Replicating Information in a Power Distribution Management System

Author: Wille Wahlroos, Tekla Corp.
Supervisor: Professor Jörg Ott
Contents

• Background
• Goal and methods
• The Distribution Management System
• Databases and replication
• Solution outline
• Conclusion
• Further development
Background

- Ongoing privatization of power distribution utilities.
  - New owners demand increased revenue.
  - Economies of scale to cut costs.
- Authorities exert regulatory pressure.
  - Consumers have to be compensated for outages.
  - Network operators are not allowed to take advantage of their monopoly.
- Higher quality of supply expected.
  - Growing dependency on electricity.
  - Customers' tolerance decreasing.
Background

- Utilities increasingly rely on information technology
  - Streamlining operations, automating tasks.
  - Efficient handling of crisis situations.
  - Reporting, analyzing, optimizing.

- Power Distribution Management Systems (DMS) support the operation of an electricity network.
  - Geographical and/or schematic presentation of the network.
  - Real-time, multi-user, interfaces to other systems.
  - Planning, supervision, calculations, simulations, ...
Goal and methods

- The goal was to outline a possible solution for expanding a DMS system to a multi-site one.
- The solution should be evolutionary, not disruptive.
- The steps taken were as follows:
  - Analyze the current system.
  - Identify the main issues.
  - Read relevant literature and papers.
  - Apply gained insight to outline a reasonable solution.
The Distribution Management System
...to multi-site
Database messages at Writer \((\Sigma = 1.04M)\)
Identified requirements

- Distributed operation.
- Reduced data transfer.
- Prevent loss of data while database is unavailable.
- Exclusive editing of records.
- Possibility to create new records while database is unavailable.
- Platform and technology independence.
Databases and replication
Relational Database

- Based on the relational model by E.F. Codd (1970)
- Stores data using the abstraction of a table.
  - A table's columns have a data type and name.
  - A row is uniquely identified based on the primary key of the table.
  - A foreign key is a reference to a row in another table.
- Queries on the stored data are made using the SQL-language.
  - Manipulation of sets, relational algebra.
ACID properties

- Operations on data are performed as transactions.
  - A transaction is a sequence of requests.
  - The result may depend on the order of execution.
  - One or several requests may fail.
  - Data may be accessed concurrently.
- A transaction should fulfill the ACID properties
  - **Atomicity**: A transaction either completes successfully, or it has no effect at all.
  - **Consistency**: A transaction takes the system from one consistent state to another.
ACID properties

- **Isolation**: A transaction must be performed without interference from other transactions.
- **Durability**: The effects of a successfully completed transaction are saved in permanent storage.

- Atomicity and Isolation are mainly a matter of concurrency control.
- Consistency is mainly defined by an application's business rules. The database management system enforce uniqueness of primary keys etc.
- Concurrency control strategies: serial equivalence, locking, time stamp ordering, optimistic
Replicated Database

• A setup where there are several instances (copies, replicas) of a database. Each contains same set of data.

• Provides:
  – Improved usability through decreased latency (WAN->LAN)
  – Improved availability, fault tolerance

• Changes to data are applied (replicated) to all instances.

• Consistency requirements must be relaxed.
  – Otherwise too much communication overhead.
Replicated Database

• Several approaches to implementation
  – Differences in complexity, throughput, consistency guarantees.

• Primary copy
  – One master, updates propagated to slaves.
  – Clients access only master.

• Update everywhere
  – Clients allowed to operate on any copy.
  – Other copies updated directly by client or by middleware.
Replicated Database

• Propagation of updates
  – Replay operations vs. send only results.
  – In Eager Replication a client has to wait for the updates to be applied at all copies.
  – In Lazy Replication updates are propagated only subsequently. Conflict resolution strategy needed.

• Problems and challenges
  – Distributed deadlock (and detection thereof).
  – High latency, communication overhead.
  – Conflict resolution strategy, system management.
Solution outline
Concurrency control

- Analysis of traffic revealed that 99% of messages are destined to the so called log tables.
  - No concurrency control needed for those.
  - Locking is a feasible solution for other data.
- Lock manager to be implemented at each Node.
  - Global lock server would be a single point of failure.
  - Voting wouldn't address network fragmentation.
- A record needs to know its primary location.
  - Request is forwarded to appropriate Node.
  - Locking is programmer's responsibility.
Network topology

- Complete graph.
  - Estimated number of Nodes ~10. Simple, robust.
  - No re-routing if link breaks.
- Alternative: overlay network.
  - Application layer routing protocol.
  - Optimal use of network resources
  - Non-trivial to implement, difficult to debug.
- Alternative: flooding
  - Lots of data duplication.
Data distribution

• Messages received by the Node from local clients are written to a ring buffer on hard disc.

• Each message is given a sequence number and contains a source identifier.

• A Message can be erased from the buffer once it has been acknowledged by all peers and the local Writer.

• If a message concerns data of another site, it is first sent to that site's Node only. There it is handled as if received from a local client. Thus the master site for a particular data item is guaranteed to be the most up-to-date.
Miscellaneous

- Analysis of traffic revealed that messages to log tables contained mostly redundant data.
  - Extracting duplicate data and coalescing several messages into one reduces burst rate by 84%.
- Id number management similar to locks
  - Preallocation of ranges to sites.
  - File as persistent storage.
Conclusion

- In the end relatively simple solutions were found.
  - Majority of operations have no need for concurrency control.
  - Reliable and robust distribution of data.
- Platform independence can be maintained.
Further development

• Time synchronization
• Monitoring
• Cold stand-by
• Automated Meter Reading