SYSTEMS AND METHODS FOR AUTOMATIC PROVISIONING OF A USER DESIGNED VIRTUAL PRIVATE DATA CENTER IN A MULTI-TENANT SYSTEM

Inventor: John Yung

BACKGROUND

[01] Enterprise customers, such as businesses, non-profits, governments, etc., continue to rely on information technology and network architectures as their communication and productivity infrastructure. With the continued increase in public network bandwidth and availability, these IT and network services may be provided by outside companies, which leverage shared resources and expertise to provide a greater cost-savings to subscribing customers.

[02] Several component provisioning systems are currently in use, even some with direct customer interaction (e.g., shared storage space provisioning). Additionally, there exists IT tools such as Configuration Management Databases (CMDB) that maintain device specifications and interactions of multi-user systems. However, provisioning an entire virtual data center remains a technical and complicated task, unsuitable for direct customer interaction.

[03] Thus, there exists a need in the art for a user friendly virtual data center provisioning application that leverages existing components, and ties in new components to go from initial customer input to fully functioning virtual data center from shared resources.

BRIEF DESCRIPTION OF THE DRAWINGS

[04] FIG. 1 is an illustration of an example system according to an example embodiment of the present invention.

[05] FIG. 2 is an illustration of an example method, according to an example embodiment of the present invention.

[06] FIG. 3A and FIG 3B are example illustrations of example graphical user interfaces, according to example embodiments of the present invention.

[07] FIG. 4 is an example multi-tenant architecture, according to an example embodiment of the present invention.

DETAILED DESCRIPTION

[08] The present invention provides a portal manager able to facilitate Virtual Private Data Center (VPDC) configurations from multiple customers. The portal manager may interface with a client interface and facilitate drag-and-drop design of a VPDC using generic components with customizable features. During the design, the portal manager may receive validation checks from the user interface and facilitate validation for physical capacity, user privileges, and component compatibility. Each of the validity checks may be required to be performed for each customer selection with respect to the overall VPDC design. For example, the shared resources may have the physical capacity to facilitate a design choice, and the customer may have the available privileges for that design choice, but if provisioning the design would result in a failure of some other customer's service expectancies of greater priority, then the portal manager may declare the choice invalid.

[09] Once the design is completed by the user, and validated by the portal manager, a generic data-structure may be compiled based on the completed design file. The generic data-structure may have the relationships and attributes described or inherent to the design layout created by the user. This data-structure may then be parsed and filled with individual devices or logical segments of devices with reference to the CMDB. Each component of the CMDB may have a unique ID, and the CMDB may maintain information about each component's compatibility with other components so that the parser may select appropriate and available components to populate the data-structure. Once the data-structure includes specific device instances, a provisioning manifest may be compiled. The provisioning manifest may include a series of instructions for the various provisioning components, such that the entire provisioning process is automated from the manifest instructions. The system may facilitate this provisioning for multiple customers, and one or more VPDCs for each customer.

[10] FIG. 1 is a block diagram of a VPDC provisioning system 100. The system 100 may provide a software platform through which a VPDC customer (called "tenant" and/or user herein) initially configures or reconfigures a VPDC. The provisioning may be executed on a separate network (not shown) that may support VPDC provisioning for several tenants. The VPDC provisioning system may execute on one or more networked servers. Tenants may access VPDC provisioning services via a portal connection that extends from the provisioning system's servers to a client device managed by the tenant.

[11] FIG. 1 illustrates a software architecture of the VPDC provisioning system 100 that may execute on a server, e.g., server 110. The system 100 may include a portal manager 110.2 in communication with a client device 120 that manages the portal connection and captures user input. The portal manager 110.2 may provide a user with a user interface for designing a VPDC. For example, various design views may be transmitted to a workspace image (e.g., a user's browser), and receive various user selections. The portal manager 110.2 and/or workspace image may construct an XML design file that may include the final result of the design process.

[12] The system 100 may include a validation module 110.4 that may validate user provisioning requests and may reject requests that fail one or more validation tests. The validation module 110.4 may perform several validation checks. First, the validation module 110.4 may perform a check of the physical capacity by a physical asset management sub-module. The physical asset management sub-module may interface with a capacity management module 130 to determine the data center system has sufficient physical assets to meet the user request. The capacity management module 130 may store data representing the physical assets of the Data Center in its entirety and further representing physical assets that are allocated for use by other VPDCs of Data Center tenants. For example, a data center may organize physical compute hosts into clusters of sixteen. The sixteen host cluster may be assigned a maximum virtual compute unit for allocation. Each virtual host may consume different assigned compute units according QoS settings (e.g. four for Enterprise, two for Balanced and one for Essential). When a Balanced VPDC has two virtual hosts, four compute units may be deducted from the cluster. In addition to a unit host count to provide a level of service, the capacity management module may evaluate total VPDC compute resource consumption and determine the best compute cluster to provision based on available resources.

