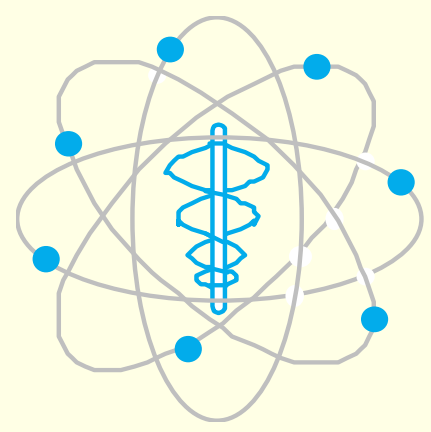
Plurix & PROCOPE



- PROCOPE Goals for Plurix
- Status of Fault Tolerance
- Working Plan for the Year 2004
- Working Plan for the Year 2005

M. Schoettner, S. Frenz, R. Goeckelmann, M. Fakler, and P. Schulthess
Distributed Systems, University of Ulm, Germany

PROCOPE goals for Plurix

- A lot of (DSM) checkpointing theory is found in literature.
- But only a few have been implemented & evaluated with real applications.
- **Plurix offers new perspectives for checkpointing and recovery**:
  - Distributed Shared Memory stores code and all data.
    → local process and kernel contexts are avoided.
    → it is sufficient to checkpoint the DSM.
  - restartable transactions instead of processes and threads
    → restartability is a base feature of the OS.
    → recovery is simplified.

- **Major PROCOPE goal: comparison of checkpointing & recovery**
  - in Mome, Kerrighed, and Plurix,
  - running the same parallel applications.
- **Benefits of the PROCOPE cooperation**:
  - deep evaluation of each DSM system,
  - pros and cons of an environment are revealed,
  - hints for future DSM systems and OS architectures.

Status of Fault Tolerance

- Global coordinated checkpointing:
  - PageServer is responsible for periodic checkpointing,
  - writes only changed memory pages synchronously to disk,
  - a network snooper reduces traffic and saves time during checkpointing.
- Booting and shutdown of nodes using checkpoints is implemented.
- Nodes can be rebooted via a running DSM in 250ms.
- Error detection basing on commit number is implemented.
- Currently, errors require a manual cluster reboot.
- Recovery of device states is an open issue:
  - e.g. a lot of textures may be stored in graphics adapter memory,
  - device states are lost in case of a node reboot.

Working Plan for the Year 2004

- Phase 1: coordinated checkpointing with a central PageServer
  - error detection and DSM recovery.
  - study strategies for device state recovery.
  - check portability of DSM-based applications (e.g. Splash-2).
- Phase 2: optimized checkpointing
  - do not stop cluster for finalizing a checkpoint.
  - (distributed PageServer).
- Phase 3: applications
  - implement a few traditional DSM applications available in all three DSM systems.
  - (develop a memory access recorder running in Kerrighed and a player for Plurix).
- Phase 4: performance evaluation (and preliminary comparisons)
  - using the applications from Phase 3,
  - (and may be recorder/player),
  - and the Plurix ray tracer, ...

Working Plan for the Year 2005

- Phase 1: multiple consistency models
  - extend the existing DSM protocol to allow co-existence of other consistency models.
  - implement sequential consistency for specially allocated memory blocks.
- Phase 2: applications
  - adapt applications from 3/2004 to sequential consistency.
- Phase 3: optimize checkpointing
  - logging,
  - dependency tracking,
  - distributed page server with independent checkpointing.
- Phase 4: **performance evaluation & comparison of all DSM systems**
  - write a publication together,
  - with different checkpointing options,
  - with same applications using sequential consistency.

Conclusion

- We hope to achieve a highly efficient checkpointing strategy for all involved DSM systems.