Distributed Peer-to-Peer Control in Harness

Christian Engelmann

Network and Cluster Computing Group,
Computer Science and Mathematics Division,
Oak Ridge National Laboratory, USA

engelmannc@ornl.gov
www.csm.ornl.gov/harness

Outline

- Harness overview.
- Distributed control.
- Group communication.
  - Reliable broadcast.
  - Atomic broadcast.
  - Distributed agreement.
  - Membership.
- Transaction types.
- Conclusions and future work.

Harness Overview

- Heterogeneous Adaptable Reconfigurable Networked Systems.
- Developed as a follow-on to PVM (www.csm.ornl.gov/pvm).
- Collaborative research effort between:
  - Oak Ridge National Laboratory, USA.
  - University of Tennessee, USA.
  - Emory University, USA.

Harness Overview

- PVM (parallel virtual machine) successor.
  - Every node in a cluster runs a virtual machine.
  - Master node controls the set of virtual machines.
  - System breaks when master dies!
- DVM (distributed virtual machine) design.
  - All nodes in a cluster form a single virtual machine.
  - They equally control the virtual machine.
  - Distributed global state database with mutual exclusive access provides virtual synchrony.
  - Fault-tolerance by symmetric database replication.

Harness Overview

- Plug-in mechanism for adaptable computing.
  - Every node runs a plug-in loader service.
  - Plug-ins serve other plug-ins or applications.
  - They are fault-tolerant when using the database.
  - Applications may assemble or reconfigure themselves from plug-ins at runtime.
  - A set of base plug-ins provide network access, process control and the distributed global state database - the distributed virtual machine service.

Distributed Control

- Provides access to control global state.
  - Manages membership (group of nodes).
  - Loads, unloads and configures plug-ins.
- Operates the global state database.
  - Global state changes are transactions, which are <u>ordered</u>, <u>executed</u>, and <u>committed</u> or <u>rejected</u> depending on the execution result.
- Offers event distribution service.
  - Notifies plug-ins or applications on member, plug-in or application failure or state change.

Distributed Control

- Supplies system-wide fault-tolerance.
  - Maintains a complete and consistent up-to-date replica of the global state at every node or a subset of nodes (1 < node subset < all nodes).
  - Enables plug-ins and applications with hot-standby (continuous service) or warm-standby (checkpoint and restart service) high-availability.
  - The distributed virtual machine, loaded plug-ins and connected applications survive until at least one member is still alive.

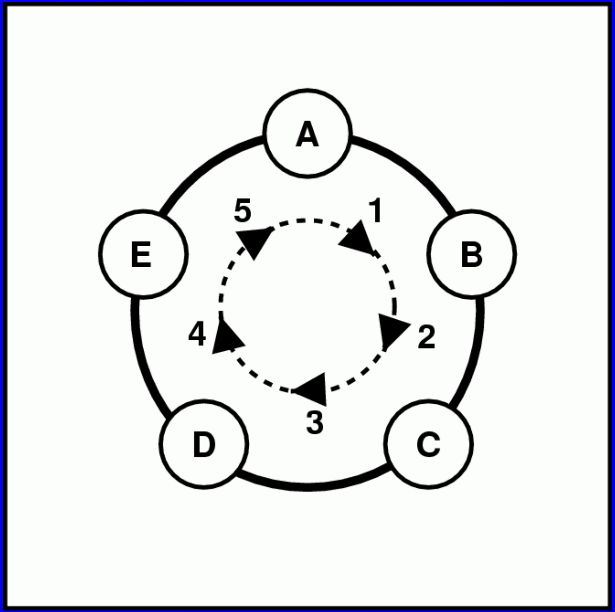
Group Communication

- Ensures fault-tolerant messaging.
- Nodes form one scalable peer-to-peer ring.
- Messages go only in one direction.
- Algorithms perform state replication using symmetric transaction ordering, execution, commit and rejection at the nodes in the ring.
  - Reliable broadcast.
  - Atomic broadcast.
  - Distributed agreement.
  - Membership.

