Single System Image OS for Clusters: Kerrighed Approach

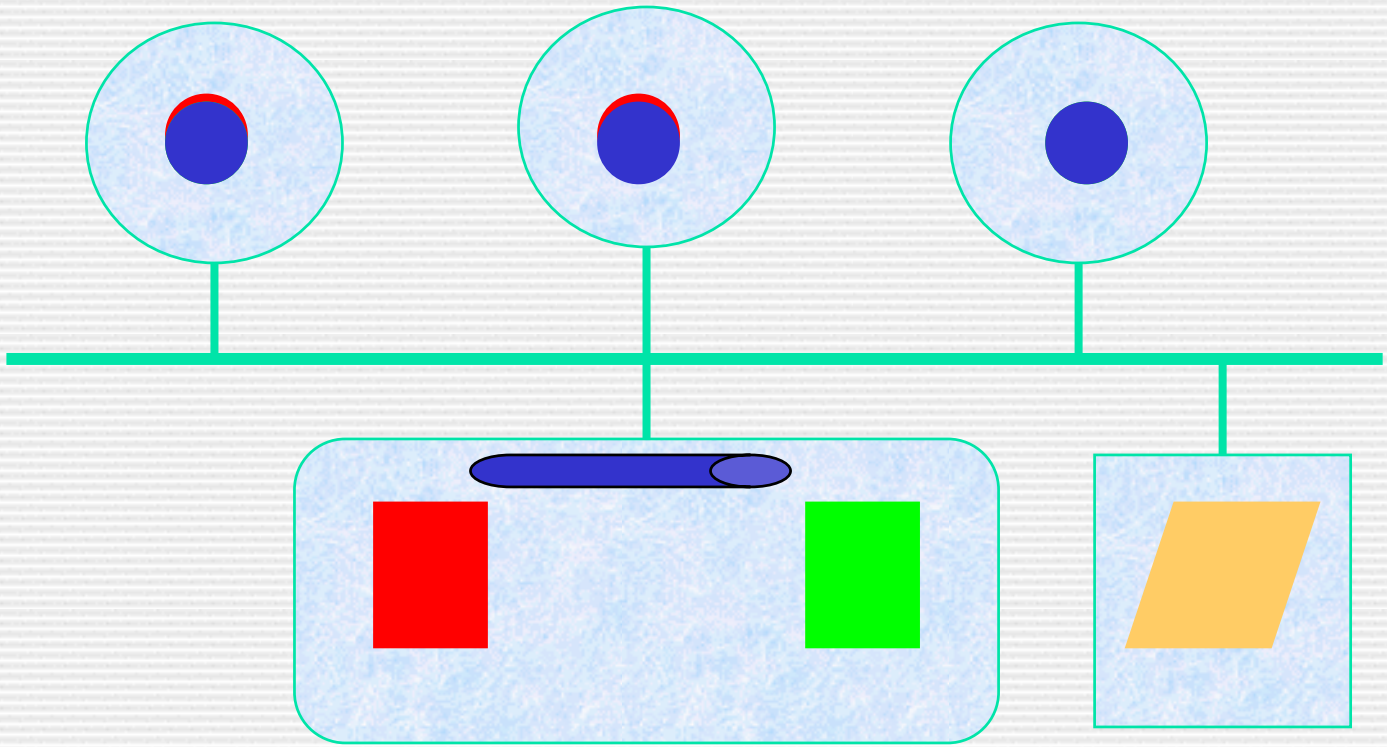
Christine Morin
IRISA/INRIA
PARIS project-team
Christine.Morin@irisa.fr
http://www.irisa.fr/paris

Clusters for Scientific Computing

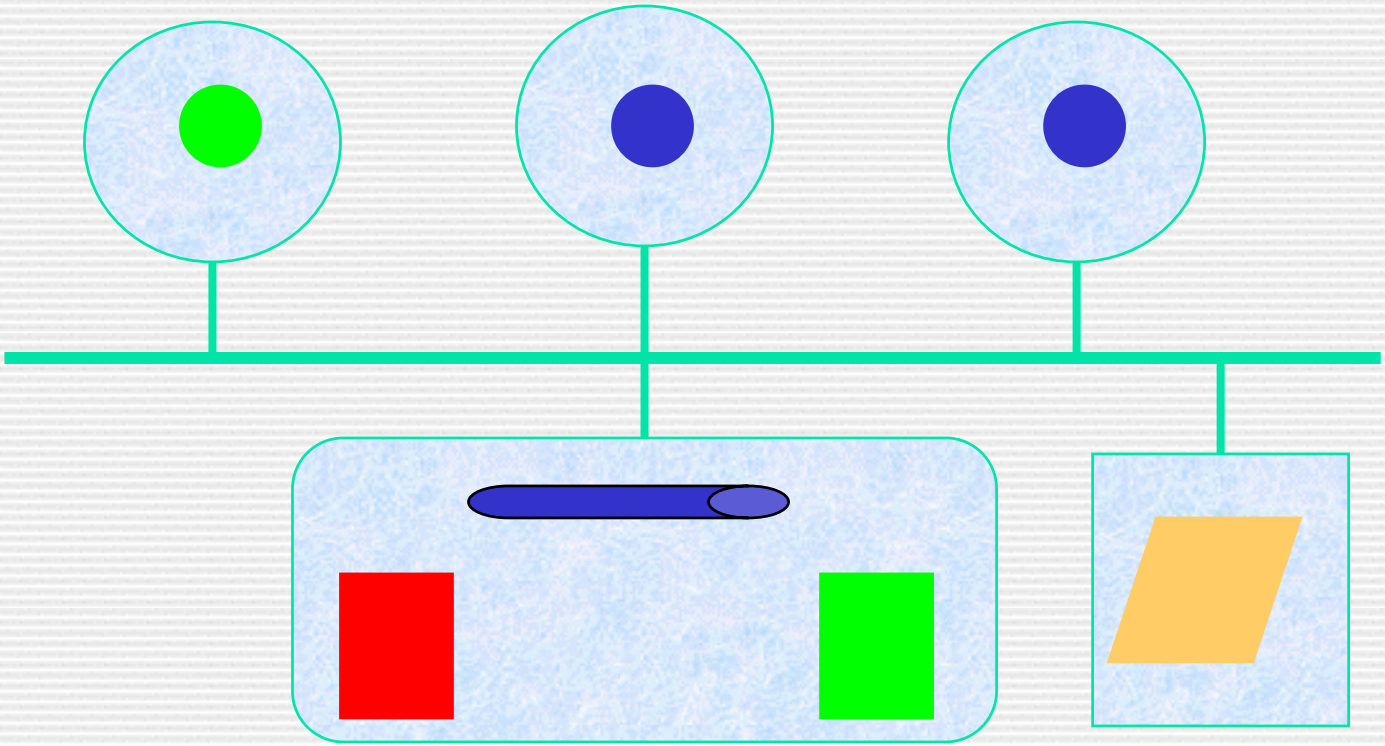
- Clusters are now recognized as a reasonable platform for scientific computing
  - For applications which are not worth executing on expensive machines
  - As departmental computation servers
- Small to medium size clusters are going to be common place computer architectures
  - Up to 64 nodes

Scientific Computing Programming Paradigms

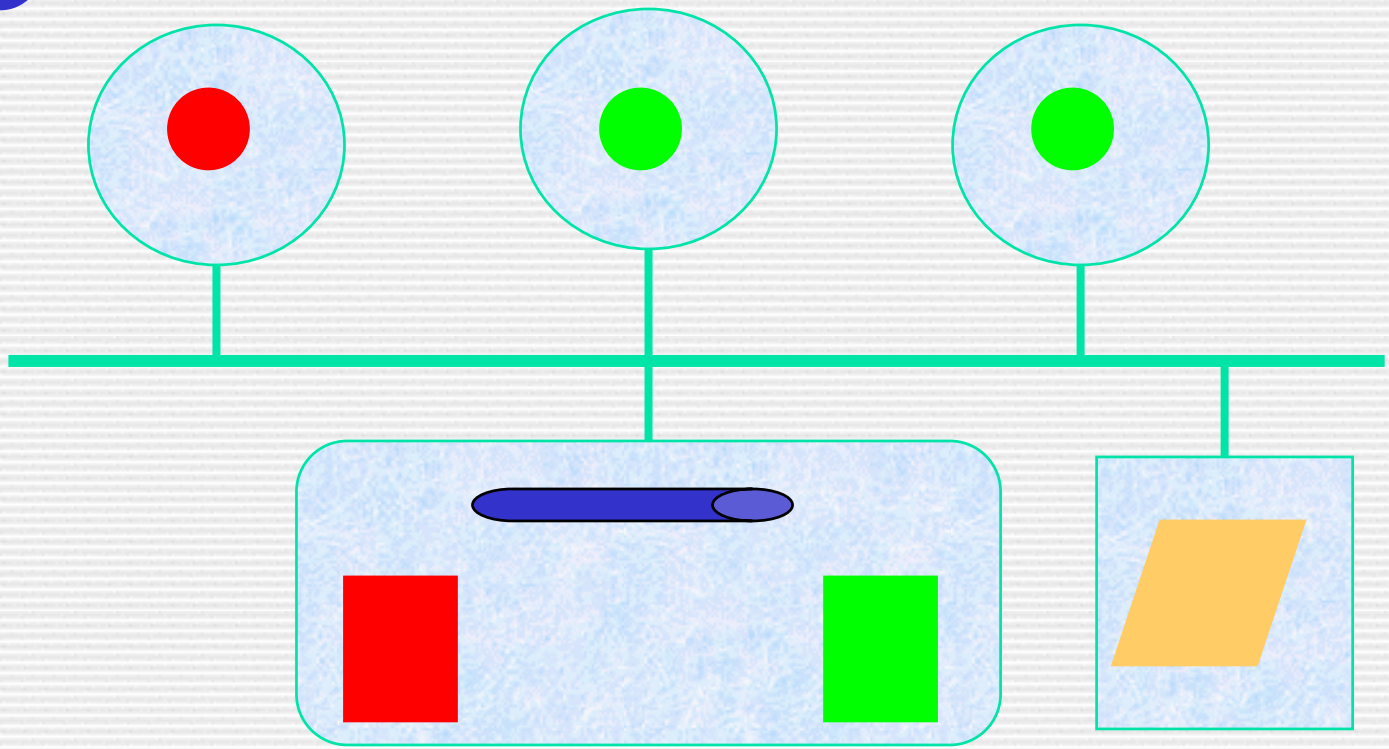
Application Execution on SMP (1/3)



