谈谈我的windows服务器运维管理

我们开发的页游General War(http://gw.gamebox.com)上线运营也有半年多了，服务器的开发到运维基本都由我一手包办，在服务器上线之后我们又招了一个程序员接手后续功能的开发，而我则主要转到后台工具开发和服务器运维上。说到服务器的运维，我的全部经验就是维护过几台小型企业的域控，在linux上部署过几个web服务，以前做游戏的时候运维都是交给运营方去打点，而这次我是主动承担了这部分的工作。

由于我们的游戏服务器框架(EasyGame)是基于.net技术开发的，所以选择windows服务器来部署是比较自然的事情，虽然借助mono也可以在linux上部署，但毕竟不如windows的成熟。而一说到windows服务器，很多人会跟我说windows如何低效运维如何麻烦如何的不专业，我个人认为这只是观念问题罢了。由于linux的学习曲线比windows的高，过滤掉了一大批小白用户，所以给人感觉linux的运维更加高效更加专业。但实际上，不管是linux还是windows重要的还是看你如何去用。我个人是不太喜欢命令行方式的，因为使用命令行会带来记忆的负担，要记住那么多命令和参数实在是一件麻烦事，命令行的输出信息缺乏有效的组织和排版，阅读起来也是低效的。但是windows的图形管理界面也是让我想吐槽的地方，大部分系统设置的界面都没有经过良好的组织和设计从而变得极其难用而且低效（比如注册表，组策略，服务管理等等），windows的配置修改就像魔法一般的存在，经常让人摸不着头脑。因此我自己开发了一套自动化运维部署工具来帮助我管理服务器。

我的部署工具的设计目标是这样的：

1. 支持分布式的服务部署和管理，能够动态的管理多台宿主服务器
2. 不用登陆服务器来进行管理，所有对服务器的操作可以在GUI控制台程序上完成
3. 用GUI来降低使用难度，用友好的图表来组织服务器log数据和监控数据，便于阅读
4. 尽可能的降低配置的复杂度
5. 使用脚本来灵活的扩展工具以适应不同的需求

我们的服务器程序本身是分布式的，需要能够部署在多台服务器上，因此部署工具本身也是分布式的，我们有一个主控服务(ServiceManager)用于管理所有的宿主机以及宿主机上运行的服务，每台宿主机上有一个宿主机守护进程(ServiceHost)来对宿主机进行管理并向ServiceManager注册，从而组成一个服务器集群，然后用户通过一个GUI控制程序连接到ServiceManager上就可以管理和操作了。对于服务的配置，我采用的原则是约定优于配置，能不配置的东西尽可能做到免配置，对经常修改的配置和不太需要改动的配置分开处理，分成静态配置和动态配置。比如后端服务经常需要配置服务端口来暴露他的服务，端口的配置是相当麻烦的，搞不好就会出现端口占用冲突，所以这一块我们就使用动态端口来进行免配置，服务在ServiceManager上注册自己所开的端口，别的服务就可以得到服务端口。分布式服务会有多种服务类型，服务之间的互连，动态扩容等如果都通过配置文件来做也会非常麻烦，因为这一块我们使用服务名字约定来实现免配置，每一种服务使用一套名字约定来命名，比如前端代理服务器(AgentServer)就使用Agent.id来命名，当一个AgentServer在ServiceManager上注册之后就会广播给后端服务，后端服务在得到这个消息后决定是否要连接Agent来提供对外服务，一个后端服务在启动之后也可以查看当前集群中有哪些服务，然后决定是否要去建立连接。这样整个集群上的服务之间的连接就做到了免配置、启动顺序无关并且可以动态扩容。

对于GUI控制程序来说，GUI是对用户友好的，但缺点是开发成本较高而且不容易扩展，而命令行是天然的程序接口，对于程序本身十分友好，因此我的目标是要结合2者的优点，实现一个即容易使用又容易扩展的GUI程序。我的设计是这样的：首先，每个服务程序使用标准输入来接收运行命令，实现服务的启动、设置、管理和关闭，使用标准输出来输出格式化的log数据，这样的好处就是我的服务本身不依赖于服务容器，可以在命令行里可以当作一个普通的程序来运行和调试，整个设计是非侵入式的而且十分的KISS。然后，由ServiceHost作为服务容器来运行服务，ServiceHost接管服务的输入输出流，这样控制台程序就可以向服务发送命令了。每个服务所支持的命令及参数都是由服务自己决定的，在控制台程序上我们可以通过一个配置文件来告诉控制台程序这个服务支持哪些命令和参数，然后控制台程序会根据配置自动生成出对应的图形命令按钮和参数输入面板。通过配置就可以生成不同服务的管理控制界面，使得GUI有了很好的灵活性。最后，使用按钮虽然降低了使用难度，但是不利于批量操作，如果我一次有很多服务需要管理，那么通过点击按钮就是低效的，所以我集成了JavaScript来支持自动化脚本，这样就可以实现自动化的批量操作。

对于服务的更新，我集成了svn来实现资源文件的上传和分发，对小规模的集群来说，svn分发速度不是问题。如果是大规模集群的话，用svn作为资源分发源就有点捉襟见肘了，可以考虑集成p2p服务来实现资源的分发。服务本身支持多版本管理，在服务发布的时候可以选择不同的服务版本，来实现AB测试和灰度发布。

最后，整套部署工具的开发只用了不到8000行代码，2周的时间开发完成。目前我们架设了5台物理服务器来运行近40组游戏服，2台linux服务器装mysql做存储，2台做游戏应用服务器，1台做web资源下载和后台游戏管理工具。有了这套部署工具后，停服更新重启40组服务，最快只需要不到10分钟，运行也十分稳定，没有出现过crash事故。游戏从上线到现在，我们基本保持每周发布一个新版本的频率，这样高频率的版本更新导致游戏逻辑上难免存在很多bug，好在有部署工具的帮助使得查询log非常方便，对我们快速的定位和分析bug起到了很大的帮助。

总体而言，windows服务器的运维管理其实也不难做，懒惰的程序员会让运维变成一项适合懒人的工作。

标签: EasyGame, 服务器运维, 分布式

posted on 2014-01-03 01:42 qiaojie 阅读(11124) 评论(13)