# 动态负载均衡系统设计总览

## 系统总体设计

### 动态负载均衡系统需求

基于Hyper-V的动态负载均衡系统是一款服务于桌面Hyper-V虚拟机的优化插件。

### 动态负载均衡系统总览

基于Hyper-V的动态负载均衡系统为C/S架构，由两部分构成，分别是运行于Host的资源动态均衡器（Load-Balancer）和运行于VM的域内性能监测器（Perf-Detector），域内性能监测器为客户端，Host资源动态分配器为服务端，二者通过基于共享内存的Hyper-V socket进行通信，系统总览图如下图#所示：

域内性能监测器实时运行在开启状态的虚拟机，实时监测域内的性能指标，包括性能计数器中CPU、内存相关指标，更细粒度监测进程列表以及各进程占用的资源。每个开启的虚拟机中的性能监测器通过基于共享内存的HvSocket与Host通信，将性能信息发送给Host端的资源动态均衡器。资源动态均衡器可以直接监测各虚拟机的CPU占用率，但在静态内存分配的情况下，Host无法获取域内实际内存占用情况，更无法获取域内进程相关信息，因此资源动态分配器需要接收域内性能监测器发来的性能相关信息。资源动态均衡器根据各虚拟机的性能指标，以及Host的剩余资源情况，选择为资源紧张或空闲的虚拟机进行调度，回收空闲虚拟机已分配的内存和CPU资源，为负载高的虚拟机分配内存和CPU资源，可以提高负载高虚拟机的性能表现。

首先要解决的问题是：监测域内哪些指标能较为准确地反映虚拟机的负载情况，域内监测器不仅要监测CPU占用率和内存占用率，还需要监测反映资源需求情况的指标如线程等待队列长度Processor Queue Length、每秒产生的页错误次数Page Faults/secs。进一步地问题是，Host动态均衡器获取了所有性能指标，并进行分析后，应如何进行动态资源调度，Hyper-V提供了内存热插拔机制，在静态内存分配情况下，开机状态下可以在一定范围内调节静态分配给虚拟机的内存大小，如下图#所示：

Host动态均衡器通过设置开启虚拟机的内存大小来调度各虚拟机的内存资源。至于CPU资源，可以通过设置开启后虚拟机的“CPU限制”、“CPU保留”参数设置虚拟机实际可占用的CPU资源，举例如下图#所示：

给某虚拟机分配2个物理CPU，虚拟机中又两个vCPU，如果“CPU限制”为100，则该虚拟机最多占用全部2个物理CPU，将“CPU限制”改为500，则限制该虚拟机最多占用50%的CPU资源，即1个物理CPU。类似地，“CPU保留”指定了虚拟机最低保留的CPU资源比例，该比例需小于“CPU限制”。综上所述，可以有效动态调度CPU和内存资源。

### 详细设计

#### 域内性能监测器

GUI主线程用于多域PC界面的显示，各功能页面的切换，和用户交互并将用户的操作递交给后台逻辑进行处理。

#### 负载均衡器

1. 多域PC启动后，GUI主线程显示用户窗口界面。
2. 。

#### 异常和错误处理

1. 连接网络时用户输入密码错误，提示用户密码错误，让用户重新选择网络或重新输入密码。

#### 相关数据结构

用户使用信息配置文件、域配置信息文件、域状态结构、系统状态结构。

* 1. 域状态类 VMState

数据类型：class VMState

功能：标志当前虚拟机的状态

详细数据结构：

|  |  |  |  |
| --- | --- | --- | --- |
| Key | 含义 | 类型 | Value |
| VMName | Hyper-V中虚拟机名 | String | “Win10” |
| Installed | 该域是否已经创建 | enum | Installed,Uninstalled |
| State | 当前域的状态（开机、关机） | enum | On,Off |
| DefaultUser | 默认用户名 | string | “MultiPC” |
| Password | 密码 | string | “123456” |
| IPAddress | 域的IP地址 | string | “192.168.3.2” |
| StateSnapID | 当前域使用哪一个策略状态点 | int | 5 |
| RestoreStrategy | 还原策略 | Class CurrentStrategy |  |

