Advanced software technologies for breakthrough applications

Caché Database Mirroring
Lower cost, lower risk alternative to High Availability

Technical Overview
September, 2010
High Availability Trade-offs

The Levels of Availability

- Redundant Systems
- Cold Standby
- Warm Standby
- Automatic Failover
- Continuous Availability

Cost vs. Complexity/Availability
## Service Level Agreements

### Acceptable System Downtime

<table>
<thead>
<tr>
<th>Availability Target %</th>
<th>Downtime – Per Year</th>
<th>Downtime – Per Month</th>
<th>Downtime – Per Week</th>
</tr>
</thead>
<tbody>
<tr>
<td>99.9999% (six 9’s)</td>
<td>31 seconds</td>
<td>2.59 seconds</td>
<td>0.605 seconds</td>
</tr>
<tr>
<td>99.999% (five 9’s)</td>
<td>5.26 minutes</td>
<td>25.9 seconds</td>
<td>6.05 seconds</td>
</tr>
<tr>
<td>99.99% (four 9’s)</td>
<td>52.6 minutes</td>
<td>4.32 minutes</td>
<td>1.01 minutes</td>
</tr>
<tr>
<td>99.9% (three 9’s)</td>
<td>8.67 hours</td>
<td>43.2 minutes</td>
<td>10.1 minutes</td>
</tr>
<tr>
<td>99% (two 9’s)</td>
<td>3.65 days</td>
<td>7.20 hours</td>
<td>1.68 hours</td>
</tr>
<tr>
<td>95%</td>
<td>18.25 days</td>
<td>36 hours</td>
<td>8.4 hours</td>
</tr>
<tr>
<td>90%</td>
<td>36.5 days</td>
<td>72 hours</td>
<td>16.8 hours</td>
</tr>
</tbody>
</table>
The outcome of database failure...

10 am, Thu Aug 26, 2010

• A storage area network (SAN) failed. Redundancy also failed. This hardware failure occurred shortly before 3 p.m. Wednesday.

3 pm, Thu Aug 26, 2010

• While work continued today on repairing the faulty networked storage system, information technology operations continued in a degraded mode impacting 24 state agencies.

11 am, Fri Aug 27, 2010

• Maintenance and repair work proceeded as expected overnight. Applications, servers and the damaged storage system were taken down as planned.

10 pm, Sun Aug 29, 2010

• Progress continues, but work is not yet complete for the three or four agencies that have some of the largest and most complex databases. These databases make the restoration process extremely time consuming.

“According to the manufacturer of the storage system, the events that led to the outage appear to be unprecedented. The manufacturer reports that the system and its underlying technology have an exemplary history of reliability, industry-leading data availability of more than 99.999% and no similar failure in one billion hours of run time.”
Caché Database Mirroring

- Employs Redundant, Less Expensive, Servers & Storage
- Eliminates Shared Services/Resources
- Uses Logical Replication
- Delivers “Warm” Failover/Failback
  - Integrated with Caché’s horizontally scaling Enterprise Cache Protocol (ECP)
- Asynchronous Option for Disaster Recovery
Cost Savings Example

- Eliminate shared storage
- No need for expensive cluster software
- Typical production configurations for web-based healthcare application
- Majority of savings from OS / Software Licensing & Storage
Cost Savings by Configuration Area

- Majority of savings from OS Licensing - HA Failover Cluster licensing vs. regular licensing
- Significant savings within storage – Mirroring doesn’t require expensive SAN configurations
A Mirror is a grouping of 2 Failover Members (Caché Systems); one becomes the Primary, the other becomes the Backup.

Data flows from the Primary to the Backup; Acknowledgements flow from the Backup to the Primary.

Mirror updates are synchronized through the Caché Journaling Process on the Primary.

External systems connect through Mirror Virtual IP.

Enterprise Cache Protocol Application Servers connect directly to running Primary.