[13] Second, the validation module may perform an entitlement check to determine if a user's credit and permission status affords the user permission to allocate the requested resource(s). The entitlement sub-module may review a tenant service agreement (stored in a customer image database 140) to determine if the tenant has sufficient privileges to provision a selected resource in the tenant's VPDC design. The validation module may perform a web service call to the CMDB to retrieve an available credit limit for the calling user. For example, a user may have $10k MRR (monthly recurring revenue) approved for the account or pre-charged to the account (e.g., held as a positive account balance). The VPDCs may have MRR values

which may include the VPDC costs and sub-component costs, such as virtual machines (VMs) and storage. The cost of the selected VPDC may then be validated against the returned customer credit value, and deducted from the customer credit value once validated and provisioned. Third, the validation module 110.4 may include a design validation sub-module, which may determine if the tenant's selection represents a technically compatible selection.

[14] The system 100 may include a workspace manager 110.6 that may build a VPDC image based on the user input. If the tenant is reconfiguring a previously provisioned VPDC, the workspace manager may retrieve previously configured VPDC images from storage (shown as a customer image DB 140). If the tenant is building a VPDC from scratch, the workspace may be opened in a blank design area or opened from a template of VPDCs. The system 100 may include a provisioning manager 110.8, which may commit a workspace image when the tenant has completed its VPDC configuration. The workspace image may be created and stored as an XML design file, or any number of other file formats configured to store design data for the VPDCs. The image may be stored to the customer image database. The provisioning manager may automatically provision all VPDC elements using orchestration tools.

[15] At the client device 120, the portal interface may present a workspace for configuring the VPDC. The client interface may include a workspace displaying an image corresponding to an image maintained at the server 110. The client interface may include provisioning tools to open, save, and commit a workspace image. The client interface may include a template of VPDC resources that are available for addition to the workspace image (e.g., as further described with reference to Figure 3 below). The resources may be added to the workspace via drag-and-drop operations.

[16] Figure 2 illustrates one example embodiment of an example process. First, at 200, the process may receive input from a user about either a new VPDC or an existing VPDC. If the user is going to modify an existing VPDC, the process may load a previously created and saved design file (e.g., at 208). Alternatively, the process may re-construct a design file from a current configuration. In this way, a user may pull up their current configuration so that they may make modifications as needed. If the user is going to begin a new VPDC design, the example procedure may start with a base design selection (e.g. at 205). For example, the user may start by selecting one of several Quality of Service (QoS) levels (e.g., Enterprise, Balanced,

Essential). These base configurations may load a range of default settings appropriate for different environments, from testing/development to mission critical applications.

[17] Next, at 210, the user may be allowed to customize their VPDC, as the example procedure receives input, e.g., adding a new host to the configuration. Each user input may also be validated at 220 for one or more criteria (e.g., 110.4). In this example, the example procedure may check to ensure the service provider has the physical capacity to accommodate the user selection (e.g., at 222). The example procedure may check an entitlement associated with the customer to ensure the customer is entitled to construct a VPDC with the particular selection (e.g., at 224). The example procedure may check a configuration management database (CMDB) to ensure the selection made is technically compatible with the other selected options already part of the design. For example, if the user has a firewall service that attempts to take action on the inside interface with "Public" Internet, the example procedure may return an error, advising the customer that one or the other must change. Once the user input is fully validated, the example procedure may modify the XML design file to reflect the selection that was made (e.g., at 230). As illustrated in 235, the user input and validation process may repeat until the design is finalized. Once complete and finalized, the user interface may pass the finalized XML design file to a middleware application (e.g., at 240).

[18] At 245, the middleware application may construct a provisioning data-structure, which may leverage existing VM components and interfaces, while adding necessary attributes and functions for the example process. The provisioning data structure may act as a blue-print for the physical/logical allocations. The middleware application may retrieve CMDB attributes and insert into provisioning data-structure according to matching objects for CMDB and Cloud infrastructure to have consistent relationship records. From this, the example process may generate a provisioning manifest, e.g., a list of instructions the asset management system requires to finally construct and allocate the designed VPDC. This manifest may be sent to a provisioning engine, which may fully and automatically implement the validated design.