* 1. 系统状态类 HostState

数据类型：class

功能：标识host管理器在使用过程中的一些基本信息，如当前联网状态、用户当前切入的域、是否连接U盘等

详细数据结构：

|  |  |  |  |
| --- | --- | --- | --- |
| Key | 含义 | 类型 | Value |
| HostUserName | 主机用户名 | string | “bt” |
| HostPasswd | 主机密码 | string | “123” |
| IsLogin | 是否登录 | bool |  |
| NetType | 网络类型 | enum |  |
| NetState | Host网络状态 | enum | “Connected”, “DisConnected” |
| CurrentVM | 当前切入的域（Host，本地域，互联网域） | enum | Host,Local,Net1,Net2 |
| EncryptRootPath | 加密区根目录 | “D:\\Encrypt” | EncryptRootPath |
| UConnect | U盘是否连接 | bool | true,false |
| ServerIP | 远程服务器的IP地址 | string | “192.168.3.1” |
| SelfIP | Host IP地址 | string | “192.168.3.25” |
| PC-ID | PCID | String |  |
| isAuthorized | 表示当前是否被授权 | bool | true,false |
| IsForbidden | 是否被禁止使用 | bool | true,false |

* 1. 当前用户类 CurrentUser

数据类型：abstract class

功能：该类是一个抽象类，有两个子类，NormalUser和Administrator类，分别表示普通用户和管理员用户。

* NormalUser类

普通用户类。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 含义 | 来源 |
| Password | String | 表示当前用户的密码 | 配置文件 |
| CurrentState | bool | 表示当前的登录状态 | 登录模块 |
|  |  |  |  |

* Administrator类

该类继承自User类。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 含义 | 来源 |
| QuestionAndAnswer | dictionary | 表示密保问题和答案 | 用户输入 |

* 1. 加密区信息类 EncryptDocumentation

数据类型：class

功能：加密区的相关信息

详细数据结构：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 含义 | 类型 | Value |
| RootPath | 加密区根目录 | string | “” |
| TotalSize | 加密区总大小 | string | “120GB” |
| UsedSize | 加密区已使用大小 | string | “100GB” |
| EncryptAlog | 加密算法 | string | “AES256” |
| EncryptKey | 加解密密钥 | string |  |

#### 流程图

# 域内性能监测器

### CPU监测模块

#### 功能

授权验证线程。

#### 设计

1. 该线程启动时间：在用户连接网络后，输入远程服务器IP地址，点击确认连接服务器后；
2. 授权验证线程需要将主机Host的IP地址发送给远程服务器，当服务器需要远程配置还原策略时，服务器需要建立与Host的socket；
3. 授权验证线程发送授权码等授权信息后，如果超时，socket会产生异常，在检测到该异常后要求用户重新输入授权码；
4. 网络监控线程实时监控网络状态，若发生断网则通知授权验证线程（授权验证线程存在），如果授权验证在等待回复，授权验证线程终止等待授权验证，将消息：“断网，授权验证失败，请重新连接网络”提示用户，GUI主线程设置授权状态为为未授权。

#### 异常和错误处理

1. 用户输入远程服务器IP，并点击确认连接后，本线程会首先尝试ping远程服务器，若无法正常连接，则提示用户重新检查服务器IP，并测试网络；
2. 授权未成功时，提示用户授权失败，根据授权失败信息，可以让用户重新输入授权码。
3. 授权验证线程发送验证码给服务器后，需要等待远程服务器应答，若此时网络断开，socket会直接响应异常，授权验证线程捕获该异常后，清空该socket，并提示用户网络断开，需要用户检查服务器IP或检测网络联通性，并重新输入授权码。

#### 相关线程

GUI主线程：本线程需要GUI界面用户输入的远程服务器IP地址，另外，若验证不成功，需要通知GUI主线程提示用户验证不成功，需要重新确认服务器IP和授权码；

远程服务器消息接收线程：运行在远程服务器端，监听Host端发来的授权请求，详情见远程服务器部分。

#### 相关数据结构

授权验证线程发送一次授权验证请求，所依赖的数据结构为一个AuthorityRequest类实例；接收远程服务器的验证信息是一个AuthorityResponse类实例，具体数据结构如下：

1. 授权请求类 AuthorityRequest

数据类型：class AuthorityRequest

功能：描述网络通信模块向远程服务器发送的授权请求

详细数据结构：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | Value |
| Index | 表示一次会话的标识符 | int | 1 |
| AuthorizationCode | 表示host管理器的授权码 | String | 配置文件 |
| PCID | 表示host管理器安装的计算机的唯一识别码 | string | 软件生成 |
| serialNumber | 表示host管理器的唯一识别码 | String | 软件生成/由pcid生成 |
| timestamp | 表示时间戳 | datetime | 读取系统信息 |

