

PowerCenter 系统设计方案

Power Server 虚拟化管理平台

北京凌云动力科技有限公司

2016-03-03



目录

目录	1
第一章 概述	2
1.1 IBM Power 的基本介绍	2
1.2 PowerVM 虚拟化技术介绍	2
1.3 OpenStack 云平台介绍	3
第二章 POWERCENTER 设计方案	5
2.1 PowerCenter 设计架构	5
2.2 使用方式	5
2.3 运行机制	6
2.4 功能模块	6
第三章 云平台支持	7
3.1 OpenStack 云平台对 Power 小机虚拟化的支持	7
3.2 power 小型机管理系统网络架构	8

第一章 概述

1.1 IBM Power 的基本介绍

小型机，在国内习惯上用来指 UNIX 服务器，在服务器市场中处于中高端位置。小型机具有区别 X86 服务器和大型主机的特有体系结构，各厂商使用自家的 UNIX 版本的操作系统和专属处理器。小型机以其 RAS（Reliability（高可靠性）、Availability（高可用性）、Serviceability（高服务性））等特性，在金融、政府、电信、工业等领域广泛使用。

IBM Power 小型机，采用行业领先的 POWER 系列处理器，具有先进的 RAS 特性、极致的可扩充性和领先性能。在国内小型机市场上，IBM Power 小型机连续 12 年获得市场份额第一名，2013 年市场份额超过 70%。IBM 在中国每年大约有 8 亿到 10 亿美元的 Power 营业额。保守估计每年的销量是 20,000 台，市场存量 200,000 台以上。

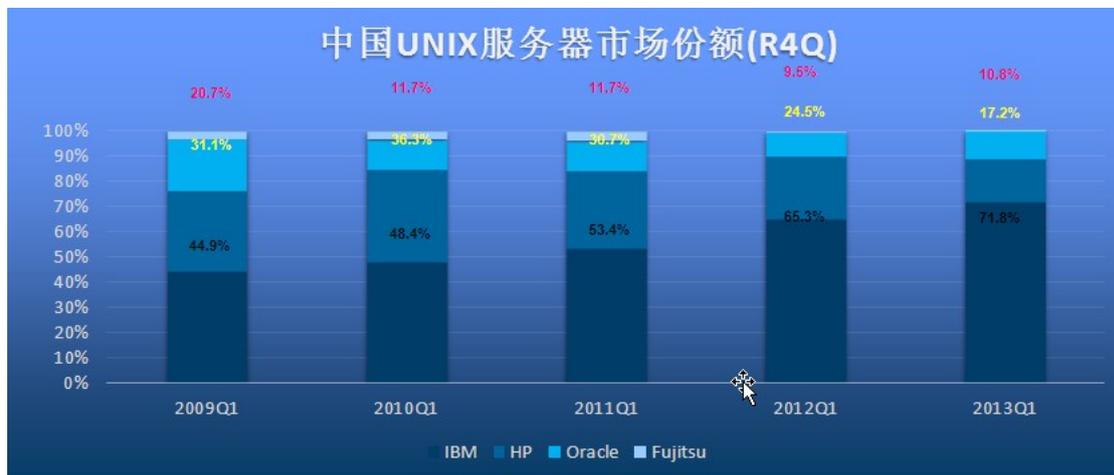


图 1 中国 UNIX 服务器市场份额

1.2 PowerVM 虚拟化技术介绍

PowerVM 是在基于 IBM POWER 处理器的硬件平台上提供的具有行业领先

水平的虚拟化技术家族，为IBM Power小型机提供了行业领先的虚拟化解决方案。PowerVM 基于十年多的演进和创新，代表一流的企业虚拟化技术，由大多数Power小型机所有者广泛部署于全球生产环境之中。

IBM Power小型机包括久经验证的工作负载整合平台，可以帮助客户在提高整体性能、可用性和能效的同时，有效控制成本。借助这些服务器和IBM PowerVM 虚拟化解决方案，企业可以整合大量的应用程序和服务、充分虚拟化系统资源，从而提供一个更具灵活性的动态IT 基础架构。换言之，IBM Power小型机结合PowerVM 技术可以无限自由地提供虚拟化优势。

