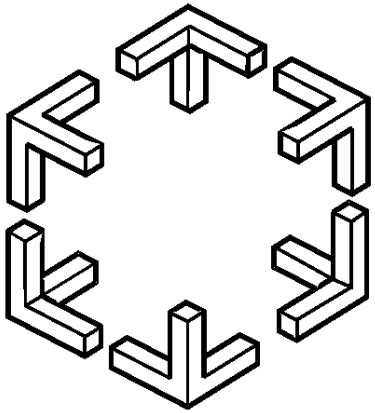
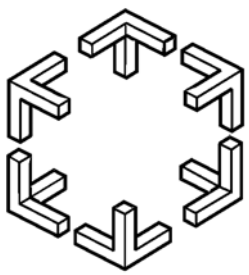
Distributed Computing and Systems
Chalmers university of technology



Adaptive Plausible Clocks

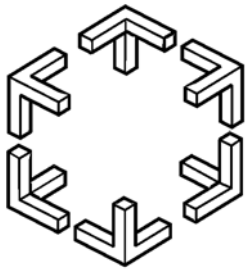
Anders Gidenstam

Marina Papatriantafilou



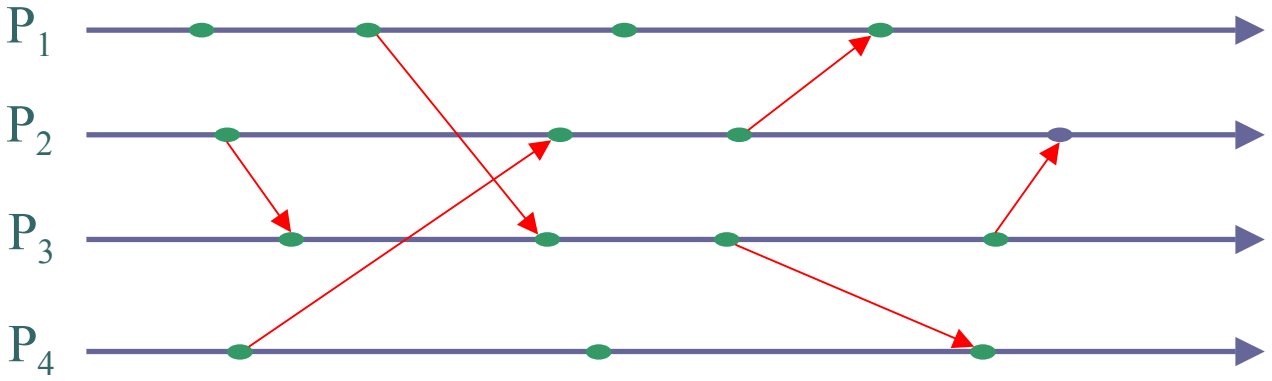
Outline

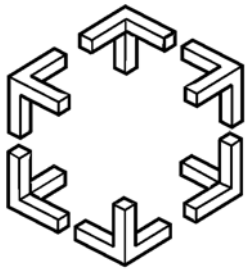
- Background
  - Time, Clocks and event orderings
  - Previous Work
- Contributions
  - Non-uniformly mapped vector (NUREV) clocks
  - How to avoid information loss
  - R-Others NUREV clock
  - MinDiff NUREV clock
  - Experimental results
- Conclusions
- Future work



Time, Clocks and event orderings

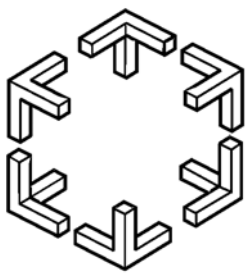
- Distributed system
  - N processes: P_1, P_2, ..., P_N
    - Communicate through messages
    - Asynchronous system
    - No physical clock
  - Events: send/receive message or local step



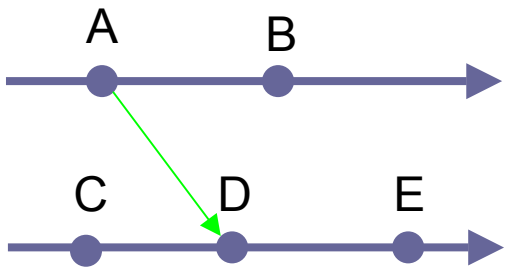


Time, Clocks and event orderings

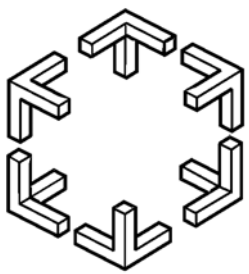
- We want to order the events of an execution
  - Why?
    - As part of some distributed algorithm
      - E.g. Caching of replicated shared objects
      - Causally consistent multicast
    - For monitoring, debugging etc.
  - How?
    - Use a logical clock algorithm (a.k.a time stamping system) to assign timestamps to the events
    - Timestamps
      - Equality and ordering operators: $=_{LC}$, $<_{LC}$
      - Concurrent if incomparable (unorderable)