1. 授权接收类AuthorityResponse

数据类型：class AuthorityResponse

功能：描述远程服务器返回的授权响应

详细数据结构：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 含义 | 来源 |
| index | int | 表示一次会话的标识符 | 软件生成 |
| isAuthorized | bool | 表示当前是否被授权 | 服务器端消息 |
| isForbidden | bool | 表示host管理器所在的计算机是否被禁用 | 服务器端消息 |
| SerialNumber | String | 表示host管理器的唯一识别码 | 软件生成/由pcid生成 |
| Timestamp | datetime | 表示时间戳 | 读取系统信息 |

#### 流程图

用户连接网络

根据远程服务器IP地址，初始化授权验证线程

用户输入远程服务器IP，点击确认

尝试连接远程服务器，若无法连接，提示用户检查远程服务器IP

弹出对话框，要求用户输入授权码，与服务器建立socket连接

用户输入点击确认后，根据验证码生成相应的数据机构，向服务器发送授权验证请求

设置计时器，等待用户返回授权响应，若超时则提醒用户

授权验证线程接收到授权响应，解析到授权结果，将结果发送到GUI主线程

授权验证线程结束

### 内存监测模块

#### 功能

授权验证线程与服务器建立socket连接，向服务器发送授权码，进行授权验证，该线程将该授权结果发送给GUI主线程。若授权验证成功，GUI主线程将执行下一步操作；若授权不成功，GUI主线程提示用户授权码不正确，需要重新输入授权码。

#### 设计

1. 该线程启动时间：在用户连接网络后，输入远程服务器IP地址，点击确认连接服务器后；
2. 授权验证线程需要将主机Host的IP地址发送给远程服务器，当服务器需要远程配置还原策略时，服务器需要建立与Host的socket；
3. 授权验证线程发送授权码等授权信息后，如果超时，socket会产生异常，在检测到该异常后要求用户重新输入授权码；
4. 网络监控线程实时监控网络状态，若发生断网则通知授权验证线程（授权验证线程存在），如果授权验证在等待回复，授权验证线程终止等待授权验证，将消息：“断网，授权验证失败，请重新连接网络”提示用户，GUI主线程设置授权状态为为未授权。

#### 异常和错误处理

1. 用户输入远程服务器IP，并点击确认连接后，本线程会首先尝试ping远程服务器，若无法正常连接，则提示用户重新检查服务器IP，并测试网络；
2. 授权未成功时，提示用户授权失败，根据授权失败信息，可以让用户重新输入授权码。
3. 授权验证线程发送验证码给服务器后，需要等待远程服务器应答，若此时网络断开，socket会直接响应异常，授权验证线程捕获该异常后，清空该socket，并提示用户网络断开，需要用户检查服务器IP或检测网络联通性，并重新输入授权码。

#### 相关线程

GUI主线程：本线程需要GUI界面用户输入的远程服务器IP地址，另外，若验证不成功，需要通知GUI主线程提示用户验证不成功，需要重新确认服务器IP和授权码；

远程服务器消息接收线程：运行在远程服务器端，监听Host端发来的授权请求，详情见远程服务器部分。

#### 相关数据结构

授权验证线程发送一次授权验证请求，所依赖的数据结构为一个AuthorityRequest类实例；接收远程服务器的验证信息是一个AuthorityResponse类实例，具体数据结构如下：

1. 授权请求类 AuthorityRequest

数据类型：class AuthorityRequest

功能：描述网络通信模块向远程服务器发送的授权请求

详细数据结构：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | Value |
| Index | 表示一次会话的标识符 | int | 1 |
| AuthorizationCode | 表示host管理器的授权码 | String | 配置文件 |
| PCID | 表示host管理器安装的计算机的唯一识别码 | string | 软件生成 |
| serialNumber | 表示host管理器的唯一识别码 | String | 软件生成/由pcid生成 |
| timestamp | 表示时间戳 | datetime | 读取系统信息 |

1. 授权接收类AuthorityResponse

数据类型：class AuthorityResponse

功能：描述远程服务器返回的授权响应

详细数据结构：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 含义 | 来源 |
| index | int | 表示一次会话的标识符 | 软件生成 |
| isAuthorized | bool | 表示当前是否被授权 | 服务器端消息 |
| isForbidden | bool | 表示host管理器所在的计算机是否被禁用 | 服务器端消息 |
| SerialNumber | String | 表示host管理器的唯一识别码 | 软件生成/由pcid生成 |
| Timestamp | datetime | 表示时间戳 | 读取系统信息 |