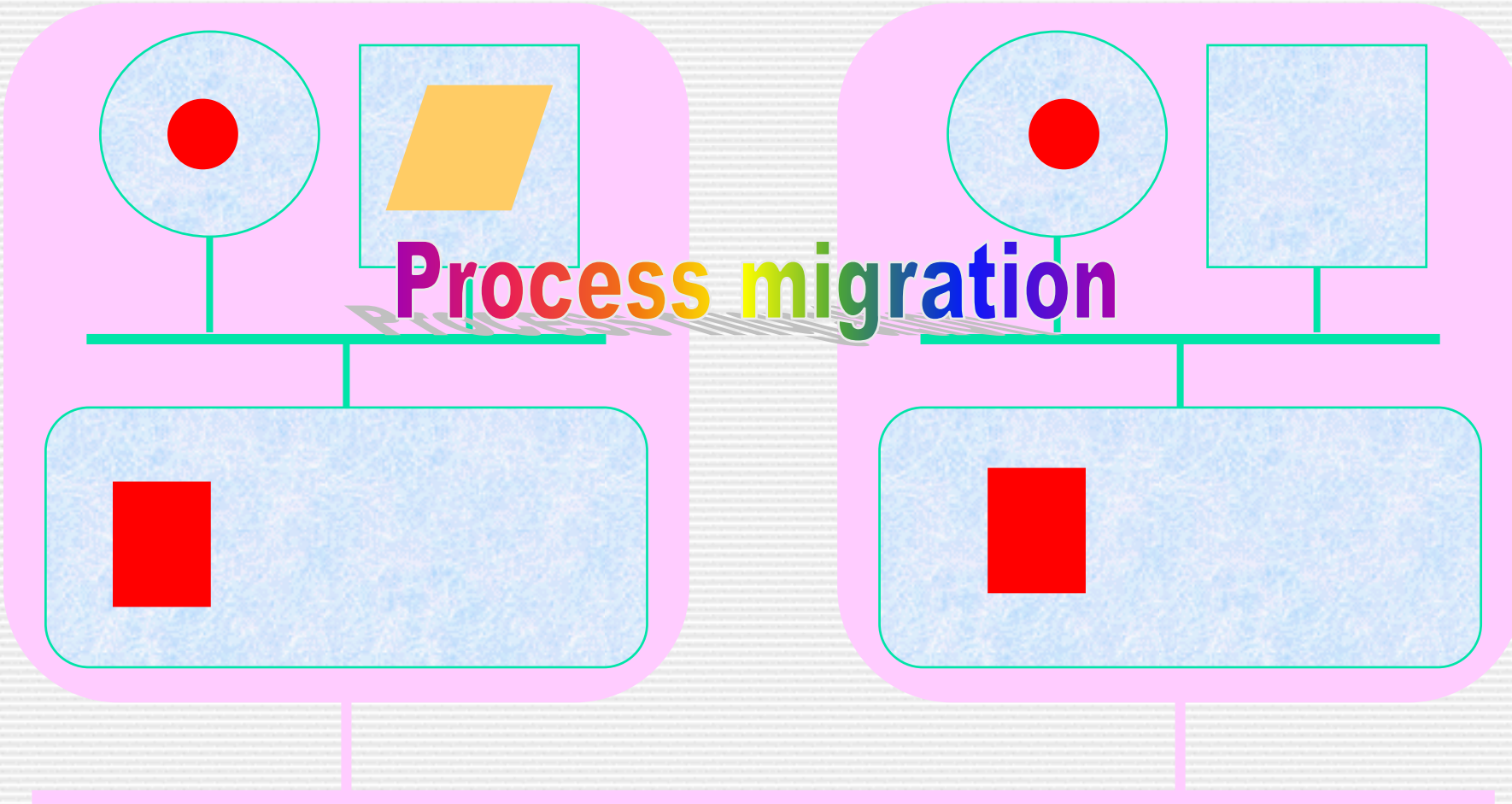
Application Execution on SMP (2/3)



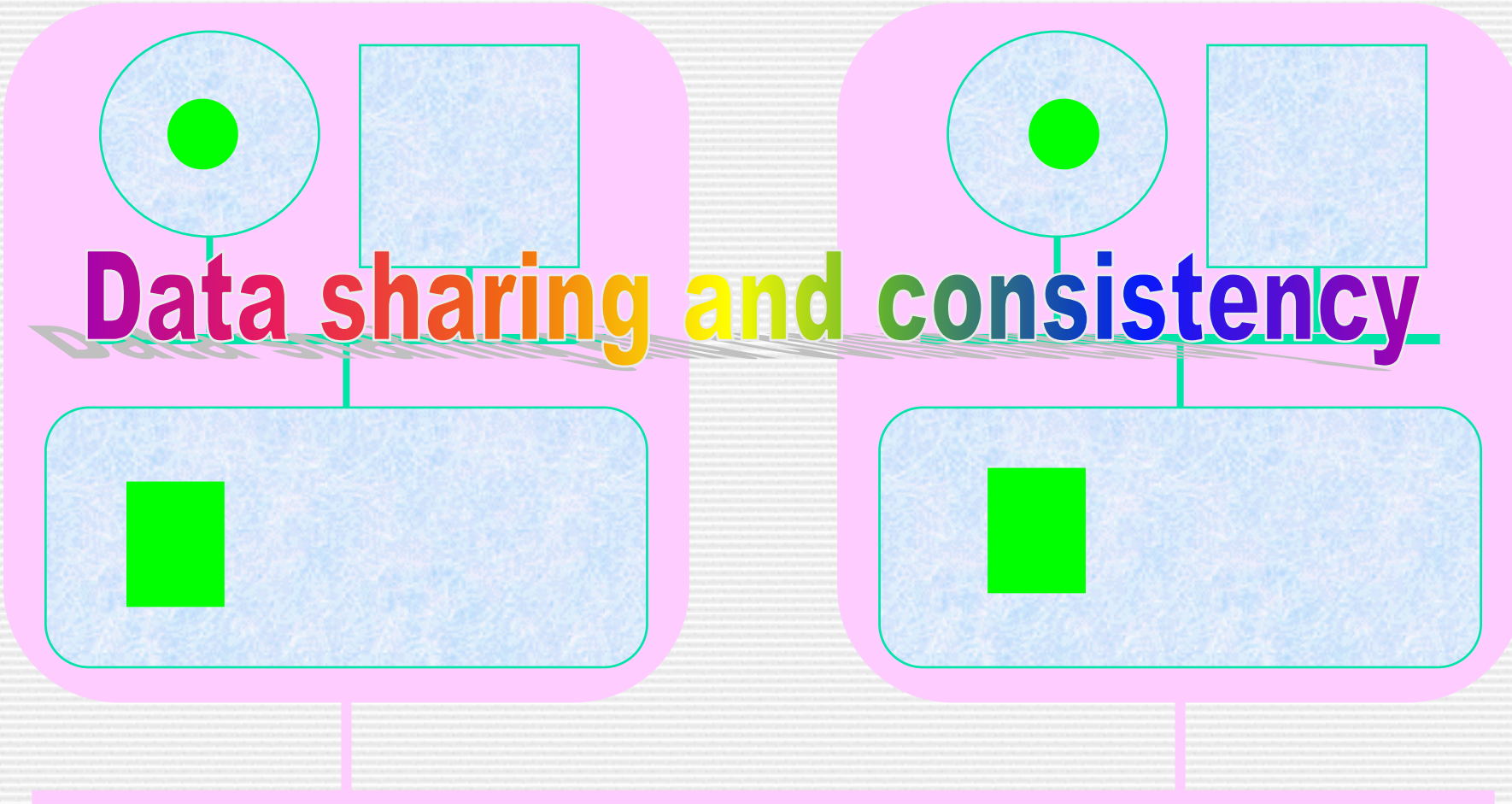
Application Execution on SMP (3/3)