Event orderings

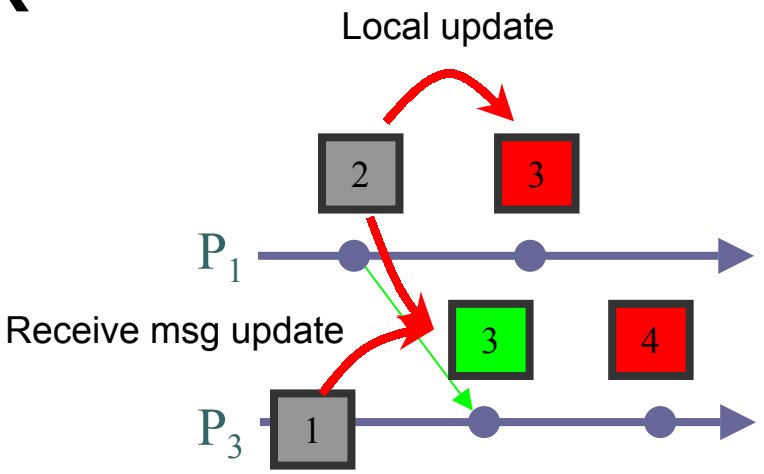


- Total order
  - No concurrency
  - Example: A < C < B < D < E
- Causal order
  - ”happened before” or ”knows about” relation
  - Example: A || C, B || C, B || D, B || E

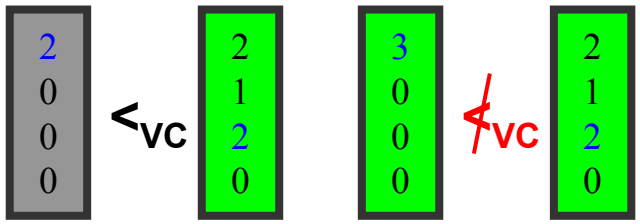


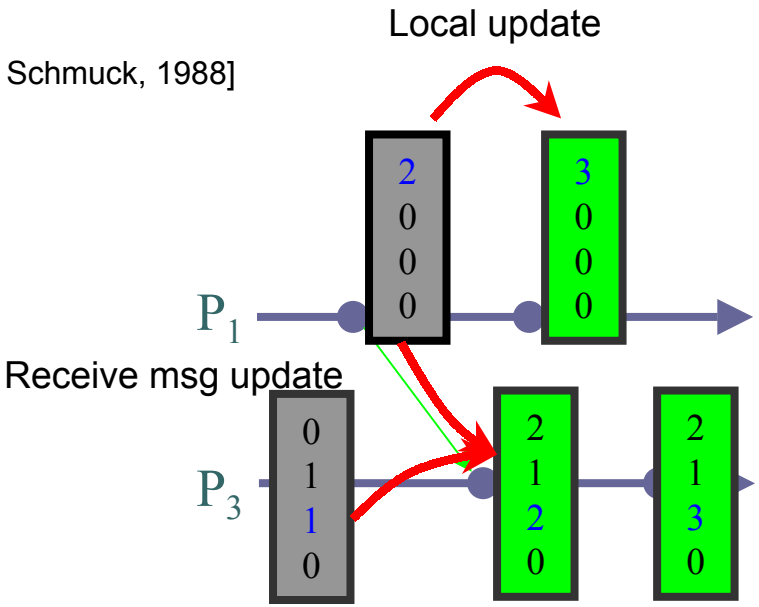
Previous Work

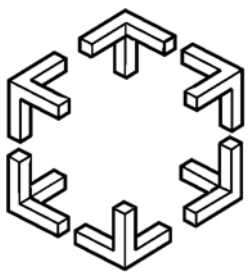
- Lamport Clocks [Lamport 1978]
  - Total order (with tie-breaker)