#### 流程图

用户连接网络

根据远程服务器IP地址，初始化授权验证线程

用户输入远程服务器IP，点击确认

尝试连接远程服务器，若无法连接，提示用户检查远程服务器IP

弹出对话框，要求用户输入授权码，与服务器建立socket连接

用户输入点击确认后，根据验证码生成相应的数据机构，向服务器发送授权验证请求

设置计时器，等待用户返回授权响应，若超时则提醒用户

授权验证线程接收到授权响应，解析到授权结果，将结果发送到GUI主线程

授权验证线程结束

### 进程监测模块

#### 功能

授权验证线程与服务器建立socket连接，向服务器发送授权码，进行授权验证，该线程将该授权结果发送给GUI主线程。若授权验证成功，GUI主线程将执行下一步操作；若授权不成功，GUI主线程提示用户授权码不正确，需要重新输入授权码。

#### 设计

1. 该线程启动时间：在用户连接网络后，输入远程服务器IP地址，点击确认连接服务器后；
2. 授权验证线程需要将主机Host的IP地址发送给远程服务器，当服务器需要远程配置还原策略时，服务器需要建立与Host的socket；
3. 授权验证线程发送授权码等授权信息后，如果超时，socket会产生异常，在检测到该异常后要求用户重新输入授权码；
4. 网络监控线程实时监控网络状态，若发生断网则通知授权验证线程（授权验证线程存在），如果授权验证在等待回复，授权验证线程终止等待授权验证，将消息：“断网，授权验证失败，请重新连接网络”提示用户，GUI主线程设置授权状态为为未授权。

#### 异常和错误处理

1. 用户输入远程服务器IP，并点击确认连接后，本线程会首先尝试ping远程服务器，若无法正常连接，则提示用户重新检查服务器IP，并测试网络；
2. 授权未成功时，提示用户授权失败，根据授权失败信息，可以让用户重新输入授权码。
3. 授权验证线程发送验证码给服务器后，需要等待远程服务器应答，若此时网络断开，socket会直接响应异常，授权验证线程捕获该异常后，清空该socket，并提示用户网络断开，需要用户检查服务器IP或检测网络联通性，并重新输入授权码。

#### 相关线程

GUI主线程：本线程需要GUI界面用户输入的远程服务器IP地址，另外，若验证不成功，需要通知GUI主线程提示用户验证不成功，需要重新确认服务器IP和授权码；

远程服务器消息接收线程：运行在远程服务器端，监听Host端发来的授权请求，详情见远程服务器部分。

#### 相关数据结构

授权验证线程发送一次授权验证请求，所依赖的数据结构为一个AuthorityRequest类实例；接收远程服务器的验证信息是一个AuthorityResponse类实例，具体数据结构如下：

1. 授权请求类 AuthorityRequest

数据类型：class AuthorityRequest

功能：描述网络通信模块向远程服务器发送的授权请求

详细数据结构：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | Value |
| Index | 表示一次会话的标识符 | int | 1 |
| AuthorizationCode | 表示host管理器的授权码 | String | 配置文件 |
| PCID | 表示host管理器安装的计算机的唯一识别码 | string | 软件生成 |
| serialNumber | 表示host管理器的唯一识别码 | String | 软件生成/由pcid生成 |
| timestamp | 表示时间戳 | datetime | 读取系统信息 |

1. 授权接收类AuthorityResponse

数据类型：class AuthorityResponse

功能：描述远程服务器返回的授权响应

详细数据结构：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 含义 | 来源 |
| index | int | 表示一次会话的标识符 | 软件生成 |
| isAuthorized | bool | 表示当前是否被授权 | 服务器端消息 |
| isForbidden | bool | 表示host管理器所在的计算机是否被禁用 | 服务器端消息 |
| SerialNumber | String | 表示host管理器的唯一识别码 | 软件生成/由pcid生成 |
| Timestamp | datetime | 表示时间戳 | 读取系统信息 |