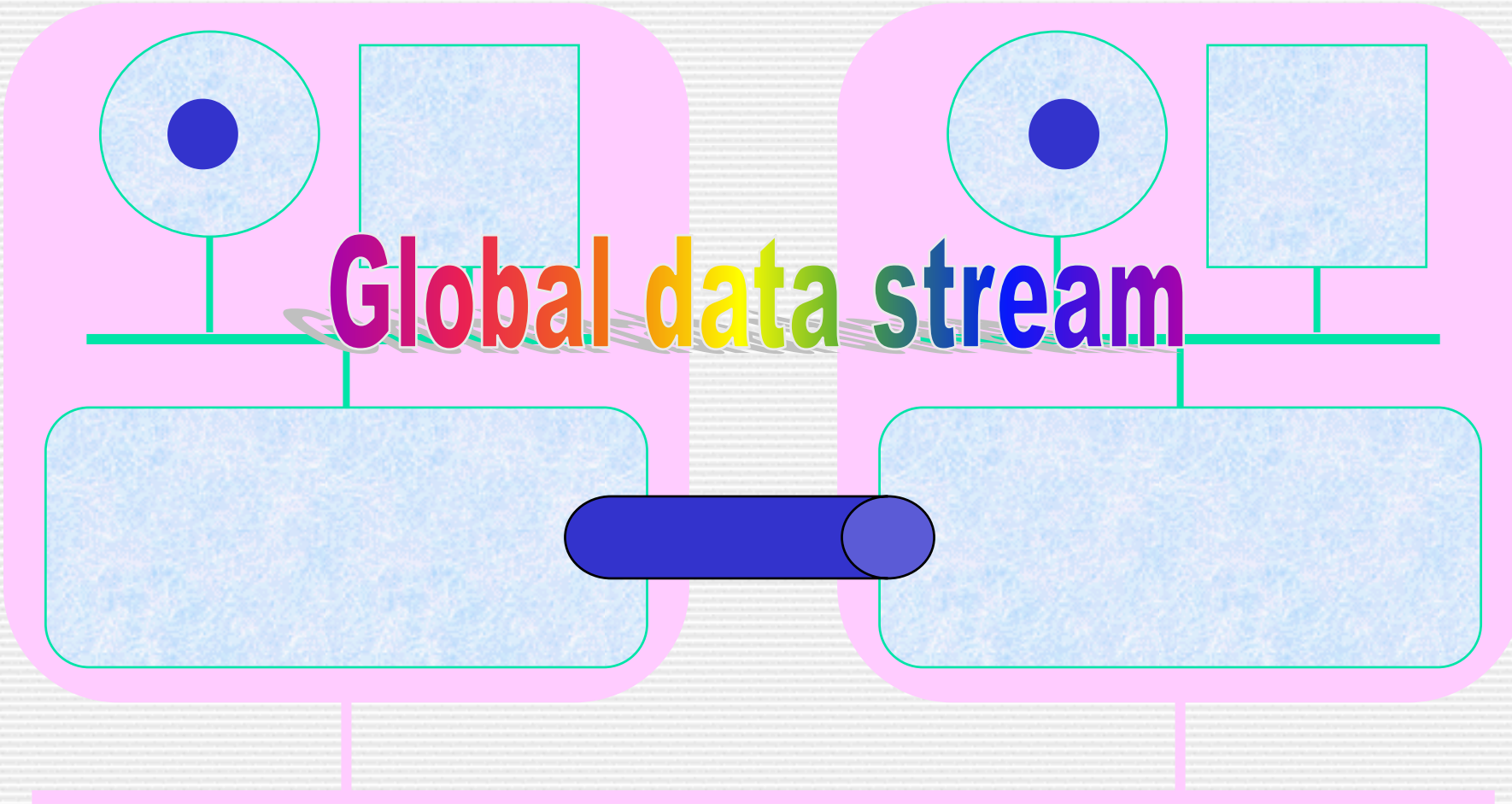
Application Execution on Clusters



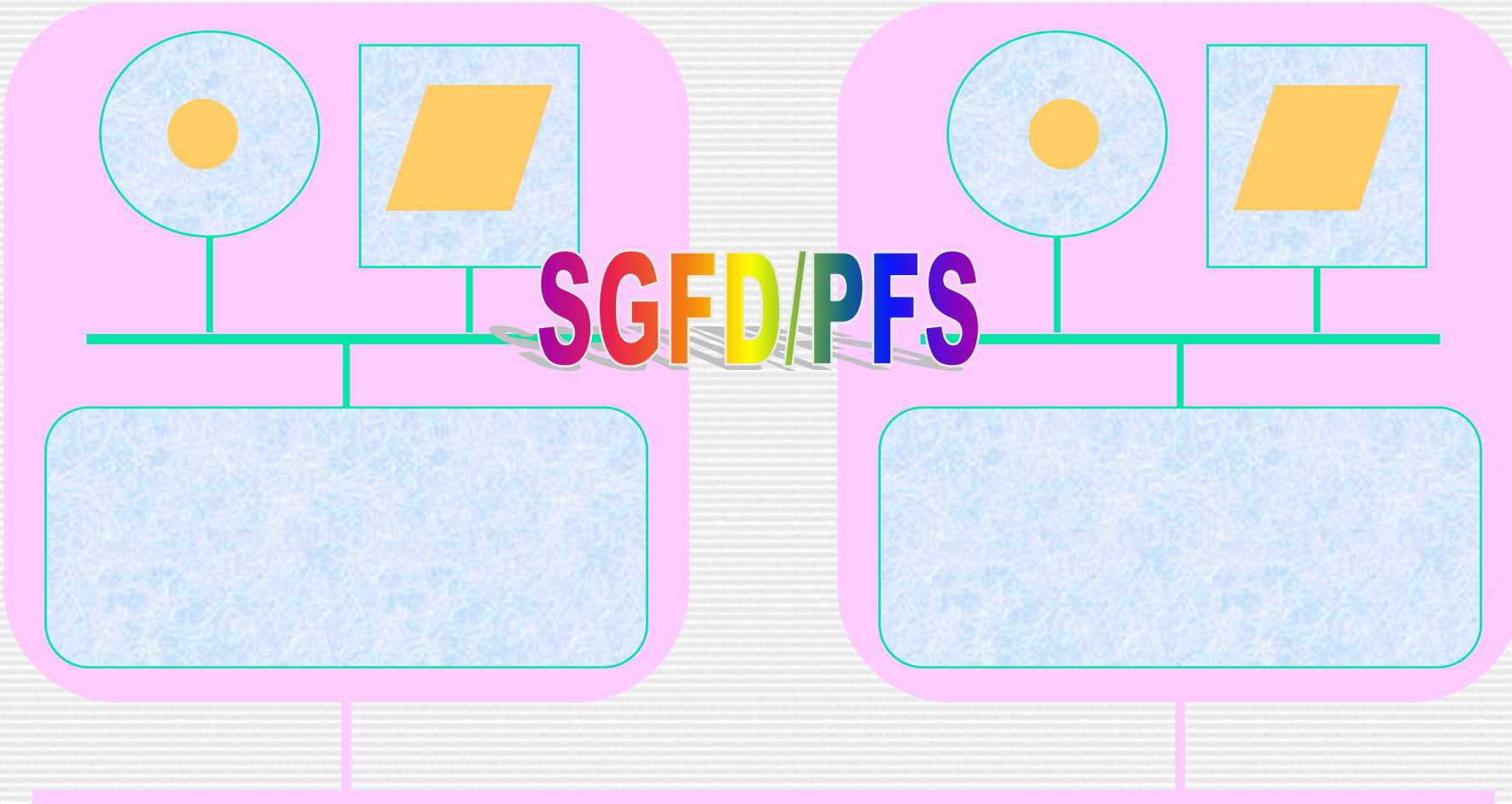
Application Execution on Clusters



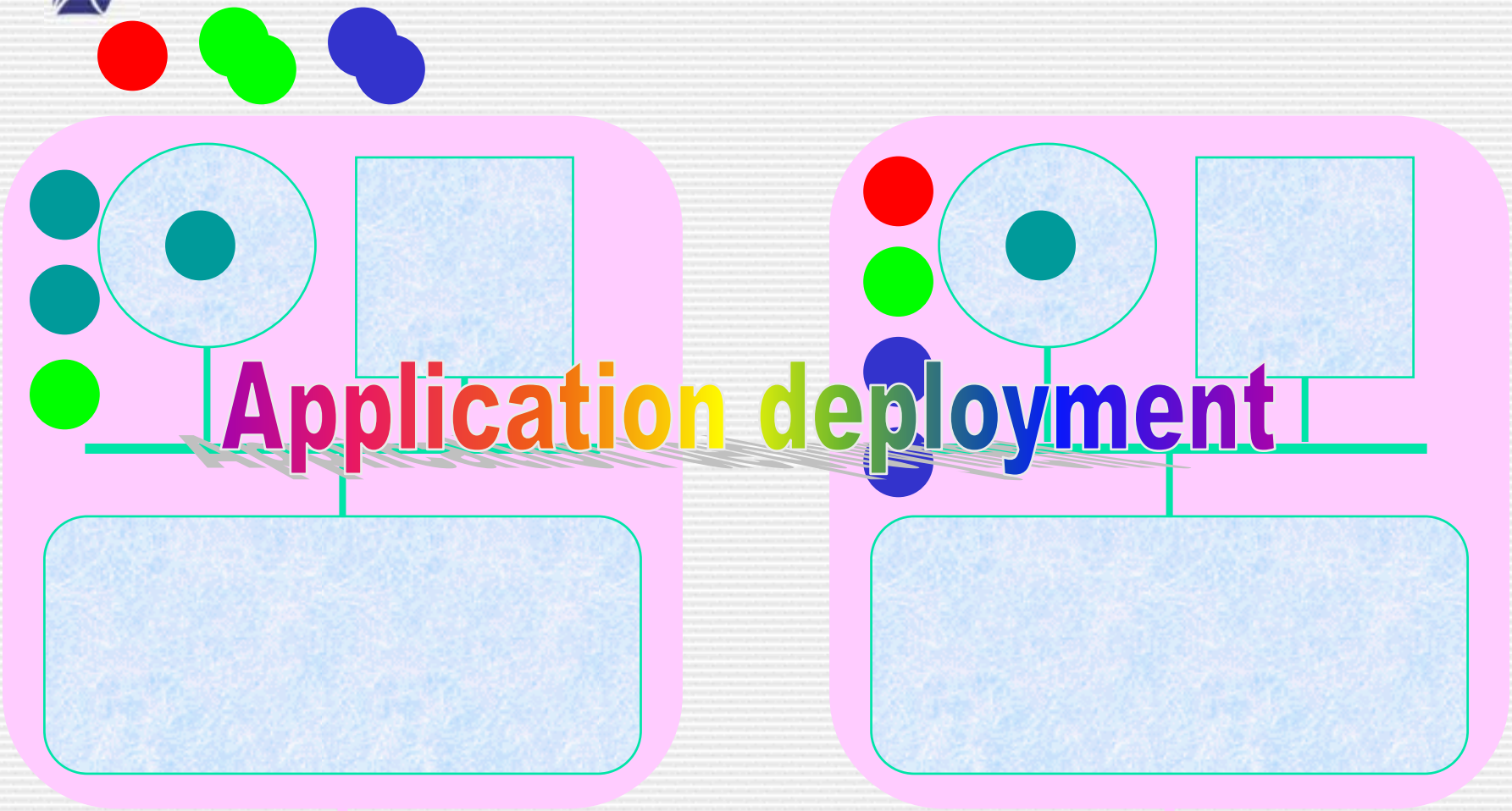
Application Execution on Clusters



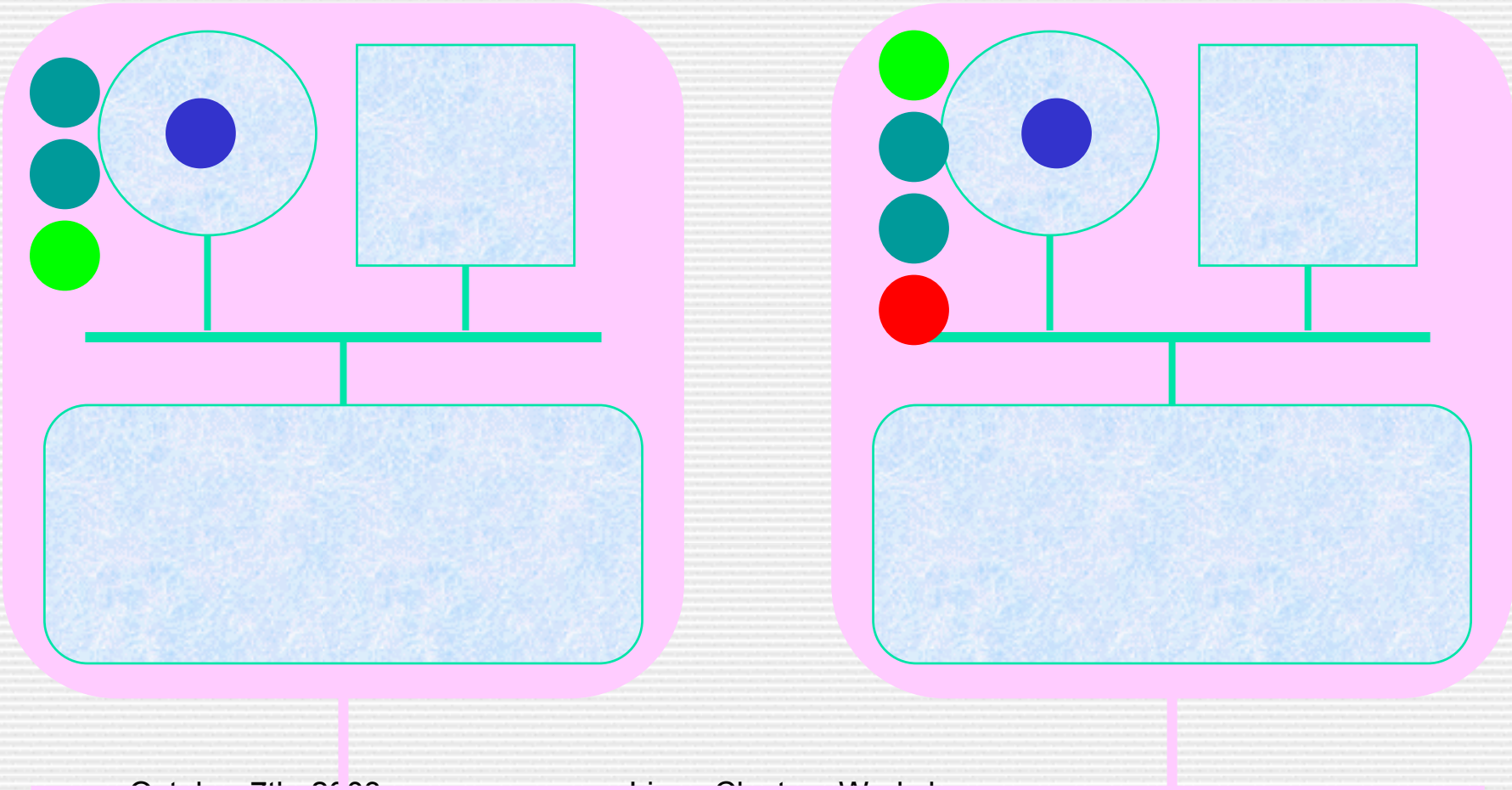
Application Execution on Clusters



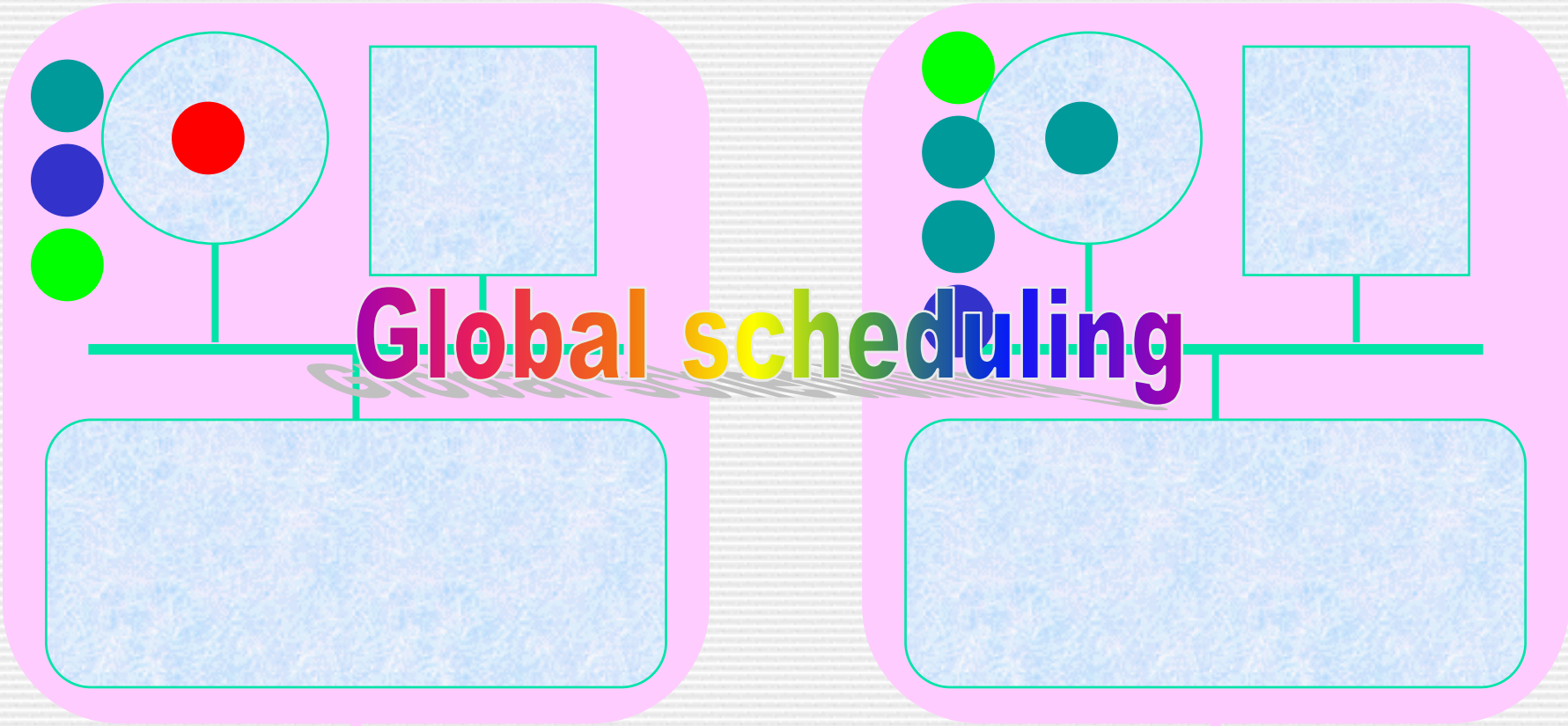
Application Execution on Clusters



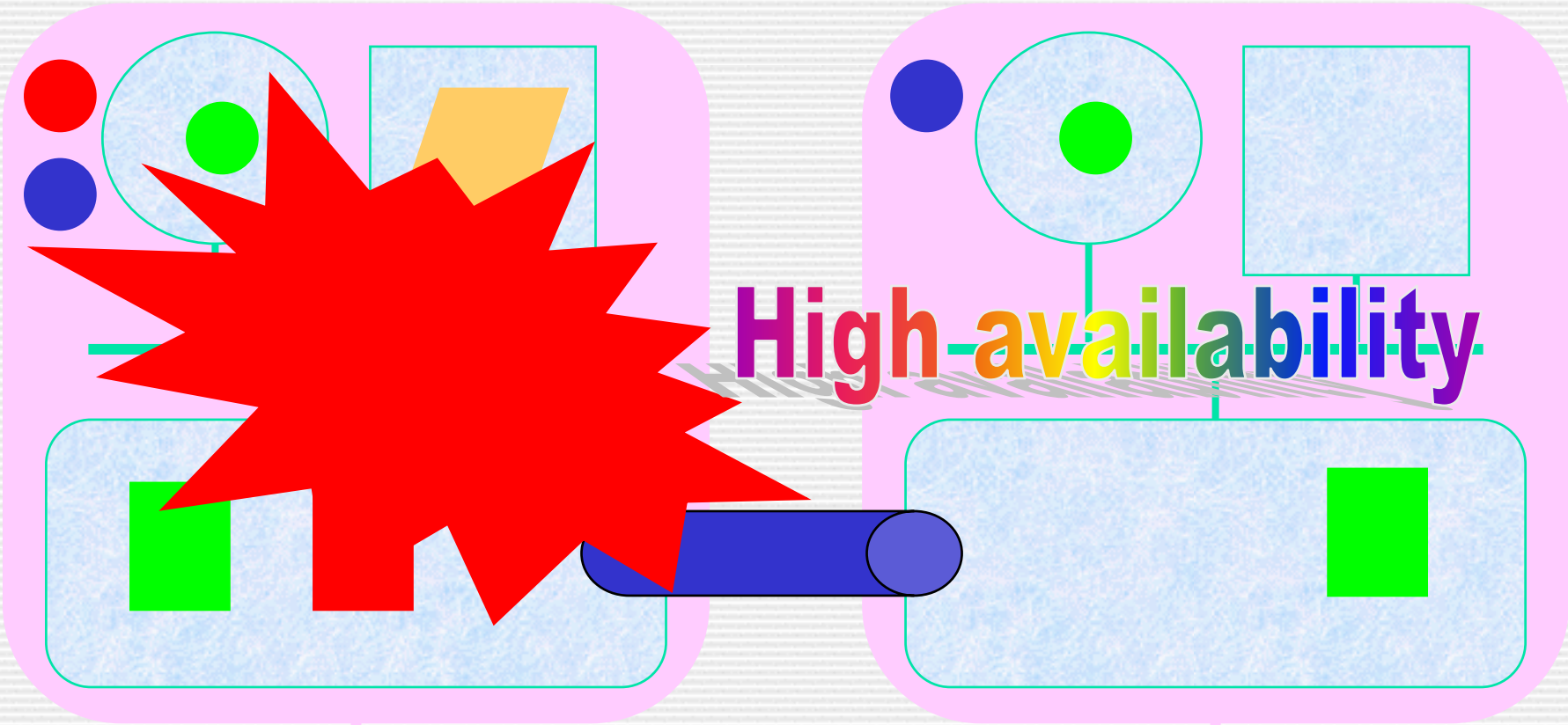
Application Execution on Clusters



Application Execution on Clusters



Application Execution on Clusters



Single System Image

- Virtual SMP
  - Same interface as a traditional OS for an SMP machine
  - Same vision for all applications
  - Efficiency
- Properties of a SSI OS
  - Resource distribution transparency
  - Intra- and inter- application resource sharing
  - High availability

