RESONATE®

Newsletter 2004 Q4

Product News _____

Success Story _____

Real-time Database Synchronization for Business Continuance _____

Scheduled or Asynchronous Replication _____

On-demand or Synchronized Replication _____

Real-time Distributed Data Synchronization _____

Dynamic Database Resources Provisioning _____

Introduction _____

Content-based Routing _____

Improving Windows Terminal Services _____

Overview _____

Extending the Session Directory _____

Product News

Central Dispatch 4.1 allows customers to manage clusters through a mix environment of 3.x and 4.x agents and administrate Central Dispatch site through either CDMaster or Dispatch Manager.

Central Dispatch 4.1.1 supports Red Hat Linux ES & AS 3.0.

Real-time Database Synchronization for Business Continuance

Disaster can strike at the least opportune time, in many forms, such as Natural disasters, electrical outages, Security breach or Terrorist attacks. The growing amount of corporate data forces IT managers to deal with two issues:

- Geographically Distributed data to ensure business continuity
- Reduce total cost of ownership

Traditional data replications mechanisms do not safeguard data stored neither between scheduled duplication of content nor during the replication process itself.

Scheduled or Asynchronous Replication

Asynchronous database content replication consists of duplicating data from a master database to several databases at scheduled interval of time. The approach relies on the distribution and execution of transaction logs across multiple servers. There is an obvious lag time or time delay caused by the latency associated with the connectivity through a wide area.

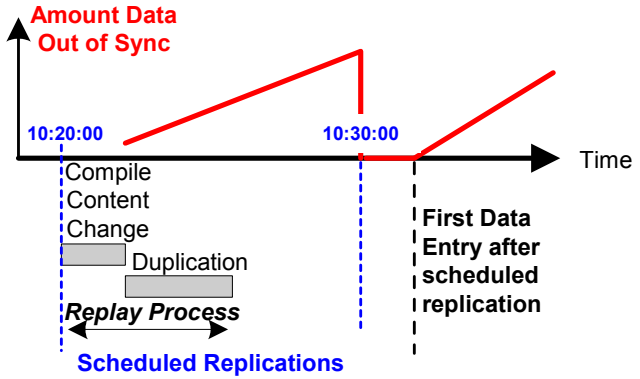

Fig 1. Scheduled Replication Procedure

Prior to the duplication, the replication mechanism marks the state of the

CDConsole 4.1 adds support for Windows 2003, Solaris 9 64 bit & Linux.

Global Dispatch 3.3.1 enhanced to support Red Hat Linux AS & ES 3.0.

Success Story

Redundancy for Wireless Carrier - *Upoc Networks*

The Challenge:

Finding a product that allowed highly flexible load balancing & availability in a high volume-messaging environment. We needed to be able to scale to meet ever increasing messaging needs and provide redundant messaging and queuing connections to the top tier wireless carriers such as AT&T, T-Mobile, Alltel & Verizon, as well as employ standard HTTP load balancing for our web/wap front end. Within our messaging core we also needed a solution that could handle a high number of transactions and remain very scaleable. Deploying hardware-based load balancers would not have been a cost effective solution on a per-carrier basis.

The expansion of our messaging capabilities relied heavily on being able to cluster and distribute carrier-facing servers with ease. Current hardware based solutions would not be able to determine system & network loads in a manner intelligent enough to manner – something that resided on the servers themselves would have to be found. Although we had great success with the Cisco Local Director product, it would never meet our requirements as our business rapidly grew.

master database and starts to identify the SQL requests that have been executed against the master database since the last duplication. Those WRITE requests are actually replayed against all the slave nodes. So even in the case replication are scheduled at a very high frequency, the content of the slave nodes will not be up to date for at least the duration of the replication process.

On-demand or Synchronized Replication

One alternative is to initiate very small incremental duplication of content to slave database nodes, as soon as a record is added or updated in the master database. This design distribute the overhead of the replication to every WRITE request, reducing significantly the probability the master database is unable to be accessed during the replication. However, the mechanism becomes impractical for replication across WAN.