PowerVM 为采用IBM POWER处理器的系统上的AIX、IBM i和Linux环境提供了行业领先的虚拟化技术。IBM Power 小型机结合PowerVM 技术可以支持构建动态基础架构、降低成本、管理风险并提高服务级别。

此外，由于IBM Power 小型机平台采用行业领先的POWER系列处理器，具有先进的RAS特性、极致的可扩充性和领先性能，而PowerVM 正是在此基础上构建而成的，因此能够提供安全灵活的虚拟化环境。

1.3 OpenStack 云平台介绍

为了更好的理解，我们首先来说一下云计算的类型。根据所提供服务的类型，云计算有以下三种落地方式：

IaaS(基础架构即服务)，用户能从中申请到硬件或虚拟硬件，包括裸机或虚拟机，然后在上边安装操作系统或其他应用程序。

PaaS(平台即服务)，用户能从中申请到一个安装了操作系统以及支撑应用程序运行所需要的运行库等软件的物理机或虚拟机，然后在上边安装其他应用程序，但不能修改已经预装好的操作系统和运行环境。

SaaS(软件即服务)，用户可以通过网络以租赁的方式来使用一些软件，而不是购买，比较常见的模式是提供一组账号密码。



OpenStack 就是一种 IaaS。2010 年 7 月，RackSpace 和美国国家航空航天局合作，分别贡献出 RackSpace 云文件平台代码和 NASA Nebula 平台代码，并以 Apache 许可证开源发布了 OpenStack，OpenStack 由此诞生。OpenStack 提供了租户管理、认证鉴权、计算、网络、存储等服务与管理功能。

第二章 PowerCenter 设计方案

pCenter 是一个 IBM Power 服务器集群管理软件,可以让用户通过简单的操作来管理整个集群。pCenter 的设计架构采用了主流的 B/S 架构,减轻了系统维护与升级的工作量,降低了用户的成本。