Kerrighed

- Combining high performance, high availability and ease of programming
  - Global resource management
    - Processor, memory, disk
  - Integrated resource management
  - Dynamic resource management
    - To deal with configuration changes
- Small clusters
  - < 100 nodes
- Extension of the standard OS running on each node
  - Linux based prototype

Kerrighed Features

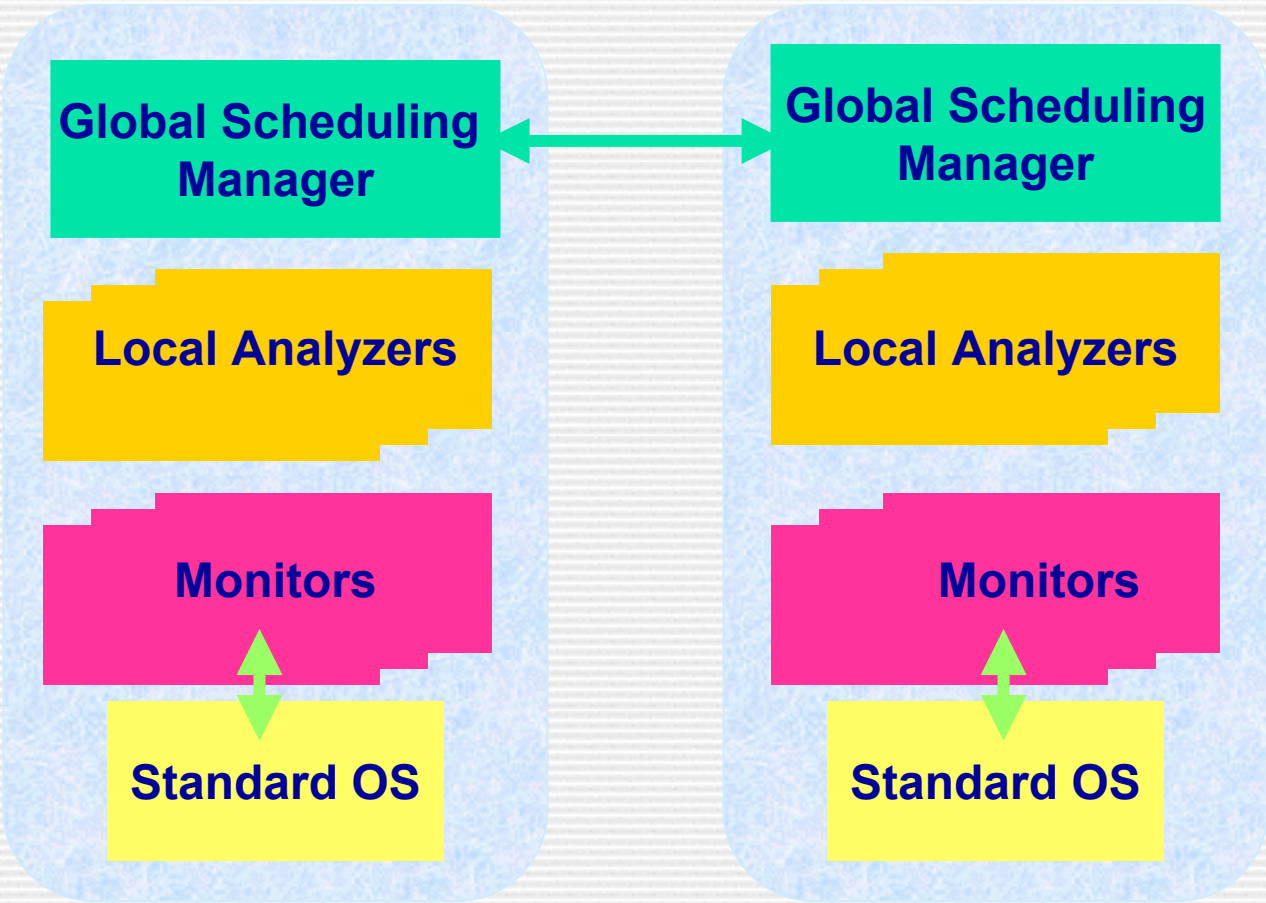
- Configurable global scheduler for global process management
- KerNet for global data stream management
- Containers for global memory management
- Checkpointing
- Portable high performance communication system

Kerrighed provides a full Pthread interface on a cluster

Configurable Global Scheduler

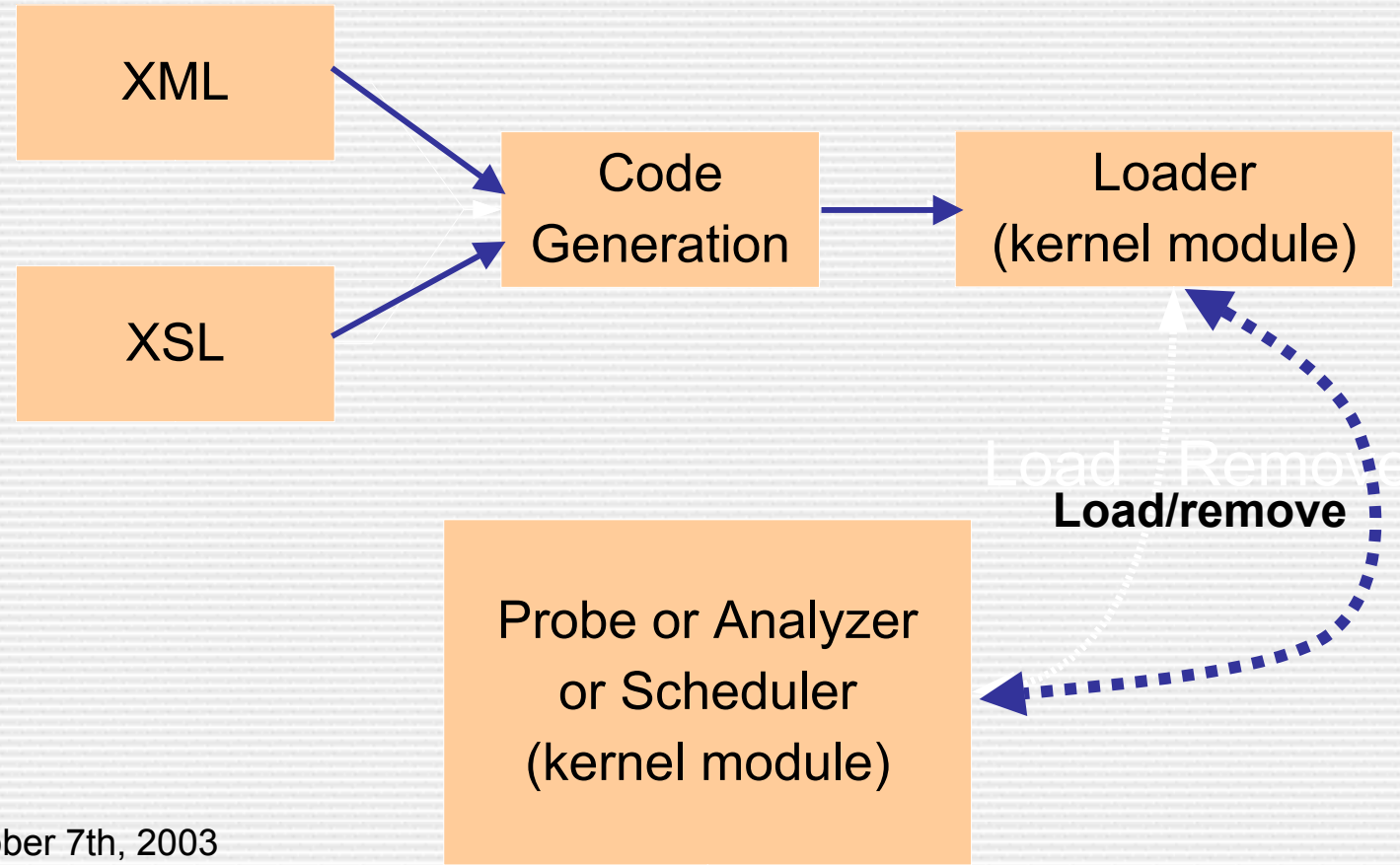
- Design goals
  - It should be possible to implement any traditional placement or load balancing strategy
    - Development and integration of global scheduling policies should be easy
      - Development environment
      - Modular architecture
    - Dynamic configuration of the global scheduler
      - Without stopping the system and the applications
    - Not only configurable but adaptive global scheduler
  - Efficient process management mechanisms
    - With minimal modifications to the OS kernel
      - No modification to the local OS scheduler