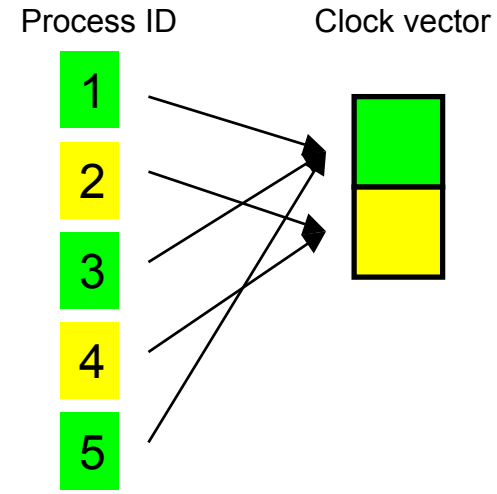
- Vector Clocks [Fidge, 1991, Mattern, 1988, Schmuck, 1988]
  - N clock entries
  - Causal order



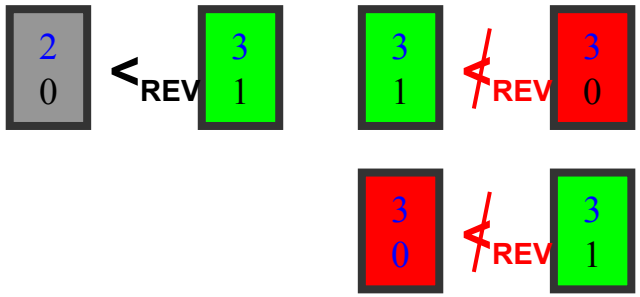


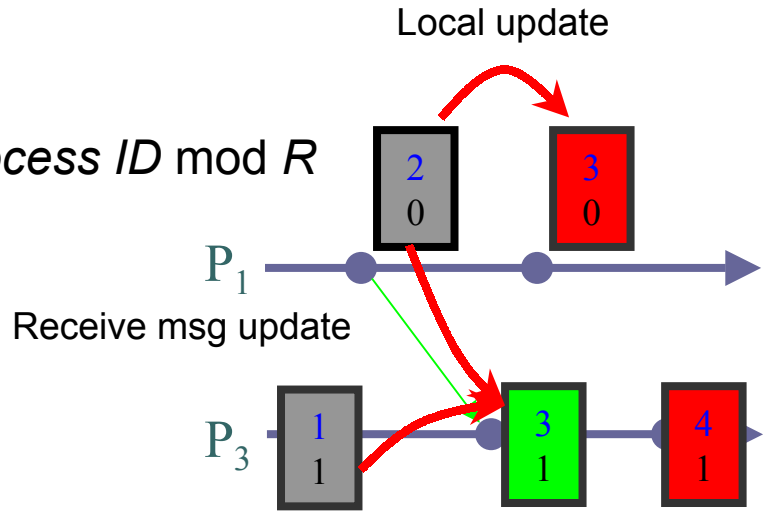


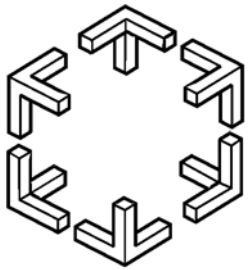
Previous Work



- Plausible Clocks [Torres-Rojas and Ahamad, 1999]
  - Class of logical clocks
    - Orders events consistent with causal order, but may also order concurrent events.
    - Includes: Lamport Clock and Vector clock
    - R-Entry Vector Clock
      - R clock entries
      - Clock vector indexed by *Process ID* mod *R*

