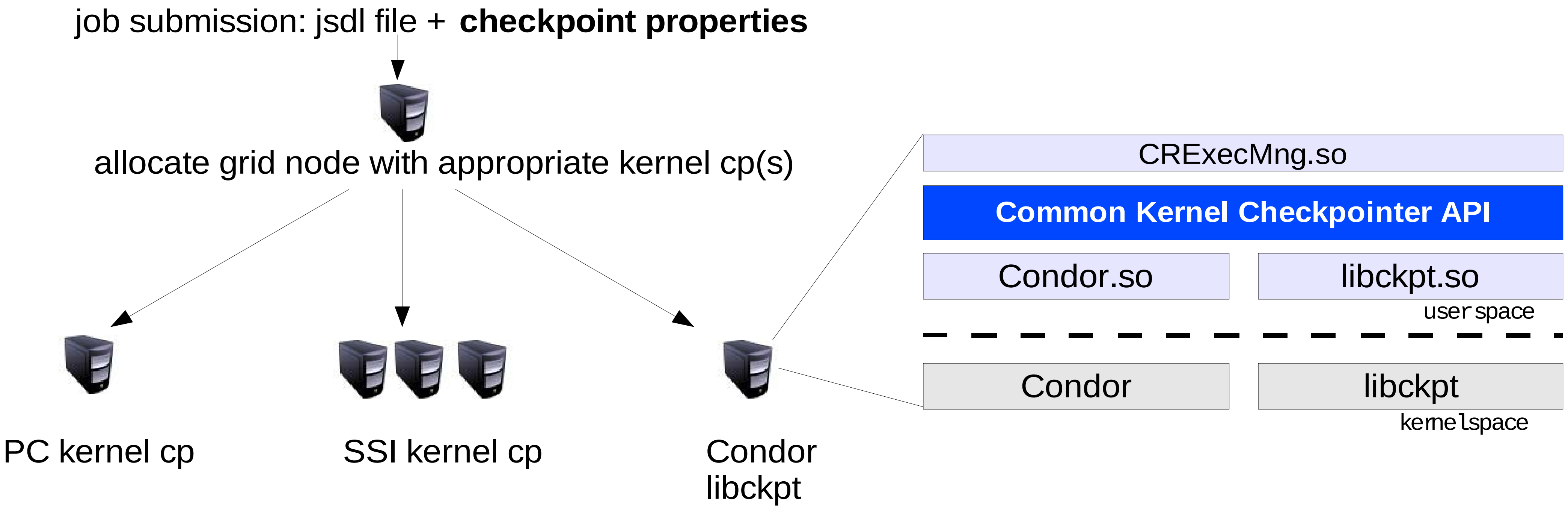
- **Checkpoint files stored in XtreemFS**

- **Checkpointing strategies controlled by job checkpointer**
  - Coordinated checkpointing:
    - for job migration & fault tolerance
  - Uncoordinated checkpointing:
    - Avoiding coordination overhead
    - For fault tolerance, only

- **Adaptive checkpointing:**
  - Based on monitoring parameters (e.g. failure frequency)
  - Used to adapt checkpointing parameters and/or strategies

Common Kernel Checkpointer API

- For accessing different kernel checkpointers in an uniform way.
- Implemented by a translation libraries.



job submission: jsdl file + **checkpoint properties**

allocate grid node with appropriate kernel cp(s)