Modular Global Scheduler

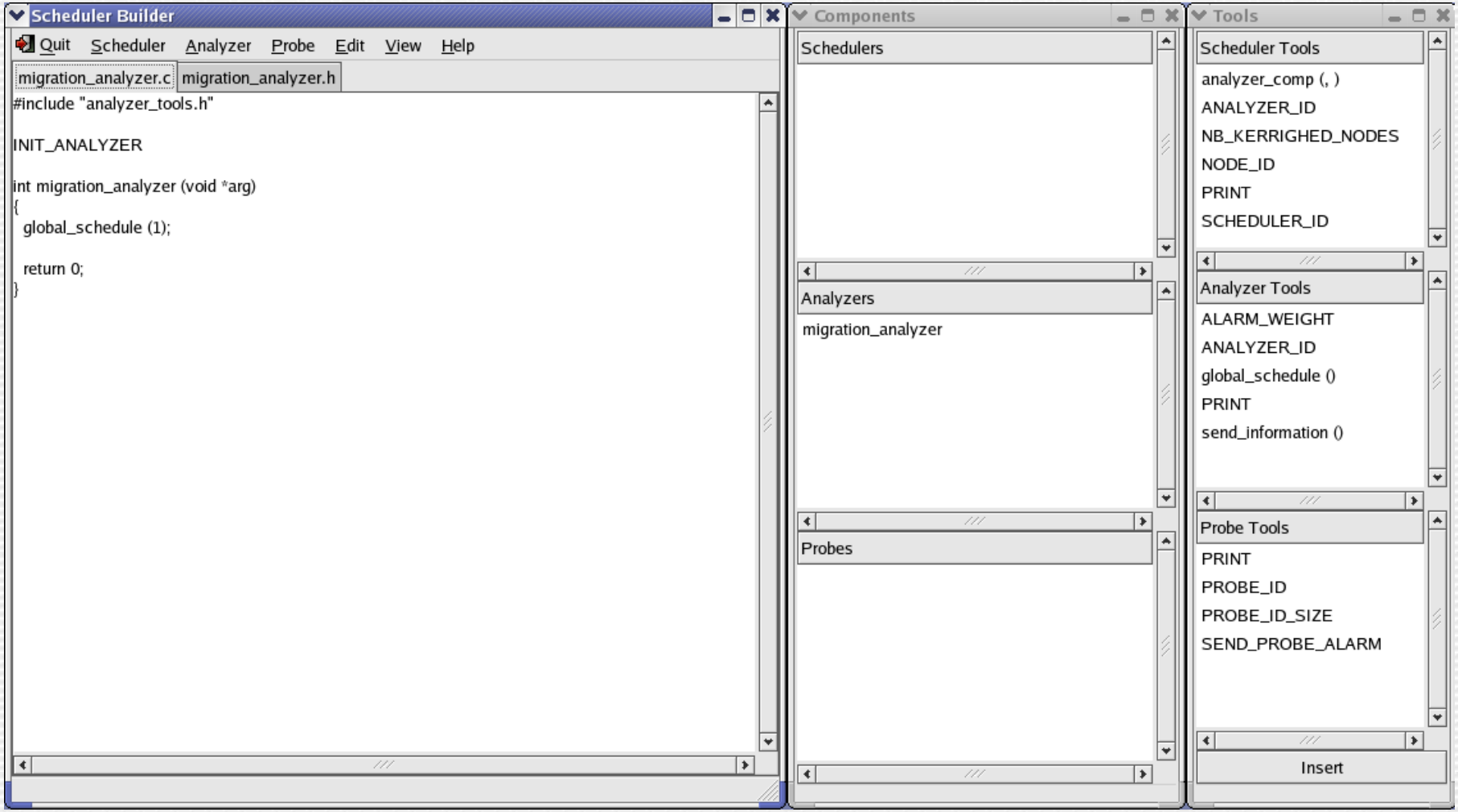


Configuration

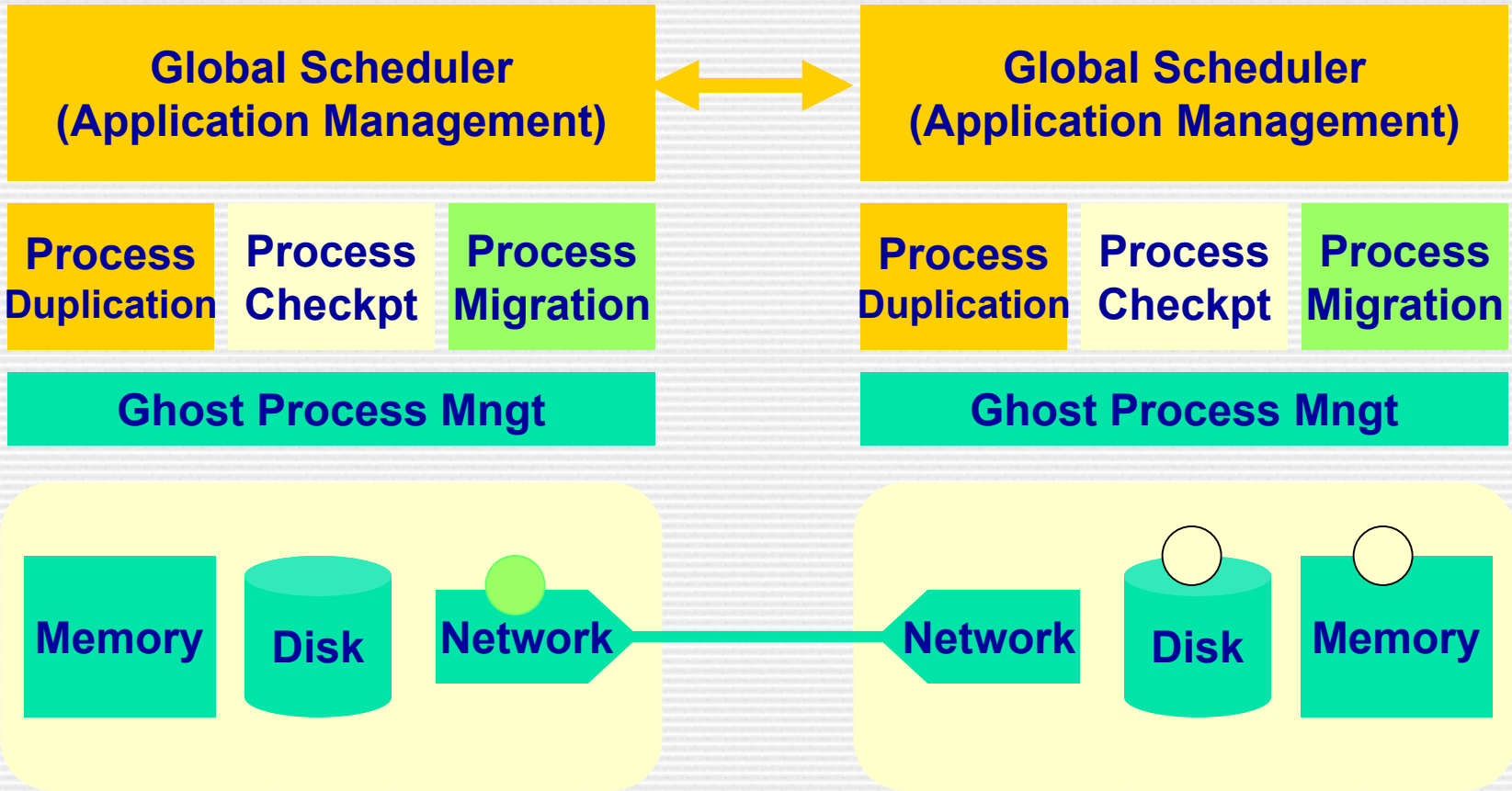
- All components are configured with XML files
- All components can be hot-loaded and hot-removed