2.1 PowerCenter 设计架构

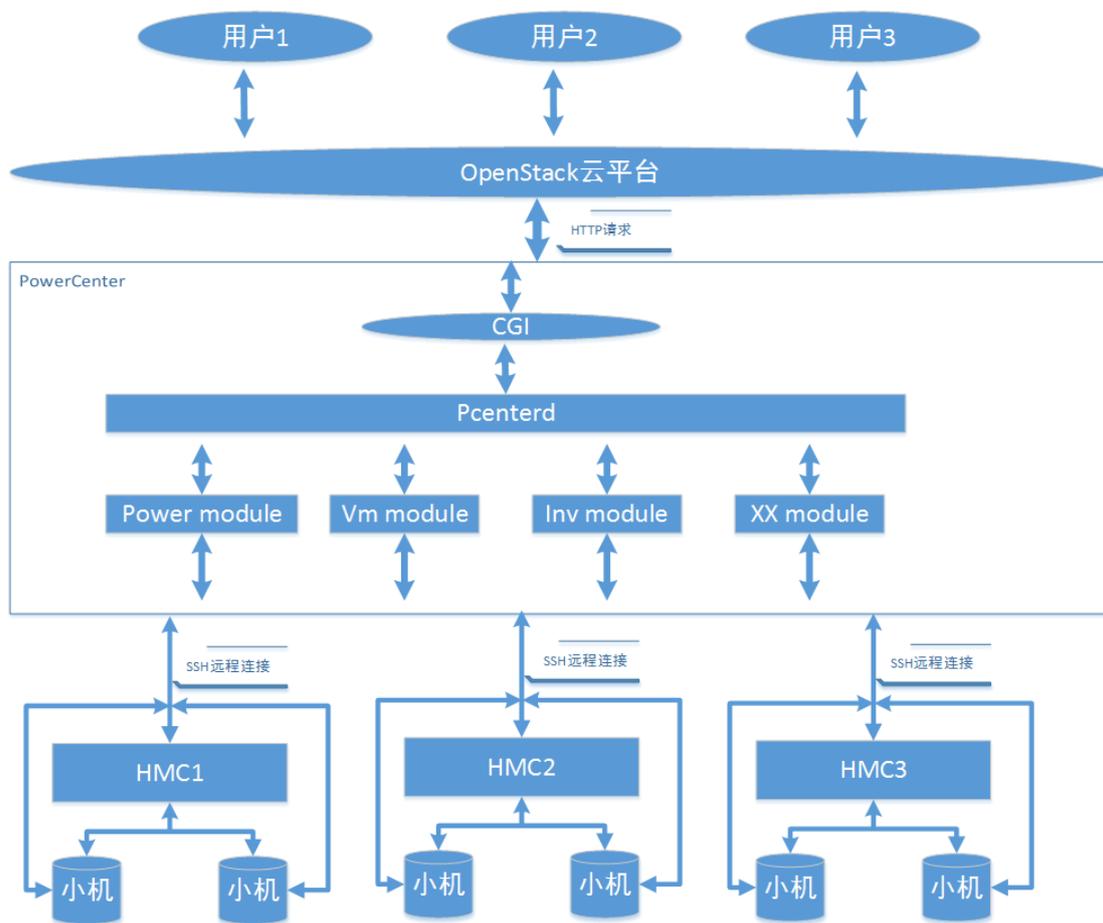


图 2 PowerCenter 系统架构图

2.2 使用方式

用户通过浏览器或 REST API 访问 pCenter, pCenter 的 CGI 模块到收到用

户的 HTTP 请求后，解析 HTTP 请求的方法和参数，如果请求的资源在 pCenter 中被定义，则把请求数据转换成 XML 格式，通过 SSLSocket 把请求数据发送给 pCenterd,并等待 pCenterd 的响应，收到 pCenterd 的响应后，把响应数据由 XML 转换成 JSON 格式反馈给用户；如果请求的资源没有在 pCenter 中定义或者程序运行错误则返回相应的错误信息。

2.3 运行机制

pCenterd 是整个软件的核心部分，当它接收到 CGI 发送过来的用 XML 封装的命令时，先解析出命令名、参数等信息，判断无误后把请求数据分派给相应的功能模块，然后等待接收被调用功能模块的处理结果，在接收到功能模块返回的数据后，把结果数据封装成 XML 格式发送给 CGI。

2.4 功能模块

pCenter 中有若干功能模块，它们实现了 pCenter 的大部分功能，如：主机操作、虚拟机操作、分组操作等功能。功能模块在被 pCenterd 调用时，会接收到 pCenterd 传过来的相关参数，根据命令、参数封装合适的 hmc 或 vios 命令，并确定命令在哪一个 hmc 或 vios 执行这些命令，然后建立 SSH 远程连接到该 hmc 或 vios 执行封装好的的命令，命令执行成功后把结果返回给 pCenterd。