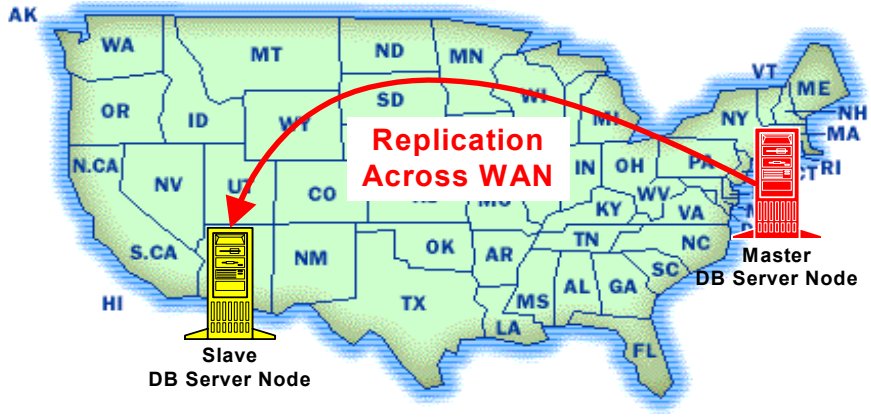

Fig 2. Synchronized replication

The latency of connectivity between site geographically including overhead of encrypting and decrypting data, breaks the model of synchronized replication (i.e Scenario B).

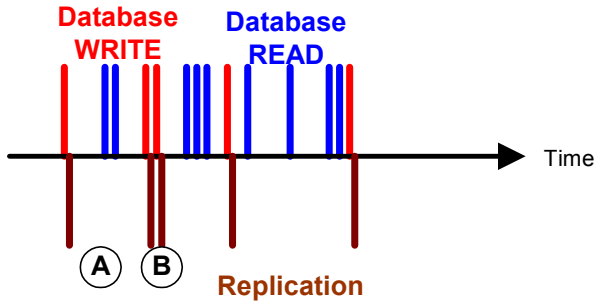

Fig 3. Timing in synchronized replication across WAN

The Success

In 2002 we deployed Central Dispatch into our production environment and immediately saw the results. Since we were now able to balance directly on the hosts that were running specific applications, clustering and spreading the load across multiple servers gave us the ability to re-route traffic and services in real-time, without so much as a blip in production traffic & transactions. The ability to distribute high volume messaging loads across multiple machines, and very simply increase the size of the cluster allowed us so scale without worrying about architecture. The highly available nature of CD is very reassuring, and the overall maintenance / upkeep of the product is very minimal. Basically, there is no babysitting involved! In an environment processing tens of millions of transactions a day the last thing my team or I want to worry about is the stability of our load balancing. CD definitely gives us peace of mind.

The learning curve involved with installation, configuration and usage is minimal – but the power that you are given is immense. On the fly selection of hosts within a load balancing group, selection of multiple fail over hosts, the ability to enforce site-wide or group wide load balancing policies make CD a key part of our messaging infrastructure. I can honestly say it is one of the best pieces of software we have purchased – And I am always eager to talk up the product and it's merits to clients, vendors and industry peers alike. Bottom line, we are able to meet the messaging demands of every major wireless carrier as a result of CD deployment.

Real-time Distributed Data Synchronization

The fundamental limitation of the traditional replication model is based on the assumption that one of the data center has to be a master data making any content synchronization sequential. The solution is obviously to make the database synchronization concurrent across data center geographically distributed and eliminate the need to elect one of those centers as a Master.

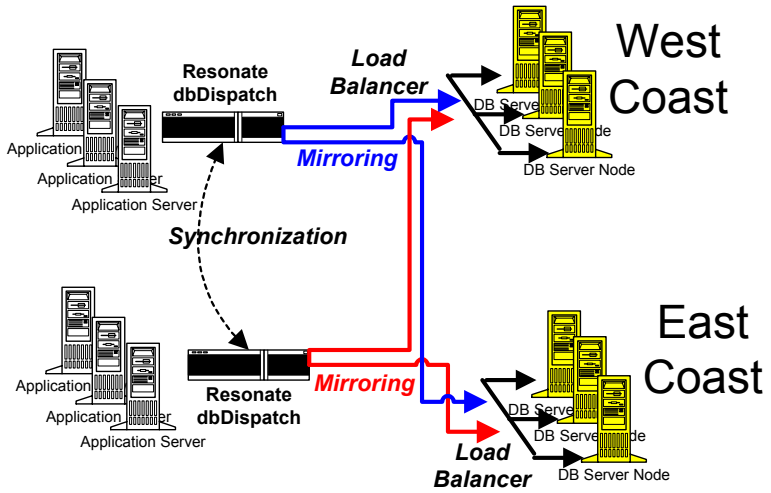

Fig 4. Real time synchronization of data centers using dbDispath

For more information on **dbDispatch™** or contact Resonate at **408.548.5639** or **800.681.6500** Sales@resonate.com

Dynamic Database Resources Provisioning

Introduction

There are many reasons to segregate data across multiple node of a database cluster.