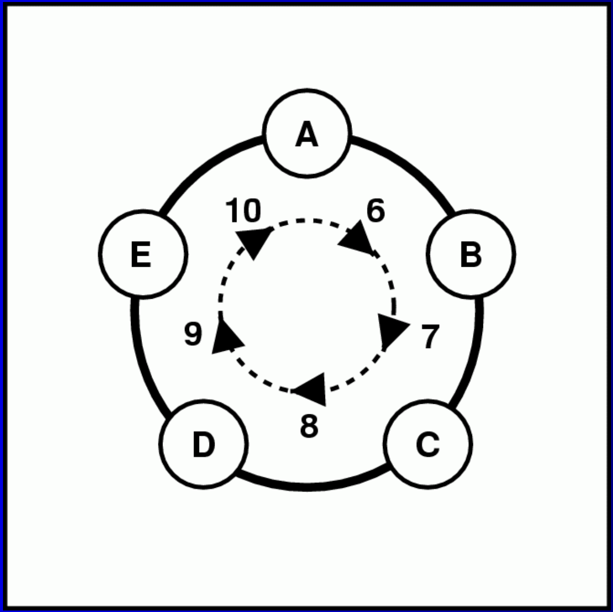
Reliable Broadcast

- Transactions are broadcasted reliably.
- Messages go twice the ring in a two-phase commit: transaction and acknowledgement.
- An acknowledgement is sent by a member if it receives the transaction the second time.
- The last acknowledgement and all unacknow-ledged transactions are stored at a member and resent during recovery from faults.
- Doubled messages are filtered by the receiver using a hop counter in every message.

