

MYCP 分析设计说明书

Copyright©2007-2010 Akee H.D

作者: Akee Yang akee.yang@gmail.com

(转载请注明本书作者! 未经允许, 严禁把本书用于商业目的!)

目录

1、	MYCP介绍	3
2、	MYCP项目背景	4
2.1.	存在问题	4
2.2.	MYCP提供的解决方案	4
3、	MYCP的应用领域	5
4、	MYCP的特点	5
5、	MYCP的设计思路	6
6、	运行开发环境	7
7、	功能描述和工作原理	7
8、	MYCP的体系结构	8
8.1.	MYCP的四层结构	8
8.2.	MYCP应用组件	9
8.3.	客户层组件	9
8.4.	WEB组件	10
8.5.	业务层组件	10
8.6.	企业信息系统层	11
9、	关键技术点	11
10、	功能模块设计	12
11、	相关软件分析及总结	12

1、 MYCP 介绍

MYCP 是用于开发、集成、部署和管理大型分布式应用、网络应用和数据库应用的 C++ WEB 应用服务器。将 C++ 的高效稳定和各種通訊标准 (HTTP/UDP/TCP/RTP 等) 引入 WEB 应用的开发、集成、部署和管理之中。

MYCP 拥有处理关键分布式应用系统问题所需的性能、可扩展性和高可用性。

MYCP 所需的多种特色和优势，包括：

- **领先的标准**

对业内多种通讯标准的全面支持，包括 HTTP、TCP、UDP、RTP，使分布式应用系统的实施更为简单，并且保护了投资，同时也使基于标准的解决方案的开发更加简便。

- **无限可扩展性**

MYCP 以其高扩展的架构体系闻名，包括内置 SOTP 通讯协议、访问用户管理和后台多种群集功能。

- **快速开发**

MYCP 独特的设计体系模式，可简化开发，加速投放市场速度。并可利用已有技能，迅速部署应用系统。

- **部署更趋灵活**

一次开发，支持多平台运行，如 Windows 和 Linux 系统。

支持所有标准浏览器客户端。

- **体系结构**

MYCP 是专门为企业分布式 WEB 应用服务系统开发。企业分布式网络应用系统需要快速开发，并要求服务器端组件具有良好的灵活性和安全性，同时还要支持关键任务所必需的扩展、性能、和高可用性。

MYCP 简化了可移植及可扩展的应用系统的开发，并为其它应用系统提供了丰富的互操作性。

MYCP 拥有最高水平的可扩展性和可用性。轻松实现访问验证功能、负载均衡和群集功能。

2、 MYCP 项目背景

2.1. 存在问题

- **服务器的问题:**

我们觉得，选择做 JAVA 的开发人员是幸福的，为什么？因为在 JAVA 环境下，有各种 WEB 应用服务器可以选择使用，随便写个 JSP ，或者 .class 类，部署到 WEB 服务器下，可以讲不到一分钟时间，就可以实现一个功能，提供远程网络服务；而且 JAVA 下也有很多著名的成熟开发框架和应用框架，方便开发人员进行开发，测试和部署。

而 C++ 的开发人员运气似乎就没有这么好了， C++ 开发环境下没有类似 Apache ， Tomcat 甚至昂贵的 WebLogic 应用服务器，要实现一个网络应用服务，都得从最基本的 SOCKET 通讯写起，虽然有 ACE 、 boost::asio 等相关类库，但同样要考虑端口连接管理，多线程管理，内存分布，访问调度以及通讯协议等问题。

等到解决完所有这些技术细节，开始做业务设计、开发的时候才发现项目进度已经去了一大半，说不写客户已经在开始提第二期的需求；并且这种系统的稳定性和维护性对开发人员来说也将是一个很大的挑战。

- **Web 开发的问题:**

Web 开发是基于脚本语言的开发，以快速开发、部署、升级和免安装特性著称；但一直以来所有的脚本语言，包括 ASP、JSP、PHP 等都有一个特性，**新语言底层封装**。

类似 JSP (JAVA) 这种**新语言底层封装**的优点是跨平台，但缺点就是系统运行效率低，部署管理的成本比较高；另外必须提供开发者一整套全新的语言规范和类库，所有需要实现的功能都必须捆绑在该规范上。