Development Environment



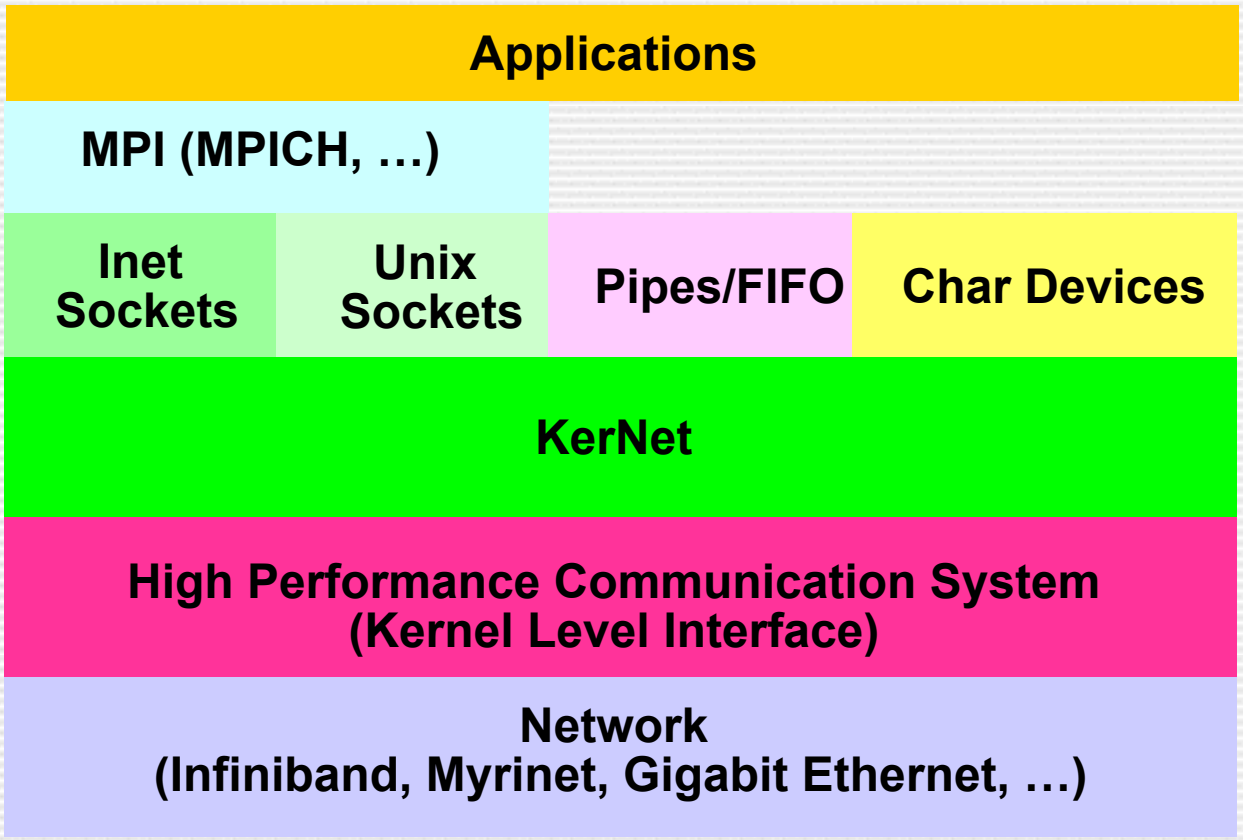
Process Management Mechanisms



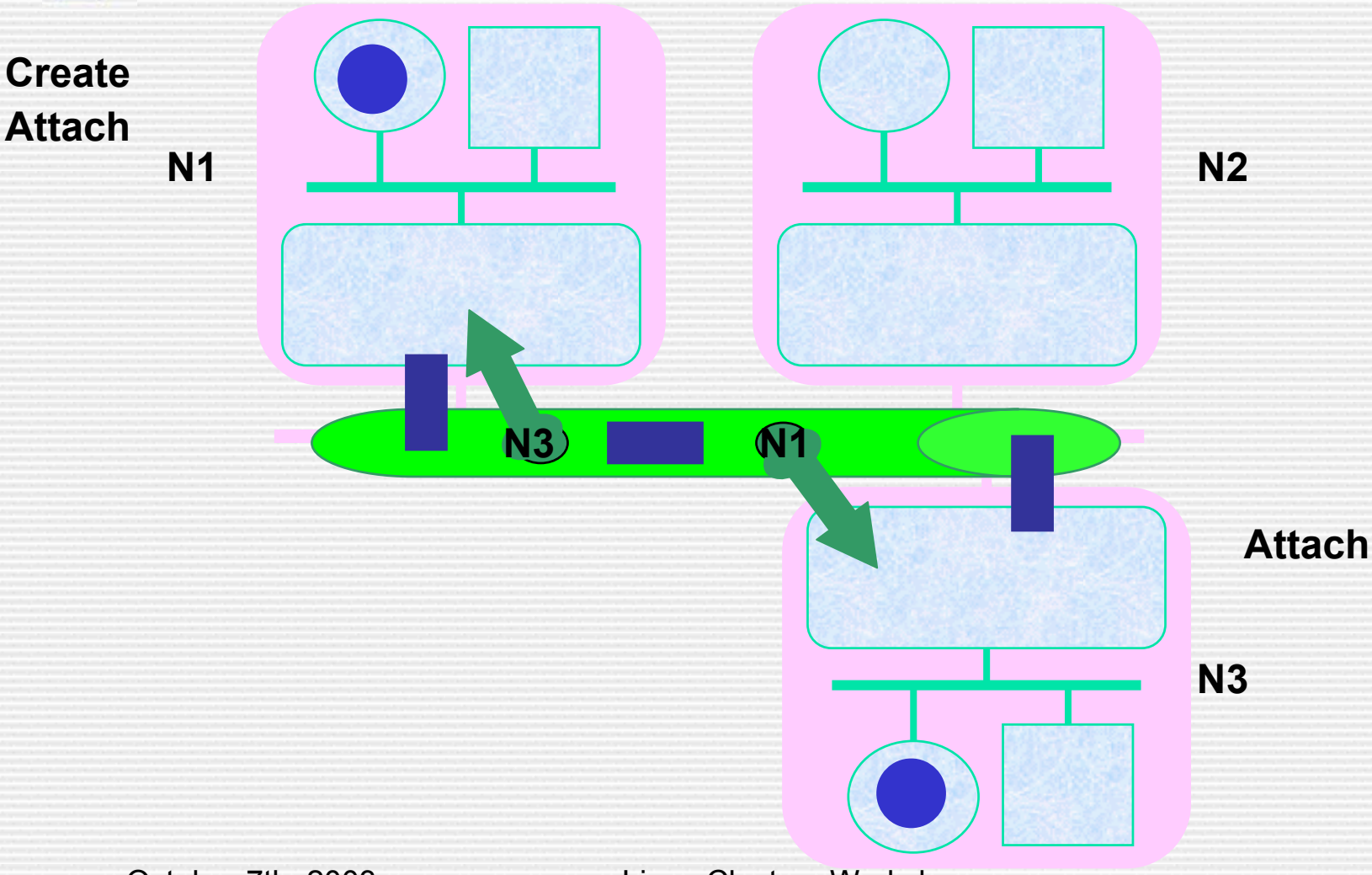
Global Data Streams

- Message communication
  - Pipe, FIFO, Unix sockets
  - Inet sockets (UDP, TCP)
  - Char devices
- KerNet
  - Efficient migration of processes using message communication inside the cluster
  - No modification to applications
  - Dynamic streams & KerNet sockets