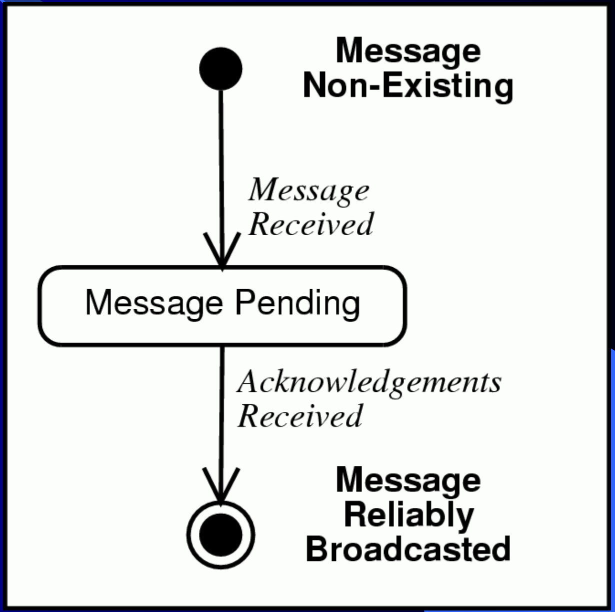
Reliable Broadcast



Phase I



Phase II

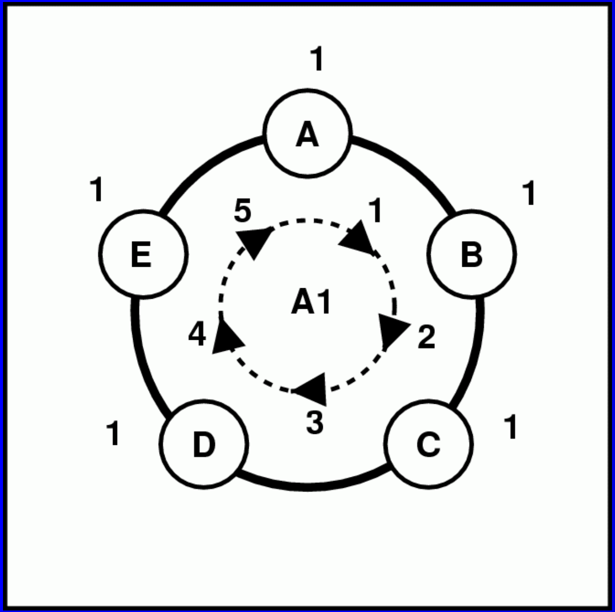


States

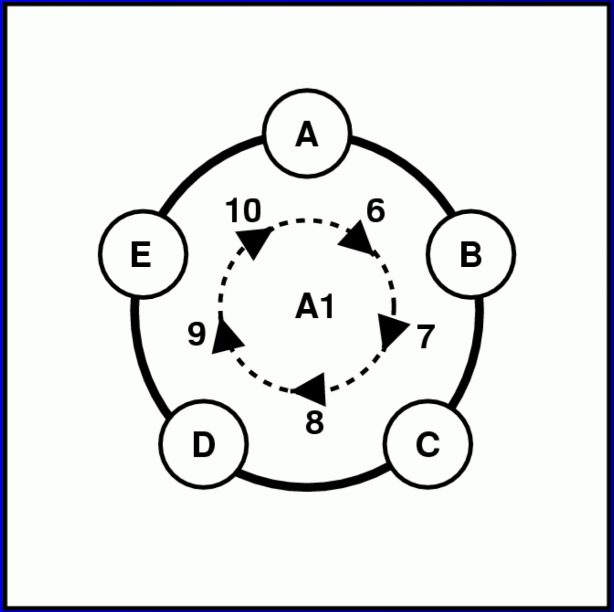
Atomic Broadcast

- Transactions are globally ordered after the reliable broadcast without blocking it.
- They are numbered using origin (member) id and sequence number i.e. no timestamps.
- Sequence numbers are synchronized.
- They are updated with higher received values and increased with every new transaction.
- Simultaneous transactions have equal sequence numbers and are ordered by origin id.
- No denial of service due to a fair share.

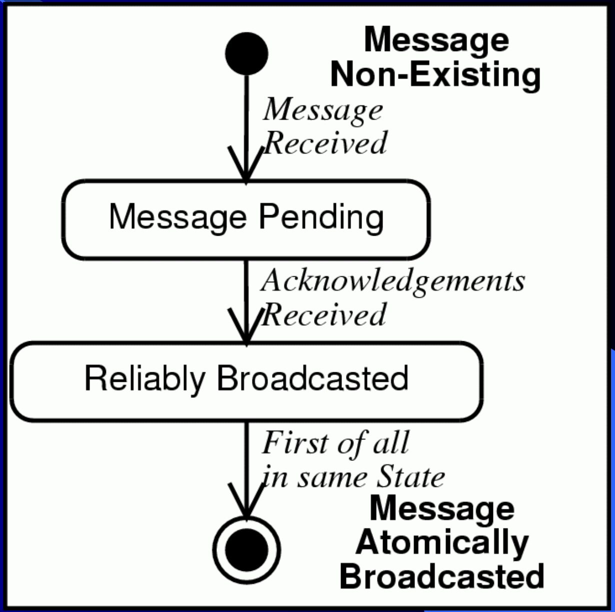
Atomic Broadcast



Phase I



Phase II

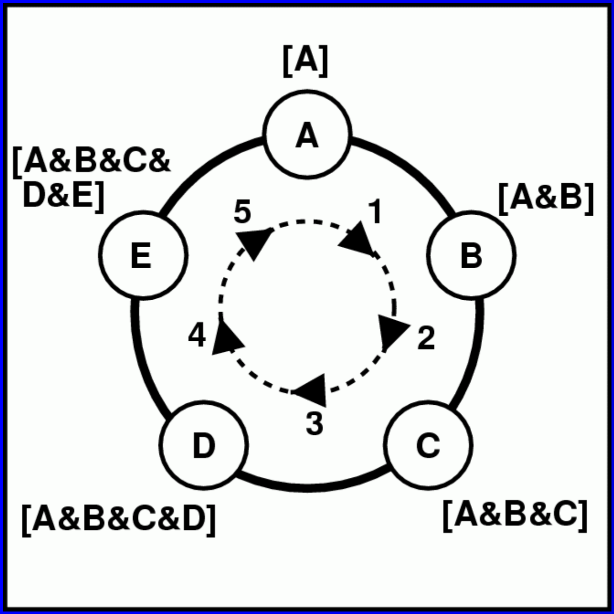

Message Non-Existing
Message Received
Message Pending
Acknowledgements Received
Reliably Broadcasted
First of all in same State
Message Atomically Broadcasted