第三章 云平台支持

3.1 OpenStack 云平台对 Power 小型机虚拟化的支持

OpenStack 云平台对 Power 小型机虚拟化支持功能模块如下图所示：

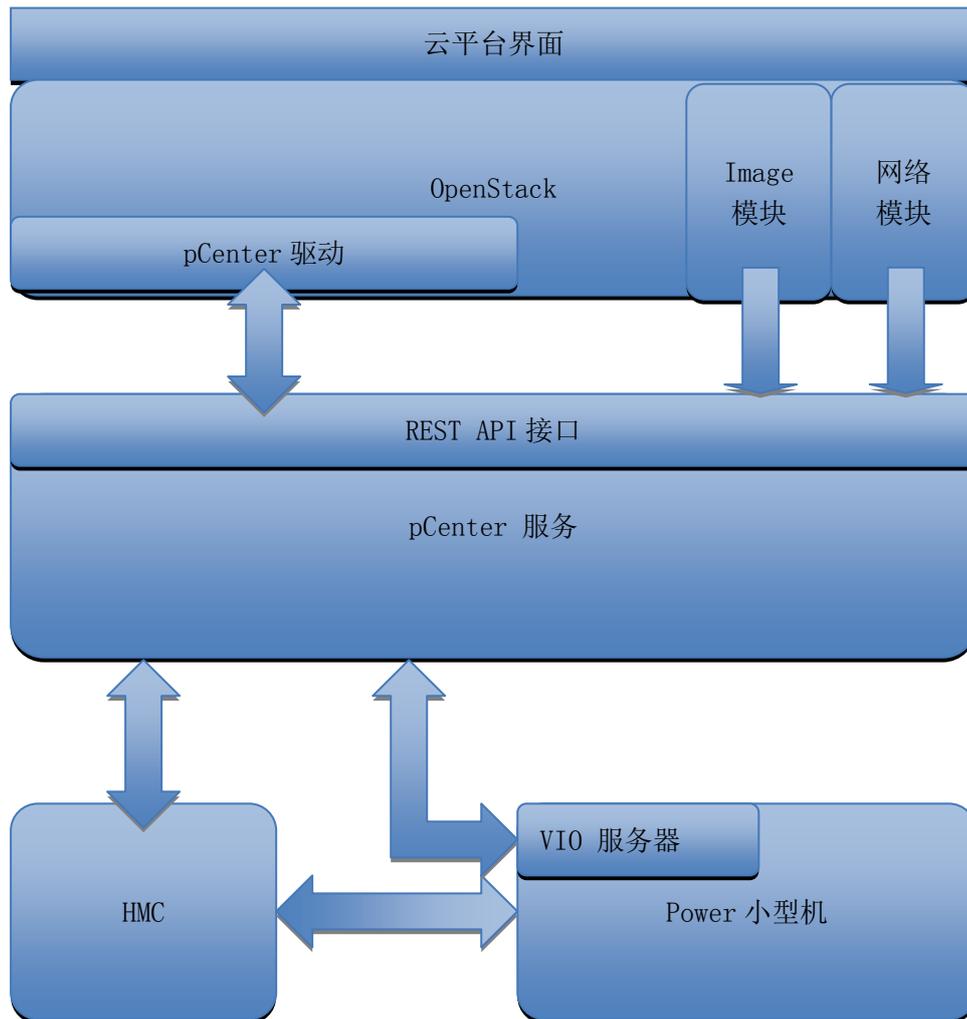


图 3 OpenStack 云平台对 Power 小型机虚拟化支持模型

pCenter 驱动是 OpenStack 云平台与 pCenter 服务联系的纽带，OpenStack 云平台通过它与 pCenter 服务的 REST API 接口进行通信，获取 pCenter 服务管理的 Power 小型机的信息，比如管理的 Power 小型机的电源状态、CPU 个数、内存大小、硬盘大小、网络连接信息等。

REST API 接口是 pCenter 服务对外提供的标准访问接口，外部程序或者客户端需要通过 REST API 与 pCenter 服务通信。

pCenter 服务是整个 Power 小型机管理系统的核心，它通过 HMC 来对 Power 小型机进行管理、控制、监控、获取小型机资源信息等，并且能够支持在 Power 小型机上进行微分区的创建、删除、查询，微分区硬件设备比如 SCSI 客户端、虚拟以太网卡的创建、修改、删除、查询等。pCenter 服务同时能够直接与 Power 小型机的 VIO 服务器通信，以进行 Power 小型机微分区硬盘、网络的创建和管理。

pCenter 在收集所管理的 Power 小型机的各种硬件信息比如 CPU 个数、内存大小、磁盘大小、网络信息的同时，还能够根据 OpenStack 云平台的虚拟机创建请求信息自动调度到合适的 Power 小型机上进行虚拟机的生命期管理。

HMC(Hardware Management Console)专门负责管理 Power 小型机，pCenter 正是通过 HMC 对 Power 小型机进行管理控制。

VIO 服务器是 Power 小型机上专门负责 I/O 管理的特殊分区，一般来说，它占有所有的 I/O 硬件资源比如硬盘控制器，物理网卡设备等，并基于这些物理设备产生服务器端为微分区提供虚拟 I/O 服务。而微分区将作为客户端使用 VIO 服务器提供的服务。