PC kernel cp

SSI kernel cp

Condor libckpt

CRExecMng.so

Common Kernel Checkpointer API

Condor.so | libckpt.so

user space

Condor | libckpt

kernel space

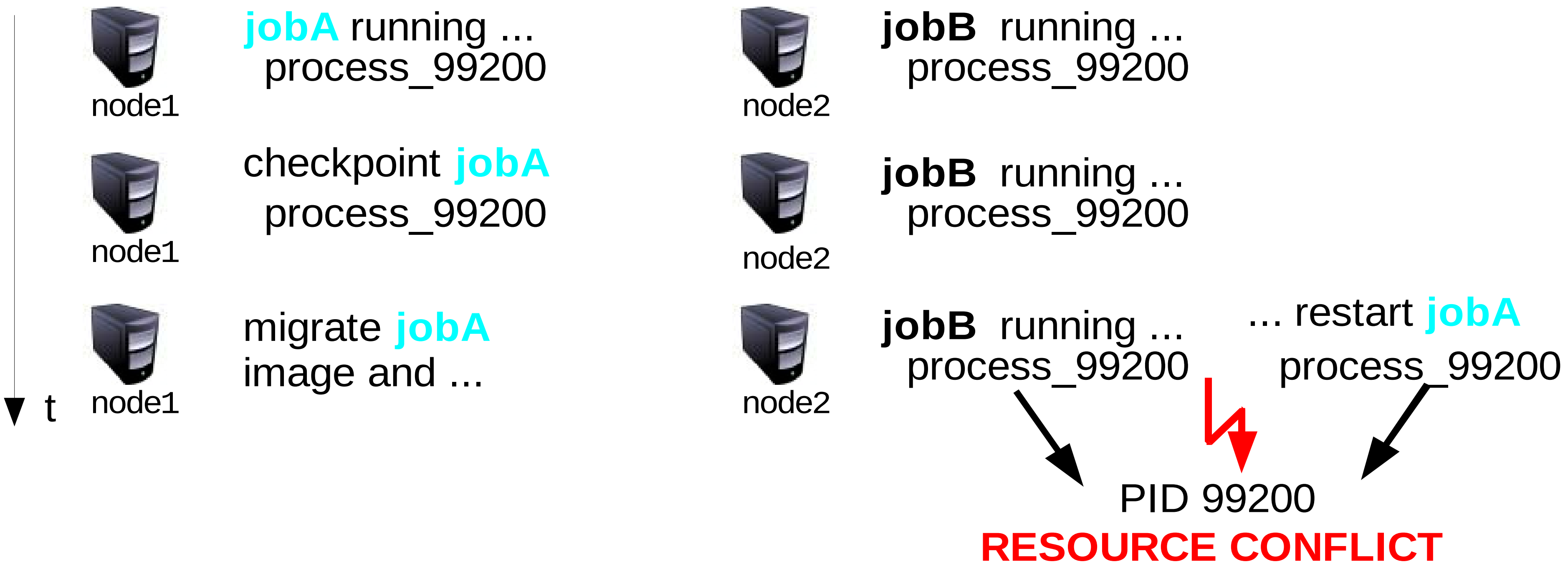
Common Kernel Checkpointer API

- **Book-keeping of process dependencies.**
- **Handle different process grouping techniques.**
- **Callbacks:**
  - Processed during checkpoint and restart operation
  - Allows applications to optimize checkpointing
  - Used to drain communication channels

Resource Conflict during Restart

- **Jobs share grid nodes → resource conflicts may occur**

No Resource Isolation:



jobA running ...
process_99200

node1

jobB running ...
process_99200

node2

checkpoint jobA
process_99200

node1

jobB running ...
process_99200

node2

migrate jobA
image and ...

node1

t

jobB running ...
process_99200

node2

... restart jobA
process_99200

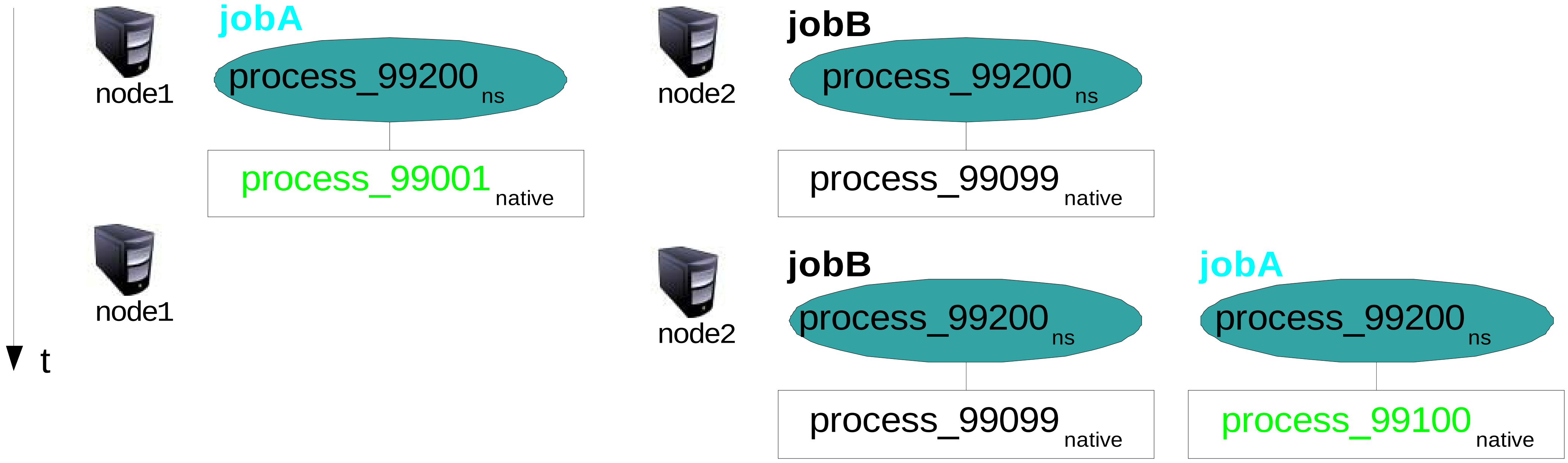
PID 99200

RESOURCE CONFLICT

Restart without Conflicts

- **Cgroups tied to hierachical name space:**

Resource isolation:



node1

jobA

process_99200$_{ns}$

process_99001$_{native}$

node2

jobB

process_99200$_{ns}$

process_99099$_{native}$

node1

node2

jobB

process_99200$_{ns}$

process_99099$_{native}$

jobA

process_99200$_{ns}$

process_99100$_{native}$

t

NO RESOURCE CONFLICT ✓

Conclusion

- **XtreemGCP is an open service architecture integrating existing kernel checkpointing solutions.**

- **Used for job migration and fault tolerance.**

- **Status:**
  - Translibs for PCs using BLCR and LinuxSSI clusters available
  - Basic Checkpoint and restart prototype working

- **Future work:**
  - Integration of cgroups
  - Communication channel draining
  - Integration of OpenVZ and Linux-native checkpointer