[19] The system may include a configuration management database (CMDB). Every item in the product catalog may have a unique component ID in the CMDB, along with a list of those components' compatibilities and incompatibilities with other components. When a user makes a design modification (e.g., adjusting the processor speed capability of a host from medium to large) the CMDB may be referenced to ensure there is some component available for facilitating

that design choice. Later, in actual back-end implementation, the example process may select that component, or a different but also compatible component for the logical provisioning.

[20] Figures 3A and 3B illustrate two simple drag and drop user interfaces. On the left side of the screen of Figure 3B is an example of an existing VPDC list, where saved VPDCs are listed. Selection of one of these finished or work-in-progress VPDCs may load the various configuration indicators into the Data Center Designer (e.g., as illustrated in the mid-screen section). Alternatively, a user may begin a new VPDC design, which may have one or more defaults loaded in, based on initial settings (e.g., QoS) indicated by the user. Default settings may also be included where a base level device (e.g., a firewall) is required. Here, the network connection may specify various components such as routers, port managers, bandwidth connections, etc. The external security section may have a default firewall, and the host banks may be empty as shown. The infrastructure catalog may be presented on the right and may contain various sections of the available product catalog, e.g., host systems and security modules (e.g., 330 and 340). A user may be able to click and drag items (e.g., firewall 2) over to the design area. As previously discussed, every action that modifies the design may be passed to the middleware for validation. For example, a user may be warned that implementing both the default firewall and firewall 2 is incompatible, and prompt the user to find out if the default firewall should be removed as a result of the addition of firewall 2. The user can fully customize the layout of the VPDC structure and design using these tools. Additional attributes are also customizable. For example, a user may move a Linux based server into the design. The user may then click on that server and adjust individual attributes, e.g., storage size, processor speed, etc.

[21] The middleware component may act as a broker between a user interface (e.g., XML-based drag-and-drop GUI) that provides the user representations of components in the product catalog, and one or more virtual provisioning mechanisms. The middleware may translate the XML GUI design into specially designed provisioning data-structures, modeled on the CMDB structure. The provisioning data-structure created in the middleware includes appropriate compatibility and design algorithms/rules, the VPDC may have every component provisioned in an inherently-validated design (e.g., due to the repetitive triple validation steps) based on the graphical layout provided by the user interface XML document.

[22] Responsive to a new provisioning request, a physical capacity check is performed to ensure the shared resource pool includes the capacity to perform the requested provisioning. Resources are classified as committed, reserved and available. Committed resources are capacity subscribed by customers. Reserved resources are capacity allocated for VPDCs in the design state. If a VPDC enters a saved state it releases allocated resources until loaded again. Available resources are capacity free for new VPDC design. The ratio of physical to virtual computing is different according to service grades. Example QoS levels may have descriptive titles (e.g., Enterprise, Balanced, Essential, etc.), and may allocate virtual machines per physical asset, according to the QoS level selected. Different physical hardware may have different sharing ratios for the different QoS levels.

[23] Just as validation is performed with respect to the requesting tenant's VPDC design, the physical capacity validation must be performed with respect to every other tenant's committed resource allocation and QoS requirements. Various formulas may be employed to ensure resource consumption does not exceed resource availability beyond the QoS levels implemented. Figure 4 illustrates one example embodiment of a multi-tenant architecture. There may be various shared assets 490, which may include servers, disk arrays, back-up drives, network connections, etc. There may be a system operation layer 450 that may span the virtual systems to facilitate the various example methods and provisioning requirements. There may be a provisioning portal manager 410.2 that may receive new VPDC requests from users, and facilitate the provisioning of those requests. There may be a resource use manager 410.3 that may facilitate the ongoing use of shared resources that have already been provisioned to users. Each tenant/user may have one or more client GUIs 420. As illustrated, some tenants may be new (e.g., tenant 1), some tenants may have ongoing VPDCs (e.g. tenants 3 to 4), and some tenants may have established VPDCs and request provisioning of additional VPDCs (e.g., tenant 2).

[24] It should be understood that there exist implementations of other variations and modifications of the invention and its various aspects, as may be readily apparent to those of ordinary skill in the art, and that the invention is not limited by specific embodiments described herein. Features and embodiments described above may be combined. It is therefore contemplated to cover any and all modifications, variations, combinations or equivalents that fall within the scope of the basic underlying principals disclosed and claimed herein.

