Overview of MAGONIA API Specifications (For Distributed Processing Base)

February 19, 2015
Nippon Telegraph and Telephone Corporation
MAGONIA Overview

- MAGONIA - a service platform resilient to disasters and changes in traffic
- For service application, we are publishing API and functions for the Distributed Processing Base, part of the MAGONIA middleware

Distributed resource pool

Resource pool distributed among multiple data centers

Resource management system

Physical resource management
Virtual resource generation/deletion/provision to users

Middleware (MW)

Middleware provides high reliability and automatic scaling function to applications

Distributed Processing Base etc

Distributed Processing Base
What is a Distributed Processing Base?

A Distributed Processing Base is...

A highly reliable, highly available and automatically scalable framework to distribute processing load etc. provided by middleware.

End users

Distributed processing (load balancing) handles large-scale traffic processing

Better reliability with data replication

Dynamic system scaling
Distributed Processing Base functions

- The Distributed Processing Base is a highly reliable scale-out system that features message dispatching, data management, executable commands and system configuration management functions etc.
- Enables application development focused on service logic, and enables efficient development

### Distributed Processing Base middleware

#### Message dispatching
Dispatches messages that are dependent on service specifications such as HTTP/SIP sessions etc
- Message dispatcher

#### Execution control
Calls applications and performs event control when messages are received.
- Message processor
- Event distributor
- Event distributor

#### Data management
Saves service data and replicates saved data
- Data container
- Original data manager
- Replica data manager
- Lock manager

#### System configuration management
Defines and changes distributed system structures (clusters) and manages VMs comprising clusters
- Generic member controller
- Clustering manager
- Data relocator
- Coordinator controller
- Mutual watch keeper

### Utilities
Common utilities
- Common parts
- Communication control
- Log manager

---

Distributed Processing Base functions and internal function blocks

---

Legend
- Related functions
## Overview of Distributed Processing Base function blocks

<table>
<thead>
<tr>
<th>Function block name</th>
<th>Function block abbreviation</th>
<th>Description of main function</th>
</tr>
</thead>
<tbody>
<tr>
<td>common parts</td>
<td>CMNP</td>
<td>Common utility for thread management, buffer management etc</td>
</tr>
<tr>
<td>communication controller</td>
<td>COMC</td>
<td>Common processing of TCP/UDP layer communication control</td>
</tr>
<tr>
<td>log manager</td>
<td>LOGM</td>
<td>Common processing of log output</td>
</tr>
<tr>
<td>data container</td>
<td>DTCT</td>
<td>Creates, updates, get and deletes data</td>
</tr>
<tr>
<td>original data manager</td>
<td>ODTM</td>
<td>Manages original data</td>
</tr>
<tr>
<td>replica data manager</td>
<td>RDTM</td>
<td>Manages replica data</td>
</tr>
<tr>
<td>lock manager</td>
<td>LCKM</td>
<td>Manages data lock</td>
</tr>
<tr>
<td>generic member controller</td>
<td>GMBC</td>
<td>Processing in each VM for initialize/shutdown processing, and changes in cluster structure</td>
</tr>
<tr>
<td>clustering manager</td>
<td>CLSM</td>
<td>Manages cluster structure data, and distributes to each VM</td>
</tr>
<tr>
<td>coordinator controller</td>
<td>CRDC</td>
<td>Decides the coordinator (that which decides the cluster structure), decides the cluster structure, and controls when cluster structure is changed (notifies each VM etc)</td>
</tr>
<tr>
<td>data relocator</td>
<td>DTRL</td>
<td>Controls and executes data transfer and redundancy when VMs are added or deleted due to malfunction or changes in traffic</td>
</tr>
<tr>
<td>mutual watch keeper</td>
<td>MWTC</td>
<td>Monitors other VMs for malfunction, and notifies detected malfunctions</td>
</tr>
<tr>
<td>message dispatcher</td>
<td>MSDP</td>
<td>Calls application sorting logic, and judges destination for sorting</td>
</tr>
<tr>
<td>message processor</td>
<td>MSPR</td>
<td>Calls applications, locks data, and executes time-limited processing</td>
</tr>
<tr>
<td>event distributor</td>
<td>EVDT</td>
<td>Sends and receives messages, triggers events when sending and receiving message</td>
</tr>
</tbody>
</table>
Distributed Processing Base processing image

- Distributed processing base provides implementation frameworks for application
- Applications such as message dispatch key extraction and service logic etc are implemented in the specified methods, while distributed processing base calls those methods.

Main APL implementation point

<table>
<thead>
<tr>
<th>Distributed processing base API</th>
<th>APL mount processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dispatch key extraction</td>
<td>Implements dispatch key extraction processing called by distributed processing base when messages are received</td>
</tr>
<tr>
<td>Service logic</td>
<td>Implements service logic (data update, message processing) called by distributed processing base after dispatch destination is determined.</td>
</tr>
<tr>
<td>Initialize/quit processing</td>
<td>Implements initialize/quit processing called by distributed processing base when processes launch or finish.</td>
</tr>
<tr>
<td>Time-limited launch processing</td>
<td>Implements periodic or time-limited launch processing called by distributed processing base with the timing specified by APL</td>
</tr>
<tr>
<td>Malfunction processing</td>
<td>Implements application processing called by distributed processing base when process health check is NG</td>
</tr>
</tbody>
</table>