- **Maintenance**: ability to set aside one database node for scheduled maintenance without affecting the overall performance of the cluster
- **Performance**: distribute the content of one or several large tables across multiple servers
- **Security**: content sensitive tables or subset of their content is assigned a separated node

Content-based Routing

In order to realize the benefits stated above, the cluster should be able to parse the content of a SQL WRITE request and route dynamically the request according to its content. A typical WRITE request

> UPDATE *employees* SET *city* = '*Sunnyvale*'
WHERE *name* = '*Lou*'>

INSERT INTO *employees*
VALUES {'Jim', 'Doe', '41', 'San Jose'}

Improving Windows Terminal Services

Overview

Windows Terminal Services relies on the Session Directory Services as a repository of the information regarding current and past sessions (disconnected), user privileges and group policies.

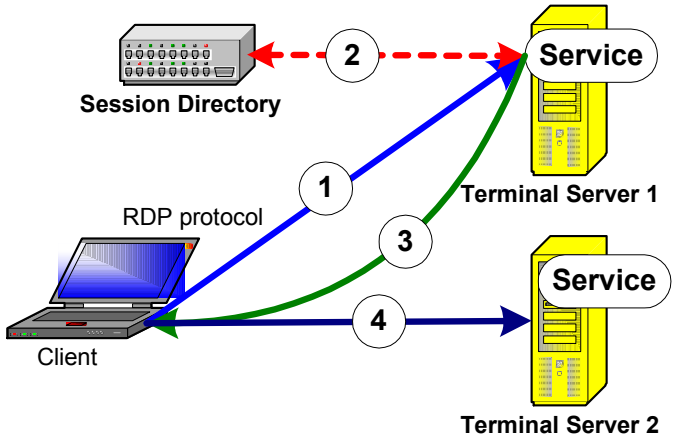

Fig 1 Connections Management of Terminal Services

Although the Session Directory Service distributes sessions across multiple terminal servers, it does not provide however, load balancing or DNS capabilities.

1. The client connect to Terminal server 1
2. Terminal server 1 consults the session directory to get the most appropriate server, 2.
3. Terminal server 1, communicate back the information to the client, in the case there is already a persistent session on Server 2.
4. The client connects to Terminal server 2

Extending the Session Directory

The Session Directory Session relies on a load balancing to maintain existing session by making the connection between a client and terminal server persistent.

The option is called Windows Terminal Service Persistency (WTSP). To enable WTS persistence in the default mode, you must configure Session Directory on each Terminal Server that is listed in your load balancing pool.

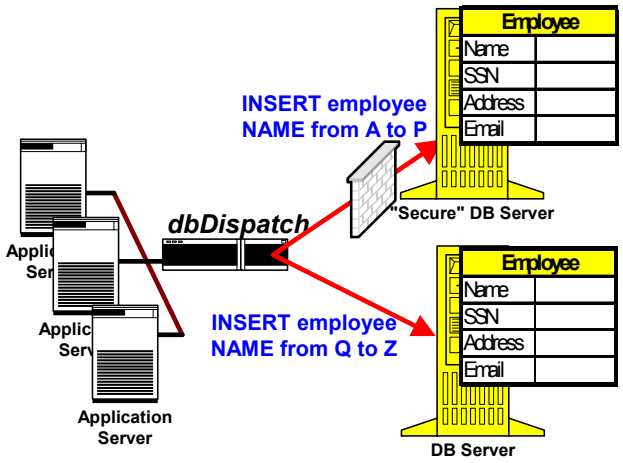

Fig 1 dbDispatch Dynamic Content Routing

For more information contact Resonate at
408.548.5639 or sales@resonate.com

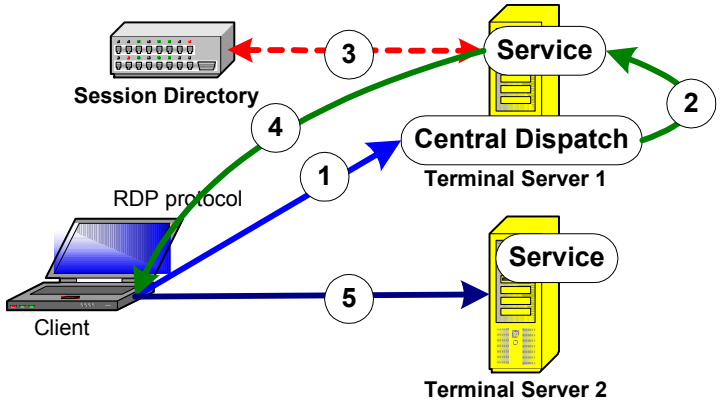

Fig 2 Connection Management using Central Dispatch

1. The client connect to Central Dispatch 4.1
2. Central Dispatch 4.1 select the “least busy” terminal server 1
3. Terminal server 1 consults the session directory to get the most appropriate server 2
4. Terminal server 1, communicate back the information to the client, in the case there is already a persistent session on Server 2.
5. The client connects to Terminal server r