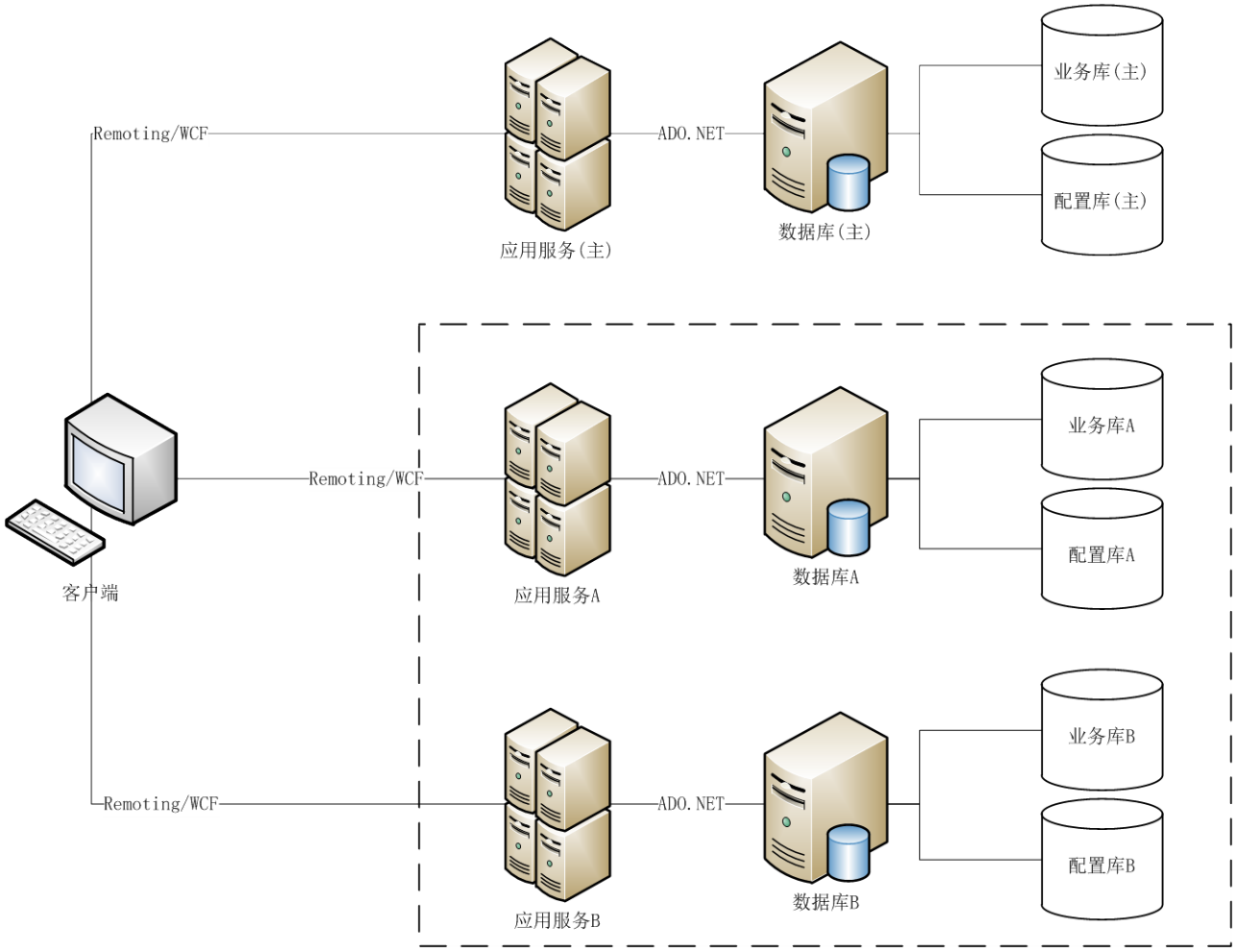
18 异构服务集群

一提到服务集群，往往联想到的是操作系统所提供的集群服务（虚拟 IP 服务），其功用是使得两台或多台服务器像一台服务器一样工作，提供更高的可用性和性能等。

本文所述的 Phenix丶异构服务集群技术，着重的是异构概念，实现的是在同一个应用系统环境中，可多个应用服务同时响应单个客户端，从而使得架设异构的应用服务集群和数据库成为可能。