#### 流程图

用户连接网络

根据远程服务器IP地址，初始化授权验证线程

用户输入远程服务器IP，点击确认

尝试连接远程服务器，若无法连接，提示用户检查远程服务器IP

弹出对话框，要求用户输入授权码，与服务器建立socket连接

用户输入点击确认后，根据验证码生成相应的数据机构，向服务器发送授权验证请求

设置计时器，等待用户返回授权响应，若超时则提醒用户

授权验证线程接收到授权响应，解析到授权结果，将结果发送到GUI主线程

授权验证线程结束

### 进程控制模块

#### 功能

授权验证线程与服务器建立socket连接，向服务器发送授权码，进行授权验证，该线程将该授权结果发送给GUI主线程。若授权验证成功，GUI主线程将执行下一步操作；若授权不成功，GUI主线程提示用户授权码不正确，需要重新输入授权码。

#### 设计

1. 该线程启动时间：在用户连接网络后，输入远程服务器IP地址，点击确认连接服务器后；
2. 授权验证线程需要将主机Host的IP地址发送给远程服务器，当服务器需要远程配置还原策略时，服务器需要建立与Host的socket；
3. 授权验证线程发送授权码等授权信息后，如果超时，socket会产生异常，在检测到该异常后要求用户重新输入授权码；
4. 网络监控线程实时监控网络状态，若发生断网则通知授权验证线程（授权验证线程存在），如果授权验证在等待回复，授权验证线程终止等待授权验证，将消息：“断网，授权验证失败，请重新连接网络”提示用户，GUI主线程设置授权状态为为未授权。

#### 异常和错误处理

1. 用户输入远程服务器IP，并点击确认连接后，本线程会首先尝试ping远程服务器，若无法正常连接，则提示用户重新检查服务器IP，并测试网络；
2. 授权未成功时，提示用户授权失败，根据授权失败信息，可以让用户重新输入授权码。
3. 授权验证线程发送验证码给服务器后，需要等待远程服务器应答，若此时网络断开，socket会直接响应异常，授权验证线程捕获该异常后，清空该socket，并提示用户网络断开，需要用户检查服务器IP或检测网络联通性，并重新输入授权码。

#### 相关线程

GUI主线程：本线程需要GUI界面用户输入的远程服务器IP地址，另外，若验证不成功，需要通知GUI主线程提示用户验证不成功，需要重新确认服务器IP和授权码；

远程服务器消息接收线程：运行在远程服务器端，监听Host端发来的授权请求，详情见远程服务器部分。

#### 相关数据结构

授权验证线程发送一次授权验证请求，所依赖的数据结构为一个AuthorityRequest类实例；接收远程服务器的验证信息是一个AuthorityResponse类实例，具体数据结构如下：

1. 授权请求类 AuthorityRequest

数据类型：class AuthorityRequest

功能：描述网络通信模块向远程服务器发送的授权请求

详细数据结构：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 含义 | 类型 | Value |
| Index | 表示一次会话的标识符 | int | 1 |
| AuthorizationCode | 表示host管理器的授权码 | String | 配置文件 |
| PCID | 表示host管理器安装的计算机的唯一识别码 | string | 软件生成 |
| serialNumber | 表示host管理器的唯一识别码 | String | 软件生成/由pcid生成 |
| timestamp | 表示时间戳 | datetime | 读取系统信息 |

1. 授权接收类AuthorityResponse

数据类型：class AuthorityResponse

功能：描述远程服务器返回的授权响应

详细数据结构：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 含义 | 来源 |
| index | int | 表示一次会话的标识符 | 软件生成 |
| isAuthorized | bool | 表示当前是否被授权 | 服务器端消息 |
| isForbidden | bool | 表示host管理器所在的计算机是否被禁用 | 服务器端消息 |
| SerialNumber | String | 表示host管理器的唯一识别码 | 软件生成/由pcid生成 |
| Timestamp | datetime | 表示时间戳 | 读取系统信息 |

#### 流程图

用户连接网络

根据远程服务器IP地址，初始化授权验证线程

用户输入远程服务器IP，点击确认

尝试连接远程服务器，若无法连接，提示用户检查远程服务器IP

弹出对话框，要求用户输入授权码，与服务器建立socket连接

用户输入点击确认后，根据验证码生成相应的数据机构，向服务器发送授权验证请求

设置计时器，等待用户返回授权响应，若超时则提醒用户

授权验证线程接收到授权响应，解析到授权结果，将结果发送到GUI主线程

授权验证线程结束

# Host负载均衡器

## 系统总体设计