*DBC/SQL Bindings Web Clients Direct Connect
Failover Members

- Can be asymmetric endian/platform/OS
- Must both have same character width (both 8-bit or both 16-bit Unicode) and same collation
- Must manually synchronize static configuration (Namespaces, Databases, Mappings, Users, Roles, etc.)
Async Members

- Mirroring provides for a special Async Mirror Member
- By default, mirrored databases on Async are read-only
- Can mount databases as read-write
- Async Members are NOT candidates for automatic failover
- Some sample uses:
  - Disaster Recovery
  - Business Continuity
  - Business Intelligence
  - User Acceptance Testing
Async Member: Sample Architecture

- One Async Member can receive updates from one or more Mirrors
- The Async Member can track one or more Mirrored databases from each of the Mirrors
- Sample use case: centralized data repository (for mining, backups, etc..)
Async Member: Sample Architecture

- One Mirror can update up to six Async Mirror Members simultaneously.
- Geographically dispersed solution - ideal for Disaster Recovery and Business Continuity.
Mirroring vs. Shadowing

- Mirroring: High Availability (w/ automatic failover) and Disaster Recovery
- Shadowing: Disaster Recovery only
- Shadowing will eventually be retired
  - Async Mirror Members can provide full featured async replication functionality
Mirroring: Application Perspective

- Mirroring is per-database
- All mirrored databases are journaled
- Applications view a failover as a server restart
  - All external clients (language bindings, ODBC/JDBC/SQL clients, SMP, Studio, etc.) will reestablish their connections
  - ECP clients will pause while failover completes; on successful failover, work will resume on the new Primary
- Mirrored databases are read-only on the Backup
- Only data in mirrored databases (CACHE.DAT) are synchronized
  - External files need to be manually/externally synchronized
Data Synchronization

- Mirroring uses the **journal write cycle** on the Primary to send journal data to the Backup.

- In Caché, a journal write can be triggered as follows:
  - On an idle system: once every 2 seconds
  - In an ECP environment: when answering “synchronous” requests (e.g., $Increment)
  - TCOMMIT in Synchronous Commit Mode
  - During a Write Daemon cycle
  - If a journal buffer becomes full

- Some configuration parameters control the behavior of synchronization...
Configuration tunable parameters

- Simplify management while providing sufficient flexibility
- Our approach:
  - When should Backup send acknowledgements?
    - **AckMode**: Received (default) or Committed
  - When should Backup be considered falling behind?
    - **QoS Timeout**: default = 2000 msec (2 sec)
  - How should network interruptions be treated?
    - **Agent Contact Required for Takeover**: Yes (default) or No
      - **Note**: if Agent Contact Required is set to No, to prevent against split-brain, you must provide external verification that the other system (existing Primary) is actually down!
Mirror Failover

- Rapid, automatic, unattended failover

- Rules-based; some triggers for failover are:
  - If the Backup doesn’t hear from the Primary every QoS period
  - If the Data Channel is closed by the Primary
  - Manual failover issued by operator or API

- The **AgentContact** tunable parameter determines whether automatic failover is successful
  - Default: **Yes** – if other agent is down, no failover
  - If Agent Contact Required = No, **user-defined ^ZMIRROR** routine must be present to answer: Is other system really down?
Additional Information

• **2010.2 Field Test** contains a good starting point for Mirroring.

• Please evaluate Mirroring and provide feedback:
  
  – **Functionality**: does our implementation of Mirroring provide you with the functionality that you need in an automatic-failover high availability solution?
  
  – **Failure Scenarios**: have we captured all possible failure scenarios for your application (which result in an automatic failover)?
  
  – **Operational Considerations**: have we covered all of the operational questions and considerations in our implementation and documentation?
  
  – **Architectural Considerations**: how will introducing mirroring into your install base impact the system architecture at your customer sites?

• **Pricing**: Mirroring will require a Multi Server license (just like ECP)

• Feedback:
  
  – Please route all issues through the **WRC or your account team**
Mirroring – Summary

- Economical high availability solution with automatic failover for database systems
  - Can reduce costs drastically while increasing availability!
- Redundant components minimize shared-resource related risks
- Logical data replication minimizes risks of carry-forward physical corruption
- Provides a solution for both planned and unplanned downtime
- Provides Business Continuity benefits via a geographically-dispersed Disaster Recovery configuration
- Provides Business Intelligence and reporting benefits via a centralized Enterprise Data Warehouse configuration