NEC hosts a cloud-based common software development environment called the "Software Factory." With standards tailored to the needs of each project, the Software Factory serves as a platform that centrally manages the development environments for servers, development tools, assets of the past, development management techniques and templates.

The goal of the Software Factory is to improve the QCD (Quality / Cost / Delivery) of NEC solutions by standardizing software development, and bring new applications to customers faster. With this streamlined development environment, NEC hopes to optimize costs by centralizing development, accelerating the development process, and automatically promoting the validation tool during the night.

Challenges

As the only companywide platform for all development activities, the Software Factory faces significant pressure to fulfill demanding and precise requirements. It is paramount for the Software Factory to operate non-stop 24/7/365, which poses significant challenges to the support team.

Beyond daily management requirements and planned upgrades that affect uptime, there is also unplanned downtime to consider. Keeping a cloud software development environment operating around the clock is a major challenge, but the Software Factory also must provide the required development environments without delay when requested.

These requirements call for high availability and high performance from a leading data center application. Virtualization technologies enable optimal server resource utilization and high availability, each virtual server continues to support development during idle maintenance periods. Unfortunately, the network still requires manual intervention for changes in configuration.

From the viewpoint of minimizing load balance and outage time, the Software Factory operates two data centers: one in East Japan and another in West Japan. Allocated to both data centers, the Software Factory supports a virtual server image for backup purposes. These backup server images are sent back and forth between the data centers.

In the case of a disaster, the most recent image is preserved at the "safe" location. This type of flexible operation requires an infrastructure that can dynamically change the network configuration in response to expansion and virtual server changes. "The problem is that it takes considerable effort to change the network when virtual servers are added, changed or removed," said Katsutoshi Nihei, Software Process Innovation and Standardization Division of NEC Corporation. "When a virtual machine is moved, changes in router configurations and addresses of the virtual machine need to happen. Such work, if done manually, is time consuming and also runs the risk of setup errors."
Network configurations are becoming much more complex relative to the increasing intensity of the system. It is difficult to get an overview of where and how the communication is flowing, which makes it difficult to manage network operations,” added Mr. Nihei. “Network outages can be devastating to the productivity of many departments. Once an error occurs, it may cause prolonged outages to determine the cause and recover it.”

Solution

With the widespread adoption of virtualization technology, there is increasing demand for a network that can change network configurations dynamically, and cope with expansion and virtual machine changes quickly. In response to these market drivers, NEC developed the UNIVERGE ProgrammableFlow (PF) Series to maximize the performance and management of the network in a virtual environment. A ground-breaking network solution based on OpenFlow™ network control technology, ProgrammableFlow delivers a radically simplified and open infrastructure for data center and cloud networks.

ProgrammableFlow completely separates the transmission control function and the packet forwarding function of a traditional network switch. ProgrammableFlow controllers offer integrated control of the network, while ProgrammableFlow switches handle packet forwarding. This approach enables flexible architectural modifications and additions, greater levels of automation, as well as efficient operations management. This architecture simplifies the network, allows detailed and dynamic control of communication channels and configuration changes, enables efficient management by GUI, and realizes automatic tracking to the network reflecting expansions and changes of the virtual server.

Already recognized for delivering virtualization, visualization and simplification of the network, ProgrammableFlow received the “Best of Show Award” at Interop Las Vegas 2011 (Infrastructure Category) and Interop Tokyo 2011 (Solutions and Services Category - Network Solutions) to name a few.

“Ideally suited for regional disaster recovery or rapid provisioning, ProgrammableFlow delivers a network that can adapt to expansion and changes of virtual servers quickly and easily,” Mr. Nihei added, “We only truly realized the full benefit of ProgrammableFlow’s advanced performance after actually using it.”

With ProgrammableFlow, the backup virtual server can just be rebooted because it automatically recognizes the virtual server and initiates communication. “Not only did it reduce the workload and dramatically improve the speed of restoration, but it also suppressed the risk of service outages,” said Mr. Nihei.

The features of ProgrammableFlow allow development environments to be deployed rapidly. For example, in a software development environment, there must be three

Results

“ProgrammableFlow now delivers a number of benefits to the Software Factory, but I must admit that I was skeptical prior to its introduction,” said Mr. Nihei. “There was already a stable network running at the Software Factory, so replacing it was unsettling for me.”

There was little reason to be concerned, positive results surfaced from early tests. “Since it was possible to gradually deploy ProgrammableFlow into our environment, we deployed ProgrammableFlow in conjunction with the existing network equipment little by little.” The existing virtual servers in the underlying network were moved slowly under ProgrammableFlow until the entire network was switched to ProgrammableFlow.

“As soon as we deployed PF, our uncertainty dissolved immediately,” confessed Mr. Nihei. “I realized its advanced performance after using it. I recall my surprise when the flow of data had been switched to a new pathway on the GUI data after the virtual server had been moved to another data center, and ProgrammableFlow recognized it automatically.”

The adoption of ProgrammableFlow has delivered impressive results for the Software Factory. The infrastructure that probably would have taken two months to deploy, was completed in only 10 days with the automation functions of ProgrammableFlow. As it became possible to switch easily between the virtual servers in East Japan’s and West Japan’s data centers, it also reduced capital investments for hardware and software resources. Despite East Japan’s data center coming close to its computational limit, it is now able to leverage several of West Japan’s virtual machines.

There is no longer the need to prepare unnecessary duplicate resources at both data centers, since they are now able to share resources.

ProgrammableFlow is also able to visualize the entire network. The GUI can clearly display the physical and virtual network topology and communication flow. “Currently, we are centrally managing both East Japan’s and West Japan’s network from our headquarters in Tokyo. ProgrammableFlow exceeded our expectations when we were setting the configuration of individual devices on the command line,” says Nihei. “Initially, we managed the network with an Excel file and made the configuration changes onsite to prepare for mistakes being made, but ProgrammableFlow has solved all of those challenges.”

The server and network management, which were managed separately in the past, will be integrated as ProgrammableFlow expands as a network solution. “We appreciate the progressive approach of cloud platform management,” said Nihei. “Customers considering deploying ProgrammableFlow don’t need to make a drastic change. Instead they benefit from a gradual migration of the server environment.”