3.2 power 小型机管理系统网络架构

pCenter 服务作为 Power 小型机管理系统的核心，需要同时连接 OpenStack 云平台、HMC 模块和 Power 小型机。同时，Power 小型机的微分区需要从 OpenStack 云平台网络模块获取 IP 地址，从 image 模块获取 image 文件以启动微分区的系统。Power 小型机管理系统网络总体架构如下所示：

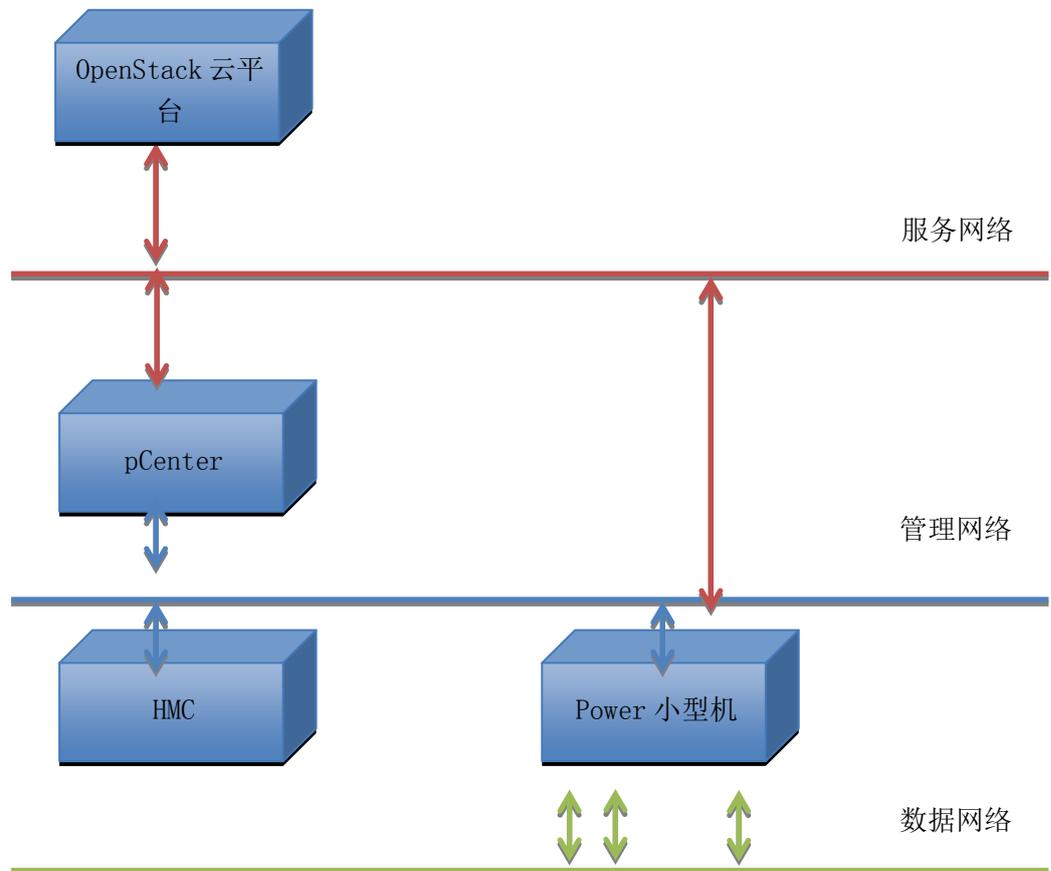


图 4 Power 小型机管理系统网络总体架构

服务网络：主要为 pCenter 访问 HMC 以管理 Power 小型机。主要作用是能够通过 HMC 获取管理的 Power 小型机的各种信息，并且能够访问 Power 小型机的 VIO 服务器以便配置微分区。同时为云环境提供服务，例如 DHCP，DNS，TFTP，HTTP 等。pCenter 连接在本网络里以便 OpenStack 云平台能够访问。

管理网络：本网络主要为管理 Power 虚机，实现对虚机的资源监控及动态调整虚机资源、虚机迁移等功能。

数据网络：主要负责部署的微分区的应用程序之间交换数据使用，这个网络可以按照具体需求进行配置。