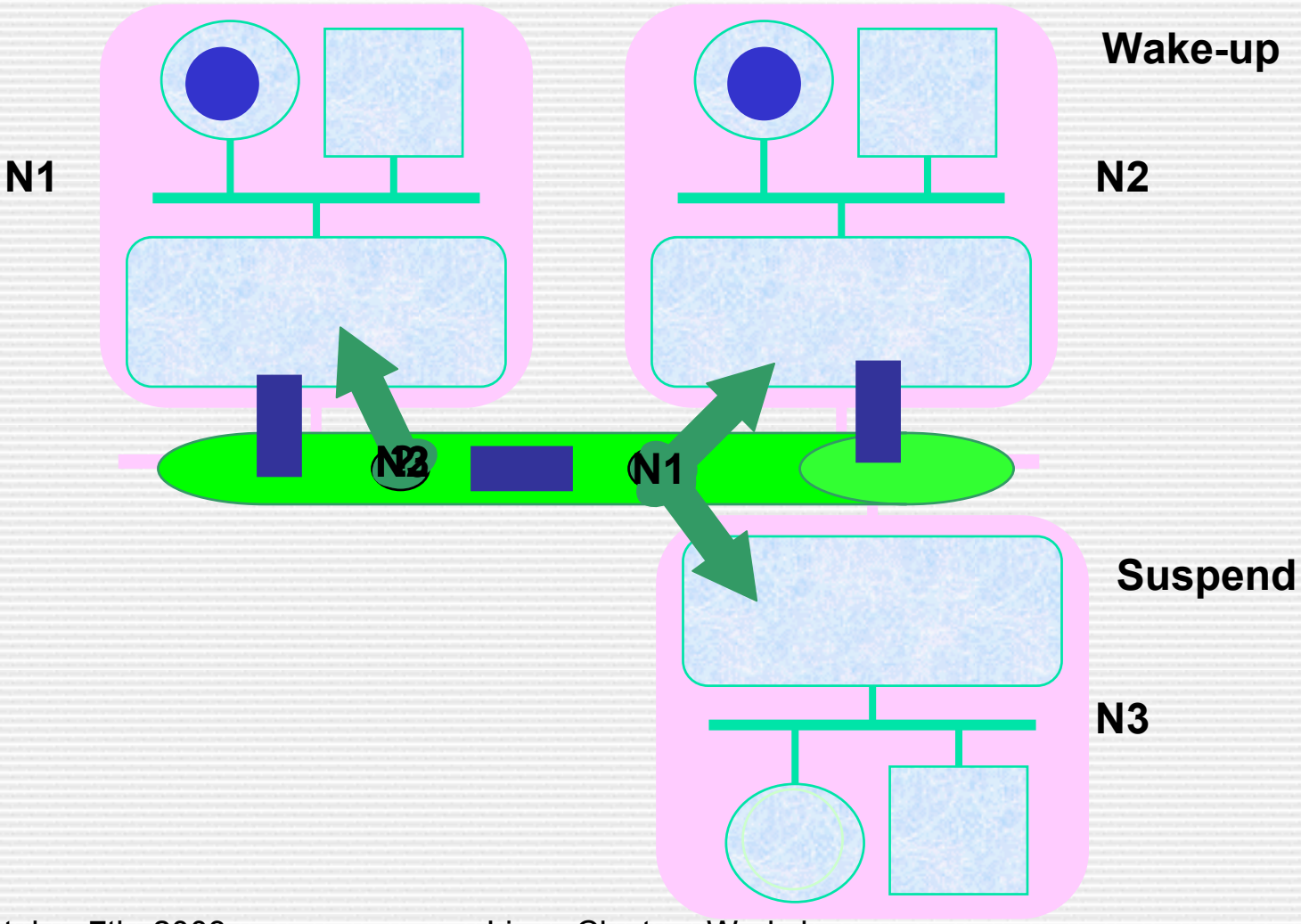
Communication Architecture



KerNet



KerNet

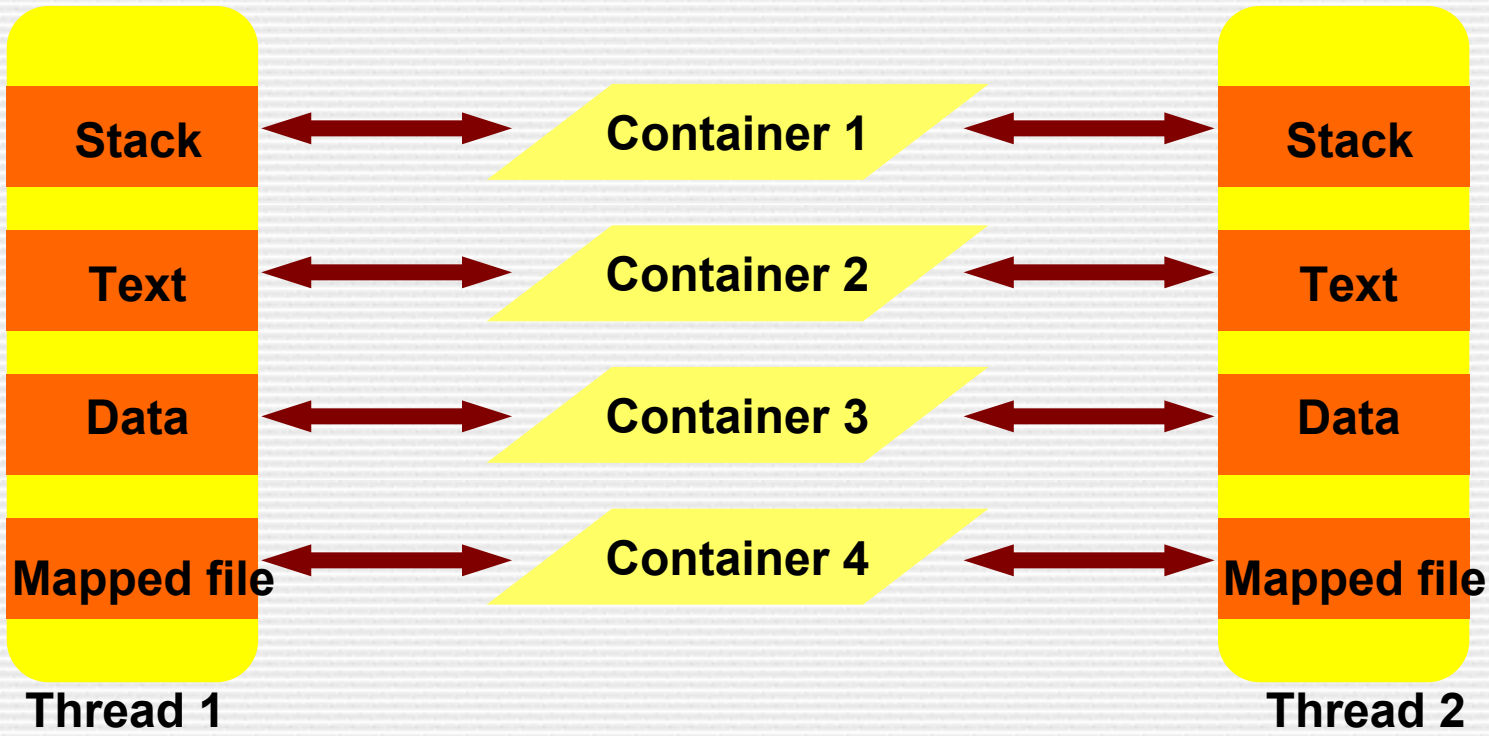


Global Memory Management

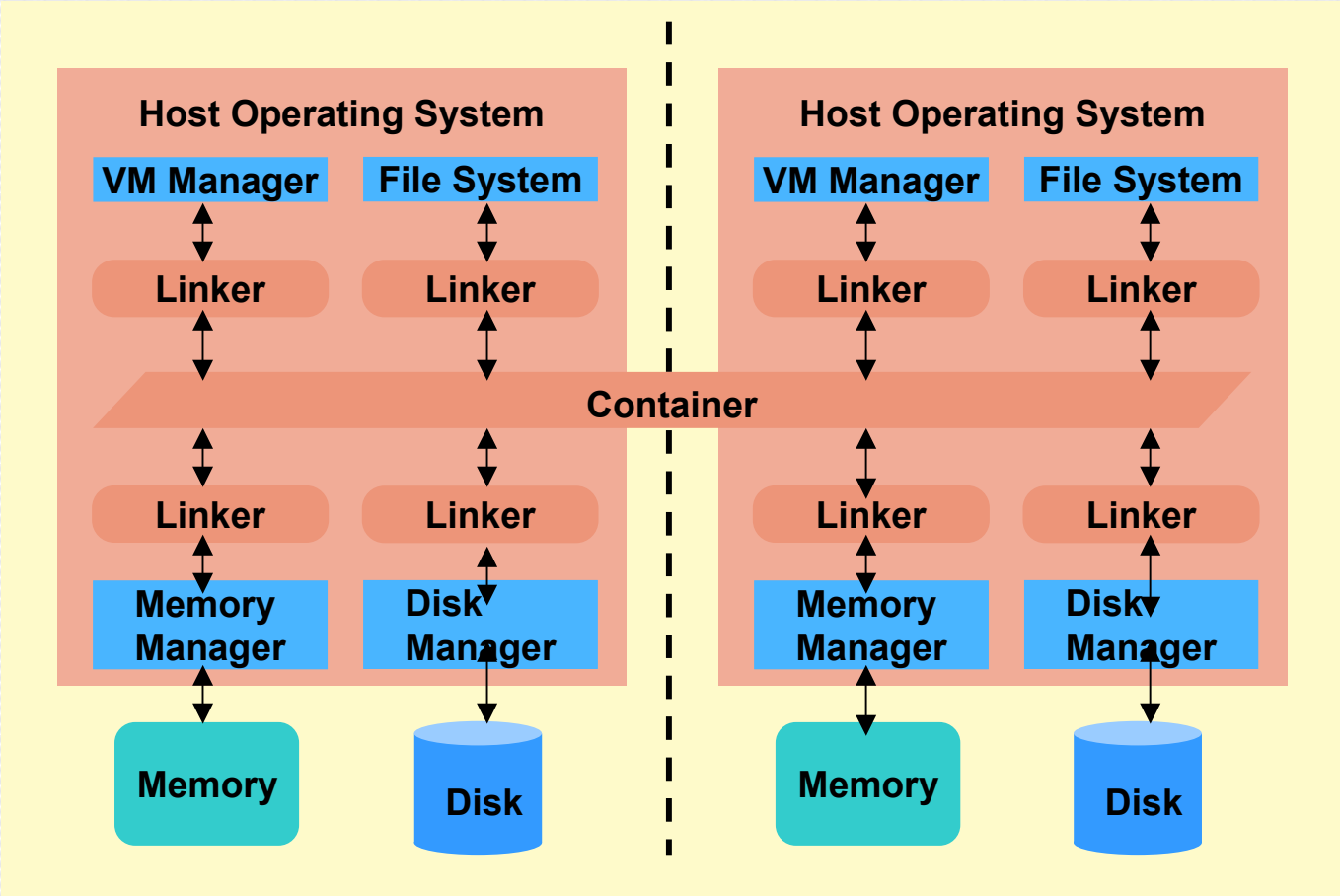
- Different services
  - Shared virtual memory
  - Remote paging
  - Cooperative file cache
- A unique concept: the container
  - Software object to store and share data cluster wide
  - Global management of physical memory
- Memory segments and files are associated to containers

Data Sharing

- Kerrighed implements a kernel level DSM based on containers
  - Sequential consistency, page granularity
- The complete address space of a process is shared including the stack of each of its threads

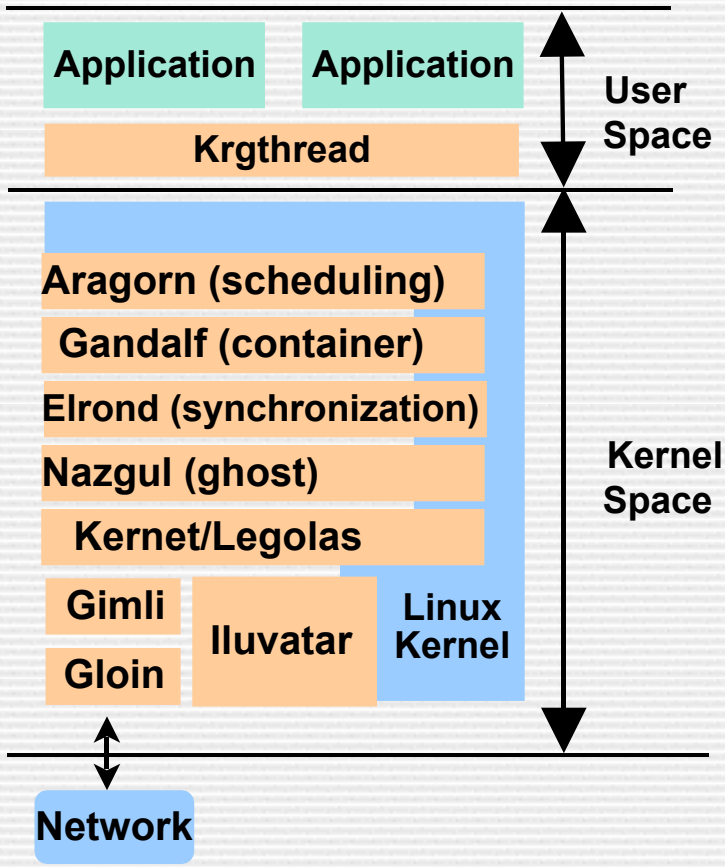


Integration of Containers in a Standard OS



Kerrighed Prototype

- **Extension of Linux kernel**
  - **9 modules** (68 000 lines)
    - Process & load balancing (Aragorn, 20000 lines)
    - Containers (Gandalf, 11000 lines)
    - Synchronization (Elrond, 8000 lines)
    - Ghosts (Nazgul, 3500 lines)
    - Communication (Gimli, Gloin, 14000 lines)
    - KerNet (4700 lines)/Legolas(2500 lines)
    - Tools (Iluvatar, 4000 lines)
  - **Limited patch to the kernel** (300 lines)
- LibKrgthread (2000 lines)



140 M/M since 1999

Conclusion

- Kerrighed: first Linux based cluster OS providing the illusion of a virtual SMP
  - Full Pthread support
    - **OpenMP, multithreading**
  - MPI
  - Configurable adaptive global scheduler for process placement and migration
  - Transparent checkpointing
- Kerrighed V0.72 available as an open source software under the GPL licence (http://www.kerrighed.org)
  - 100 downloads since mid-November 2002

Perspectives: Research Directions

- High performance I/O
  - Exploitation of cluster standard disks
- High availability
  - Transparent cluster reconfigurations after node addition, eviction or failure
- Grid-aware OS for cluster federations
  - P2P infrastructure with clusters as nodes
  - Large scale data sharing
  - Resource allocation
  - Scalable checkpointing algorithms
  - Security

Perspectives: Technology Transfer

- Kerrighed research prototype (2000-2003)
  - CRECO EDF/INRIA
    - CIFRE Ph.D. grant (Geoffroy Vallée)
    - Industrial Post-Doc (Renaud Lottiaux)
  - Experimentations with first industrial applications provided by EDF
    - HRM1D, CATHARE, Cyrano 3, Aster
- Kerrighed robustness and full set of functionalities (2003-2005)
  - COCA PEA funded by DGA
    - Partnership with CGEY and ONERA-CERT
    - 2 full time engineers (Renaud Lottiaux, David Margery)
  - Experimentations with industrial applications
    - Ligase, Gorf3D, Mixsar, RTI HLA

Perspectives: Technology Transfer

- Next Step: Kerrighed durability
  - Including Kerrighed in a Linux Distribution for high performance computing (OSCAR, …) ?
  - Kerrighed development consortium ?
  - Transfering Kerrighed to a company involved in cluster construction and software development ?

Kerrighed is registered as a community trademark.

http://www.kerrighed.org

kerrighed.users@irisa.fr

Kerrighed Team

- Faculty
  - Christine Morin (DR, INRIA)
- PhD students
  - Geoffroy Vallée (CIFRE-EDF)
  - Pascal Gallard (INRIA)
  - Gaël Utard (INRIA)
  - Louis Rilling (ENS-Cachan)
- Engineers
  - Renaud Lottiaux (INRIA)
  - David Margery (INRIA)
- Former member
  - Ramamurthy Badrinath (IIT Kharagpur, India)
    - May 2002 – April 2003