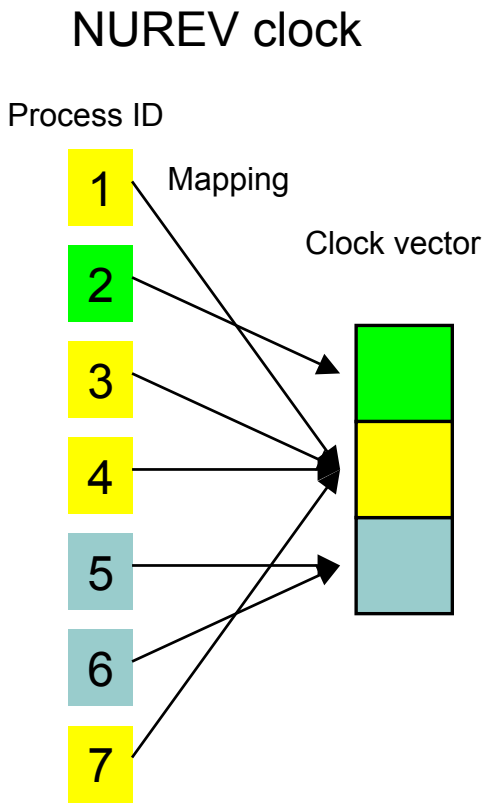






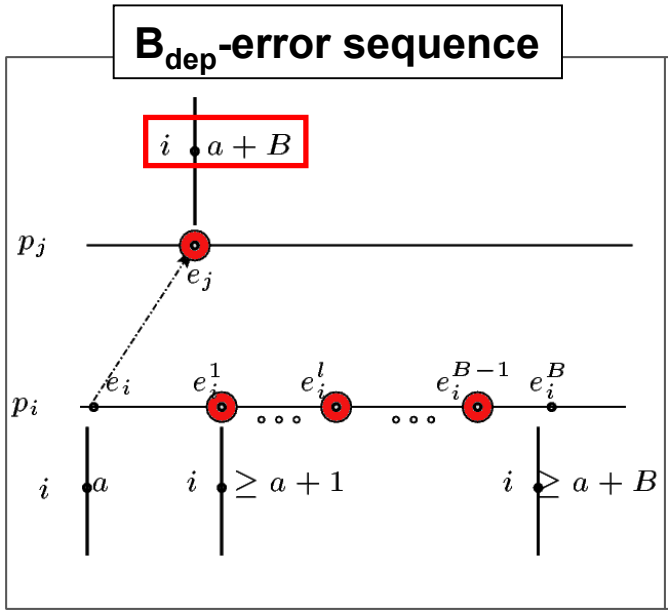
Non-uniformly mapped R-entry vector (NUREV) clocks

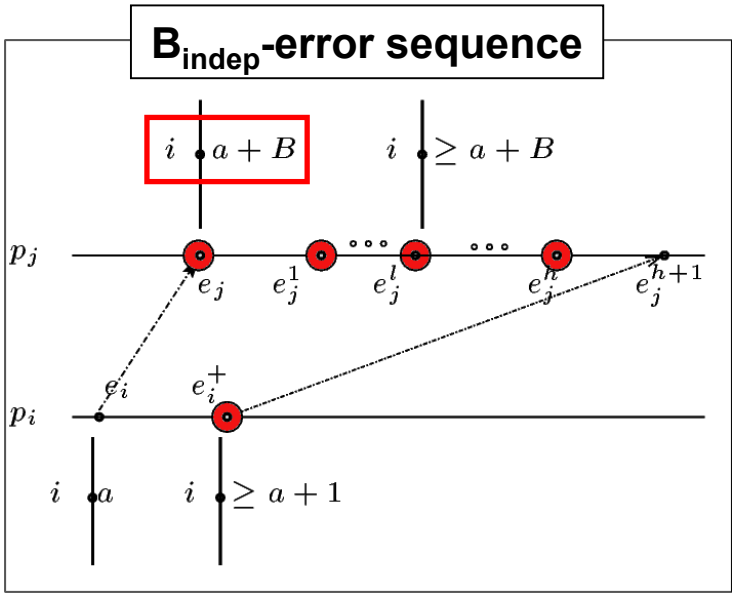
- A generalization of R-entry vector clocks
  - Allows a different mapping between process ID and clock entry in each timestamp
  - Allows (for example) self tuning and adaptation of the timestamping system
  - We have proved that
    All NUREV clocks are plausible clocks.
    - Regardless of mapping function and how it changes.



How to avoid information loss?

- Where is ordering information lost?
  - Inflation of one process key introduces ordering among concurrent events



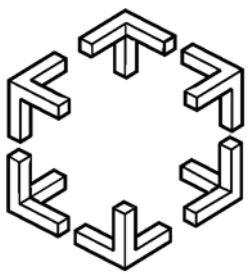


Minimize inflation at updates

- Choose the mapping so that the inflation is small.

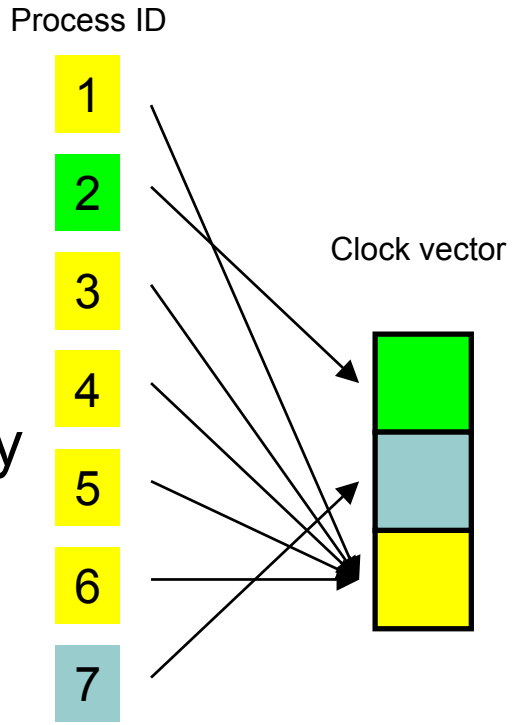
Next-Contact

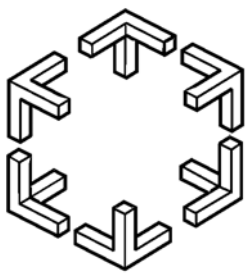
- Avoid inflating the keys of processes you won't hear from in a long time