2.2. MYCP 提供的解决方案

- **HTTP、Web 服务器:**

MYCP 是一套完全基于 C++ 编程语言开发的，HTTP 服务器、C++ Web 应用服务器；MYCP 利用 C++ 语言的灵活、高效、平台无关等特性，实现 HTTP 接入协议，提供一整套的基于 C++ Web 开发解决方案。

利用 MYCP，C/C++ 开发人员可以跟 JAVA 开发人员一样，编写一个 .c 或者 .cpp 文件，直接编写业务代码，而不用管理所有的通讯协议、传输能力、内存管理、任务调度等。专注开发业务逻辑，提高开发效率。

做过统计，利用 MYCP 做网络应用开发，开发效率可以提高 20 倍以上，而代码量减小了 80% 以上。

- **CSP+ Web 容器：**

MYCP CSP+ 是一套 XML 格式的页面开发标准，在 HTML 页面插入 CSP 标签，实现 Web 页面开发功能。

MYCP 建立一套标准组件交互框架，让 Web 页面可以跟 C++ 语言开发的组件（DLL 或 so 文件）进行交互，经过测试，这种系统的运行效率是最高的，而且在功能上不受任何语言的限制，同样实现跨平台。

3、 MYCP 的应用领域

- 公司网站和企业门户
- 大型分布式系统服务
- 硬件或嵌入式系统 HTTP、WEB 服务器
- 各种异构环境平台（跨开发语言、OS 和硬件等）

4、 MYCP 的特点

- 平台无关：一次编写，到处运行；
- 组件化：通讯能力、协议解析、任务调度和应用组件等，组件化配置调用，所有功能组件通过接口进行交互通信；
- 标准化：提供 CDBC、CSP、C++ APP 和 C++ Servlet 等标准接口，方便开发人员开发业务系统；
- 可扩展性：提供丰富开发类库和 API 接口，非常简易实现功能替换、更

新、升级和二次开发等；

- 异构：支持跨平台、跨开放语言、跨操作系统、跨硬件、跨 C/S 和 B/S 等各种异构环境应用；
- 可伸缩性：支持最小几 M 的嵌入式硬件系统，到大型分布式系统集群部署；
- 免管理：通过 XML 配置，LOG 日志跟踪等特性，MYCP 的部署管理成本低；

5、 MYCP 的设计思路

- 使用 C++开发语言：

C++语言灵活、高效、功能强大，开源资源丰富，只要具标准 C++编译器，MYCP 开发的应用组件就可以部署运行，不用代码托管，真正意义上的跨平台。

- 使用 boost 标准 C++开发库：

boost 是一个准标准库，相当于 STL 的延续和扩充，它的设计理念和 STL 比较接近，都是利用泛型让复用达到最大化。不过对比 STL，boost 更加实用。STL 集中在算法部分，而 boost 包含了不少工具类，可以完成比较具体的工作。

MYCP 主要使用了 boost 库的，shared_ptr 智能指针，asio 网络通讯，thread 多线程，filesystem 文件系统和 property 配置管理库等等。

- 组件模块化开发和配置：

MYCP 将所有的业务逻辑进行分类，分为 COMM 通讯、LOG 日志、PARSER 解析、APP 应用和 SERVER 服务，5 种组件类型。

COMM 通讯组件用于实现数据传输，网络通讯；LOG 日志组件用于日志跟踪，备份等；PARSER 解析组件实现不同应用业务的协议解析；APP 应用组件实现具体的业务逻辑处理；SERVER 服务组件提供完整服务能力；

所有的组件使用统一接口进行开发，通过 XML 文件进行配置部署，通过标准接口进行通讯和交互。

- 标准 HTTP 服务器：

MYCP 实现了解析和封装 HTTP 协议，实现标准 HTTP 服务器功能，使用 MYCP 不需要单独安装 HTTP 服务器（类似 apache），可以直接支持标准 HTML Web 应

用部署，包括代理、转发、验证等功能。

- CSP 脚本和 CSP+容器：

CSP (C++ Server Pages) 是一套 XML 格式的页面标签，利用 CSP 实现 web 页面开发功能，包括 HTML 页面输出，跟 CDBC (C++数据库互连技术) 和其他所有 C++ APP 应用组件交互等等。