States

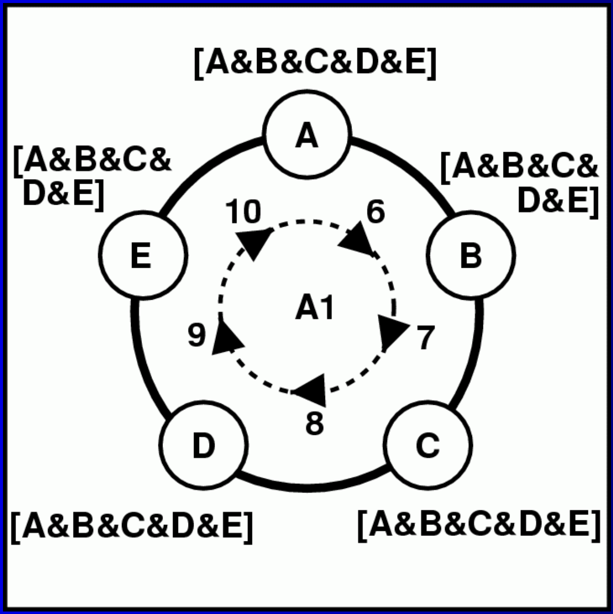
Distributed Agreement

- All members agree on a transaction execution result to commit or reject the transaction.
- Collective communication combines the individual transaction results from all members by interleaving the reliable broadcasts of their results and the reliable broadcast of the final combined result in a three-phase commit:
  - Individual result broadcast and collection.
  - Acknowledgement and final result broadcast.
  - Acknowledgement.

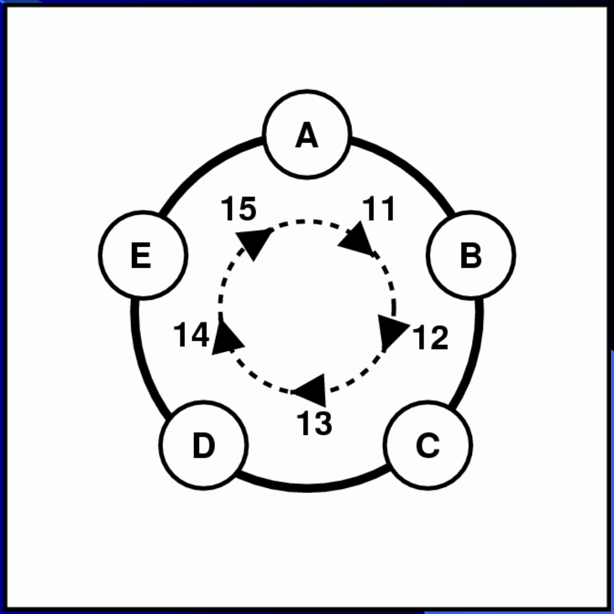
Distributed Agreement



Phase I



Phase II



Phase III

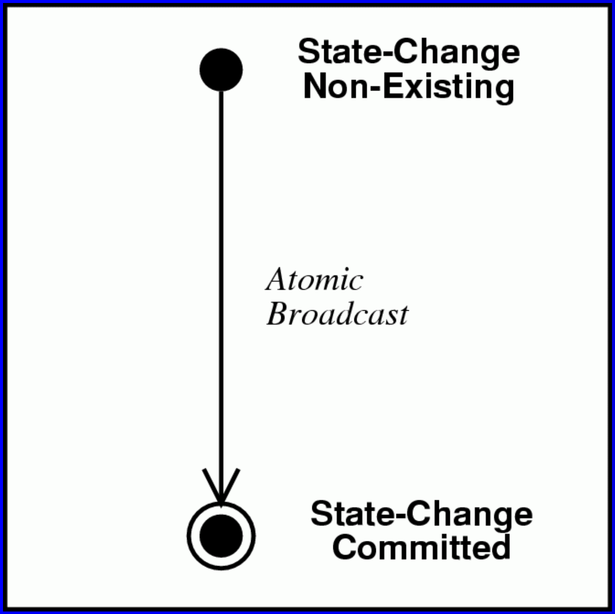
Membership

- All members agree on an initial global state.
  - New members assume the current global state.
- All members maintain the same linear history of state changes.
  - Atomic Broadcast of state changes (two phases).
  - Distributed Agreement on execution results if execution may fail (three phases).
  - State change commit depending on final result if distributed agreement was performed.