R-Others Clock (ROV)

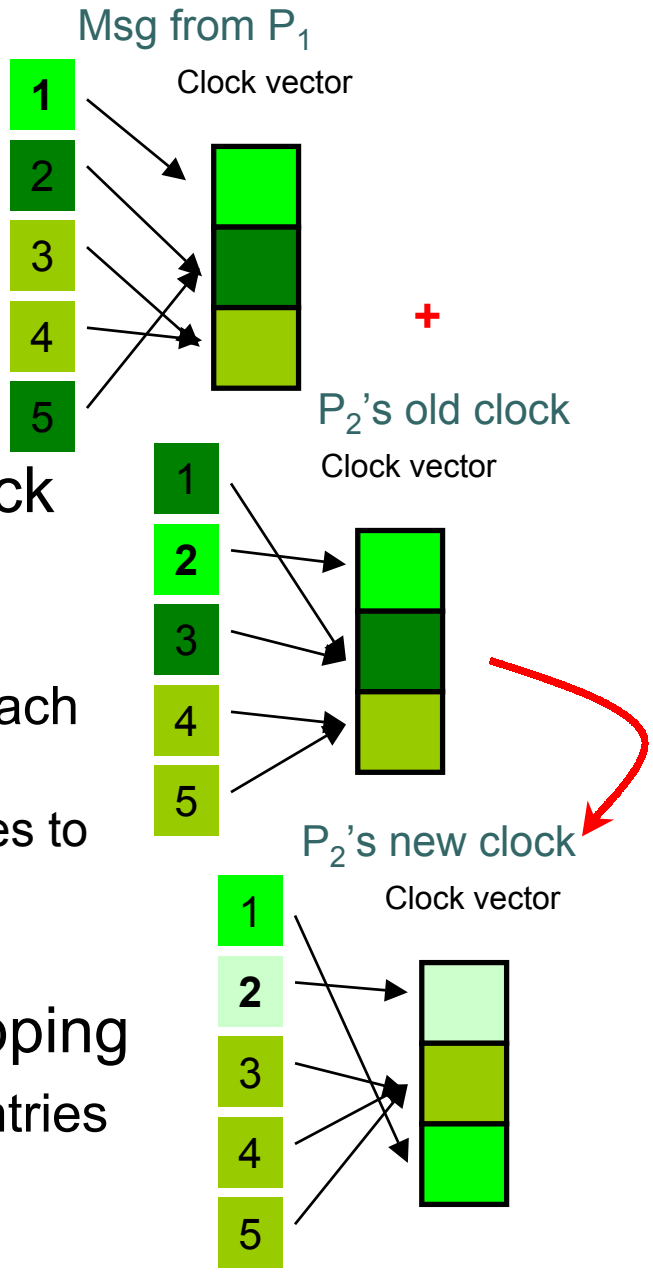
- Idea
  - Preserve recent information
  - Use exclusive entry for
    - own key
    - R-2 other processes' keys
      (Last R-2 communication partners)
  - All other process keys share one entry
- Benefits
  - Constant-size timestamps
  - Agrees well with **Next-Contact**

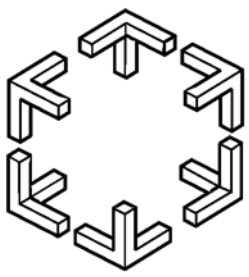




MinDiff clock

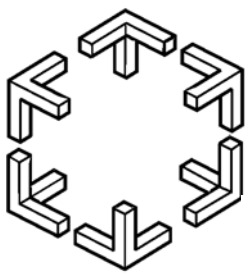
- Idea
  - Minimize the inflation at <u>each</u> clock update
    - Use exclusive entry for own key
    - Select a new mapping function on each receive update
      - Map process keys with similar values to the same entry
  - Timestamps need to include mapping
    - Small for a small number of clock entries