WE CLAIM:

1. A method of provisioning a Virtual Private Data Center (VPDC), comprising:

providing a design user interface to a user;

receiving configuration selections from the user;

validating each configuration selection;

if validated, constructing a provisioning data structure based on the validated configuration selections;

populating the provisioning data structure with actual components; and

generating a provisioning manifest to a data center provisioning system for provisioning the VPDC.

2. The method of claim 1, wherein the configuration selections includes at least a quality of service indication.

3. The method of claim 1, wherein the validation includes validating the VPDC system for sufficient capacity, validating the entitlement of the user, and validating compatibility of the received configuration selections.

4. The method of claim 1, wherein the provisioning manifest includes a list of specific instructions that, when executed, facilitate a provisioning of a validated VPDC design.

5. The method of claim 1, wherein the validating is done for each selection, after the receiving of that respective selection and prior to the receiving of additional selections from the user.

6. The method of claim 5, wherein the constructing occurs after all selections have been validated.

7. The method of claim 1, wherein the validated configuration selections are stored in an XML design file.

8. The method of claim 1, wherein providing the design user interface includes providing a design area, providing a component area, and receiving drag and drop input from the user populating the design area with component selections from the component area.

9. The method of claim 1, wherein the provisioning of the VPDC is fully automated, such that the user is a customer designing and purchasing the VPDC, and wherein the VPDC is provisioned from the provisioning manifest without assistance from an administrator of the VPDC.

10. A method of supporting Virtual Private Data Center (VPDC) provisioning, comprising executing a VPDC provisioning system, at a server, that:

communicates with a user via a portal connection, receiving configuration selections from the user

validates configuration selection against a user service level agreement, available capacity resources for a data center that supports the VPDC and technical compatibility,

if validated, constructs a provisioning data structure based on the validated configuration selections;

assigns actual components to the provisioning data structure; and

outputs a provisioning manifest to a data center provisioning system for provisioning a VPDC.

11. The method of claim 10, that further comprises:

constructing the provisioning manifest based on the provisioning data structure that is populated with actual component assignments.

12. The method of claim 10, wherein the provisioning manifest includes instructions that cause the data center provisioning system to automatically provision a VPDC corresponding to the configuration selections from the user.

13. The method of claim 10, that further comprises:

executing the method for multiple distinct enterprise customers.

14. A VPDC provisioning computer system, comprising:

a design server configured to:

provide a portal connection to a graphical user interface, and receive selections from a user describing a VPDC configuration;

validate the configuration selections against a user service level agreement, available capacity resources for physical assets that support a plurality of VPDCs, and technical compatibility;

if validated, construct a provisioning data structure based on the validated configuration selections;

populate the provisioning data structure with actual components from the physical assets;

construct a provisioning manifest based on the populated provisioning data structure; and

outputting the provisioning manifest to a data center provisioning system for automatically provisioning a VPDC.

15. The VPDC provisioning computer system of claim 14, further comprising:

a provisioning server, configured to:

receive the provisioning manifest which includes a plurality of instructions, executable by the provisioning server, the cause the automatic provisioning of a VPDC described by the selections from the user;

execute the manifest's plurality of instructions to provision the VPDC.

16. The VPDC provisioning computer system of claim 14, further comprising:

responsive to user input received at the portal connection indicating a command to modify an existing VPDC:

loading a design file describing the existing VPDC; and

providing the user interface with the design file describing the existing VPDC.

17. The VPDC provisioning computer system of claim 16, further comprising:

responsive to user input received at the portal connection indicating a command to modify an existing VPDC:

reconstructing the design file based on an associated provisioning data structure.

18. The VPDC provisioning computer system of claim 14, wherein the user interface includes an interface for selecting saved or existing VPDC designs, an interface populated with components available for user selection, and a design area for receiving drag-and-drop input from the user consisting of the components available for user selection.

19. The VPDC provisioning computer system of claim 18, wherein the components available for user selection are abstract components with adjustable attributes; and

wherein a correlation between an abstract component with a specific attribute state and one or more sufficient physical components of the physical assets is maintained by the design server.

20. The VPDC provisioning computer system of claim 14, wherein the validating is performed after each user input that modifies the VPDC configuration, wherein invalid input produces an error message and valid input is committed to a design file.

21. The VPDC provisioning computer system of claim 20, wherein the provisioning data structure is constructed from the design file.