18.1 架构实现

Phenix丶异构服务集群技术，可实现的架构如下所示：



需注意的是，虚线内的应用服务数量可不止如上图所示的 A、B 两台（注意：A 和 B 也不一定是一台服务，可能是操作系统集群服务提供的虚拟 IP 服务、或者是 Phenix丶提供的分布式架构下的同构服务集群），可配置上多个，一起组成了一个异构服务集群。

18.2 使用目的和应用场景

一般应用，我们都是采取一套分布式应用服务+单个数据库（可能采用了容灾技术，但从应用层面仍然可视为一个数据库）的架构，只有在如下需求中才用得到异构服务集群：

- 希望将消耗资源、影响性能的功能模块服务端单独运行在指定的服务器上，以减少对正常操作的影响，提高响应客户端的执行效率：这种情况，整个应用系统使用的是同一个数据库；
- 希望整个系统使用多个数据库来做分布式存储：这种情况，整个应用系统需要有一个主数据库，存放配置库的全集，用户权限验证等功能需使用到主数据库（客户端登录时调用的是主应用服务），而服务集群里的数据库不必存放全集；

18.3 开发接口

使用 Phenix丶来设计一个异构服务集群的应用系统并不复杂，仅需在两处代码段上使用 Phenix丶提供的接口方法即可。

18.3.1注册服务集群

网络配置信息（Phenix.Core.Net.NetConfig）静态类提供了注册服务集群的方法：

```
/// <summary>
/// 注册服务集群
/// </summary>
/// <param name="key">键值</param>
/// <param name="servicesAddress">服务器IP地址或者名称</param>
public static void RegisterServicesCluster(string key, string servicesAddress)
```

我们一般是在客户端启动时调用它来注册，示例如下：

```
static class Program
{
  /// <summary>
  /// 应用程序的主入口点。
  /// </summary>
  [STAThread]
  static void Main(string[] args)
  {
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);

    Phenix.Core.Net.NetConfig.RegisterServicesCluster("本地服务", "127.0.0.1");

    using (LogOn logOn = new LogOn { Title = "登录XXX系统" })
    {
      IPrincipal user = logOn.Execute<LogOnDialog>();
      if (user != null && user.Identity.IsAuthenticated)
```

```
        {
          PluginHost.Default.SendSingletonMessage("Phenix.Windows.Main", null);
        }
      }
    }
  }
```

服务集群的服务器 IP 地址是按照键值（key 参数值）储存在本地 config 文件中的，所以注册后重启程序，即使不再调用 RegisterServicesCluster()函数，这之前注册好的内容也是不会丢失的，还能继续使用（当然，如果删除了 config 文件就另当别论了）。而当再次以这个键值注册新的 servicesAddress 参数值的话，旧值就会被覆盖掉。所以，我们可以利用这个特性，在需要的时候动态变更连接服务器的 IP 地址。

18.3.2标记 Root 对象的类

受本功能影响的是业务类（集合类）的 Fetch()、Save()、DeleteRecord()、GetRecordCount()、CheckRepeated()，除此之外，比如 SequenceValue、BusinessCode 等功能都直接从主服务获取。

需注意的是，只有 Root 对象才能决定到底是调用哪个服务器，所以我们要在 Root 对象的类打上服务集群的标记，如下示例：

```
/// <summary>
/// 用户清单
/// </summary>
[Serializable]
[Phenix.Core.Net.ServicesClusterAttribute("本地服务")]
public class UserList : Phenix.Business.BusinessListBase<UserList, User>
{
}
```

这样，当调用 UserList 的 Fetch()时，调用的服务将是在 IP 等于 127.0.0.1 的服务器上。其他通过应用服务操作业务数据库的函数（比如 Save()函数等）也是如此。