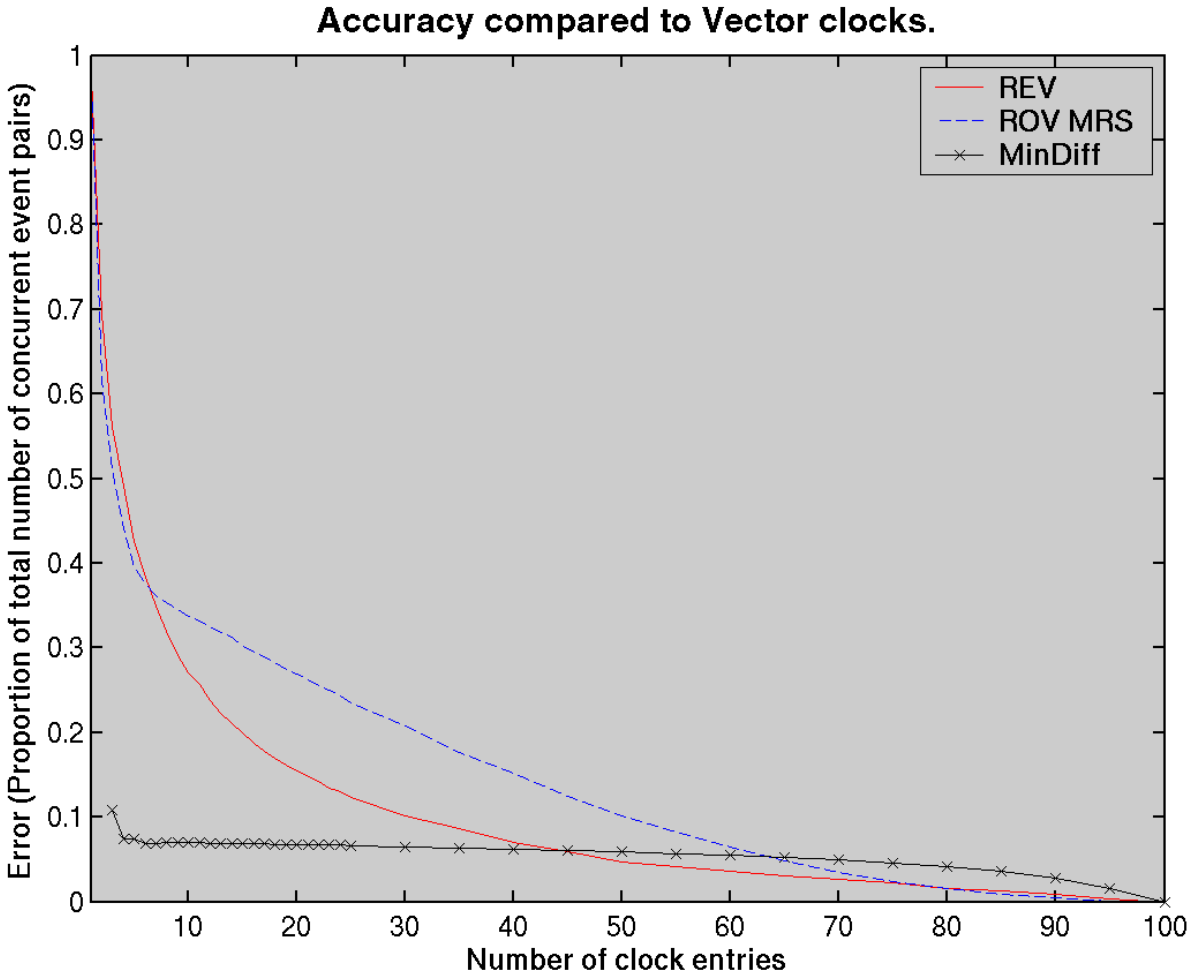


Experiments

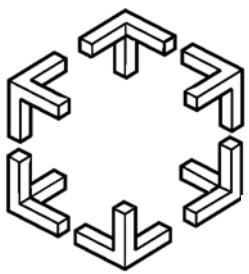
- Simulations
  - Peer-2-Peer systems
  - Client-Server systems
- Performance measure
  - #ordered concurrent event pairs / total #concurrent event pairs