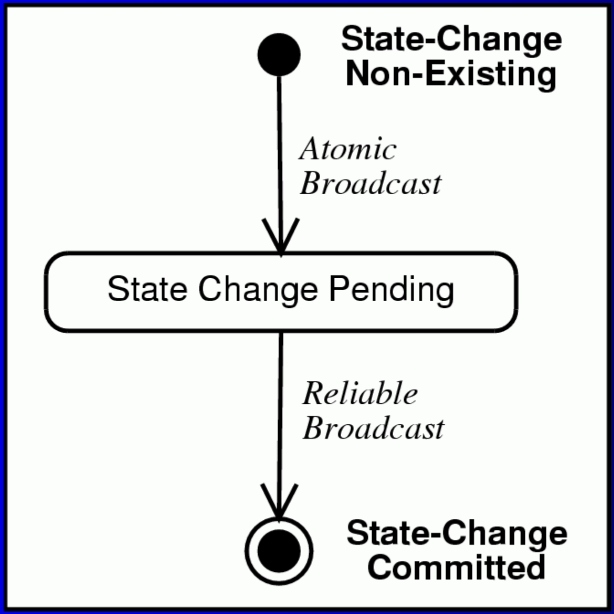
Transaction Types

- Group communication is used depending on the transaction execution target.
- Execution at no member { O(n)=2n }.
  - Atomic broadcast of transaction.
- Execution at one member { O(n)=4n }.
  - Atomic broadcast of transaction.
  - Reliable broadcast of execution result.
- Execution at several members { O(n)=5n }.
  - Atomic broadcast of transaction.
  - Distributed agreement on execution result.

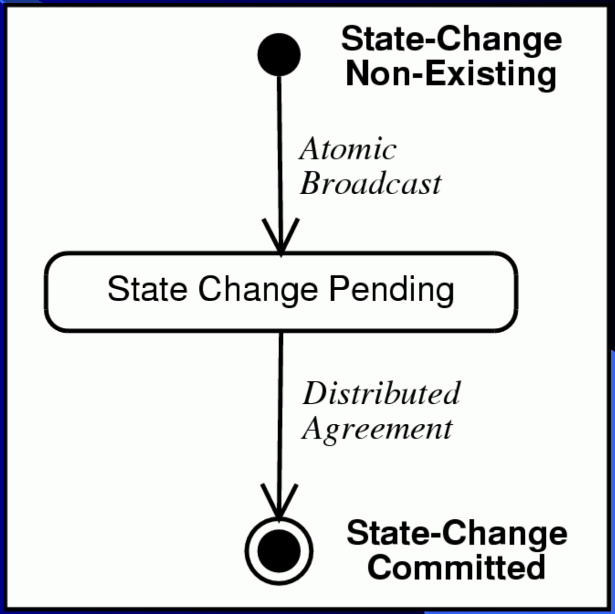
Transaction Types



Type I



Type II



Type III

Conclusions And Future Work

- Fault-tolerant distributed global state control.
- Backbone for system-wide fault-tolerance.
- Scalable group communication { O(n)=5n }.
- No timestamps, no denial of service.
- Unified fault-tolerant application design?
- Distributed parallel plug-in paradigm?
- Splitting and merging virtual machines?
- Application and plug-in development?
  - PVM plug-in, FT-MPI plug-in …

Distributed Peer-to-Peer Control in Harness

Christian Engelmann

Network and Cluster Computing Group,
Computer Science and Mathematics Division,
Oak Ridge National Laboratory, USA

engelmannc@ornl.gov
www.csm.ornl.gov/harness