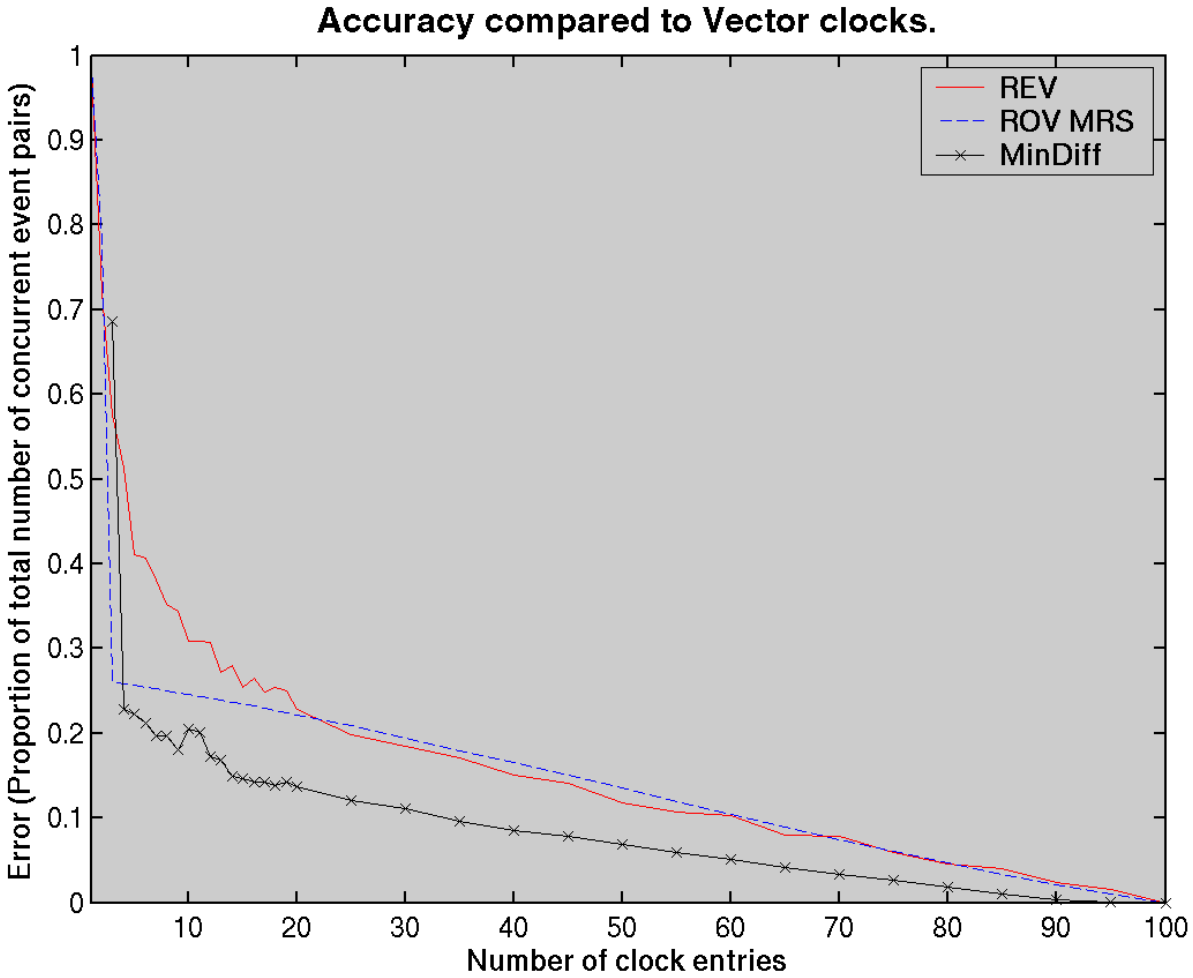
Experimental results



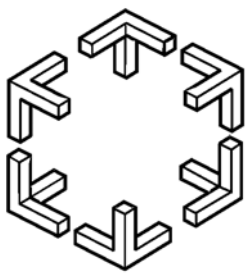
Peer-to-peer system 100 processes



Experimental results

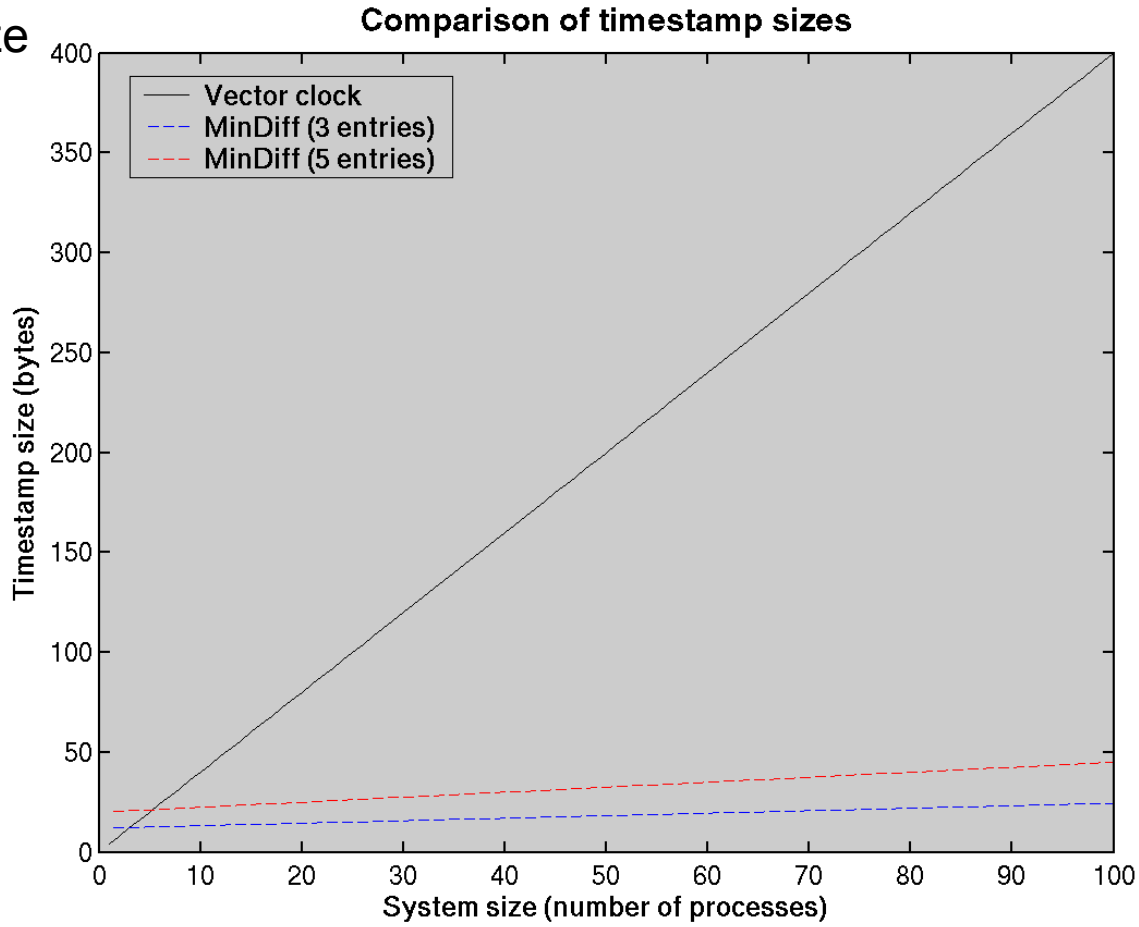


Client-server system 1 server 99 clients

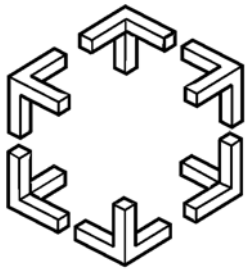


MinDiff timestamp sizes

Timestamp size (byte)

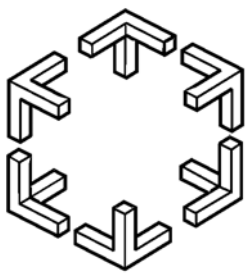


System size (#processes)



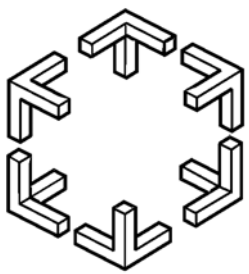
Conclusions

- Non-Uniformly Mapped R-Entries Vector Clocks (NUREV)
  - A general class of logical clocks
  - Guaranteed to be plausible
  - Includes Lamport, Vector and REV clocks
- Analysis of when and how NUREV clocks order concurrent events
- New NUREV clock algorithms
  - MinDiff and R-Others clocks
  - Improved performance at small timestamp sizes



Future Work

- Apply NUREV clocks in a group communication / ordered multicast framework
  - Work in progress
- Further investigation of mapping functions
  - Subsets with constant size representation
  - Approximations
- Bound the size of vector entries



Questions?

- Contact Information:
  - Address:

    Anders Gidenstam /
    Marina Papatriantafilou
    Computing Science
    Chalmers University of Technology
    SE-412 96 Göteborg, Sweden
  - Email:

    <andersg , ptrianta> @ cs.chalmers.se
  - Web:

    http://www.cs.chalmers.se/~dcs/