所有技术统称为 CSP+，MYCP 实现对 CSP+的容器包含支持功能。

- SOTP 简单对象传输协议：

SOTP 类似 SOAP 协议，是真正意义的简单功能，SOTP 是一套基于会话式的文本行通讯协议，跟开发语言无关，实现数据类型，数据传输、远程访问等功能；

MYCP 利用 SOTP 协议，方便实现跨平台、跨开发语言，跟 C/S 和 B/S 等各种异构环境应用开发。

6、 运行开发环境

原则上 MYCP 支持部署运行在所有具 C++编译器环境中，目前 MYCP 经过测试可正常运行有 NT、XP、CentOS、Ubuntu 等各个不同发布版本环境；MYCP 开放源代码，相信通过编译 MYCP 可以支持其他更多环境。

MYCP 使用标准 C++语言开发，主要使用了 boost 开源库协助开发，在 windows 环境上使用 VC2005 开发编译，然后在 linux 环境下通过 makefile 脚本进行自动编译链接。

7、 功能描述和工作原理

- HTTP 服务器功能：

实现 HTTP 协议的解析和封装，支持基本标准 HTTP 服务器功能，包括 HTML 页面输出、页面跳转、代理重定向、自动验证、文件上传下载等功能。

- 动态 HTML 页面输出功能：

利用 CSP 标签，跟 HTTP 协议解析组件和 C++ APP 应用组件交互，实现动态 HTML 页面输出功能。

- 标准数据库访问接口 CDBC：

定义 CDBC 标准接口，用于执行更新和查询等 SQL 语句。

- 访问远程组件技术 RCA:

利用标准 SOTP 通讯协议，定义访问组件名称、方法名称、数据类型，入参数和输出参数等，实现访问远程组件能力。

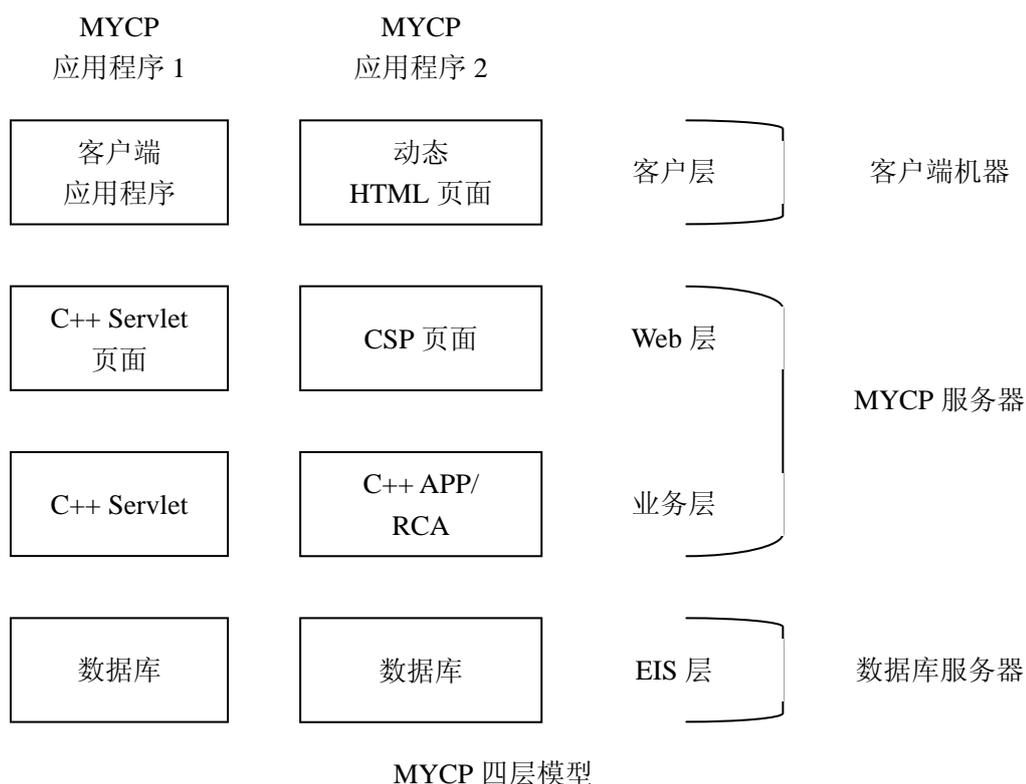
8、 MYCP 的体系结构

MYCP 是一种利用 C++编写，来简化企业解决方案的开发、部署和管理相关的复杂问题的体系结构。MYCP 技术的基础就是核心 C++语言平台，所以 MYCP 不仅拥有，例如“编写一次、随处运行”、“性能稳定”的特性，方便存取数据库的 CDBC API、C++ APP 技术，以及能够在 Internet 应用中保护数据的安全模式等等，同时还提供了对 RCA (Remote C++ APP)、C++ Servlets API、CSP (C++ Server Pages) 以及 XML 技术的全面支持。其最终目的就是成为一个能够使企业开发者加速产品开发，大幅缩短投放市场时间的体系结构。

8.1. MYCP 的四层结构

MYCP 使用多层的分布式应用模型，应用逻辑按功能划分为组件，各个应用组件根据他们所在的层分布在不同的机器上。MYCP 的多层企业级应用模型把传统两层化模型中的不同层面切分成许多层，一个多层化应用能够为不同的每种服务提供一个独立的层，以下是 MYCP 典型的四层结构：

- 运行在客户端机器上的客户层组件
- 运行在 MYCP 服务器上的 Web 层组件
- 运行在 MYCP 服务器上的业务逻辑层组件
- 运行在 EIS 服务器上的企业信息系统层软件



8.2. MYCP 应用组件

MYCP 应用组件包括客户端应用组件和服务端应用组件；MYCP 客户端应用组件是指能够访问 MYCP 服务器，具有独立功能的软件单元。

MYCP 定义了以下的应用组件：

- 应用客户端程序客户层组件。
- C++ Servlet 和 C++ Server Pages (CSP) 是 web 层组件。
- C++ APP 和 RCA 是业务层组件。
- 其他所有支持 TCP+SOTP 协议的客户端应用程序组件。

所有的 MYCP 组件通过配置，或者编程方式，方便与其他组件交互。

8.3. 客户层组件

MYCP 客户层组件应用程序可以是基于 web 方式的，也可以是基于传统方式的。

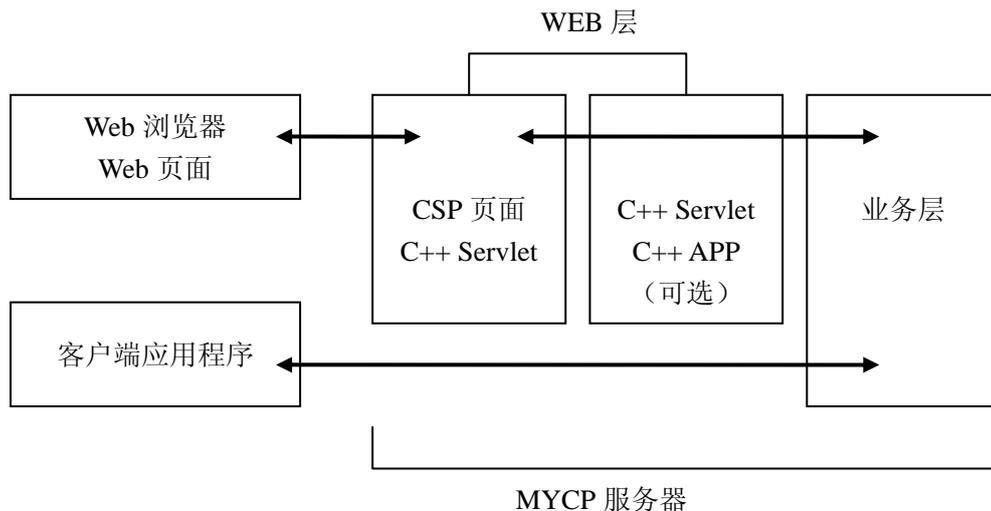
MYCP 基于传统方式的，支持目前所有编程语言或工具（不单单 C/C++ 语言），使用 TCP+SOTP 协议（一种简单对象传输协议）访问，实现更多异构环境客户端

应用。

8.4. web 组件

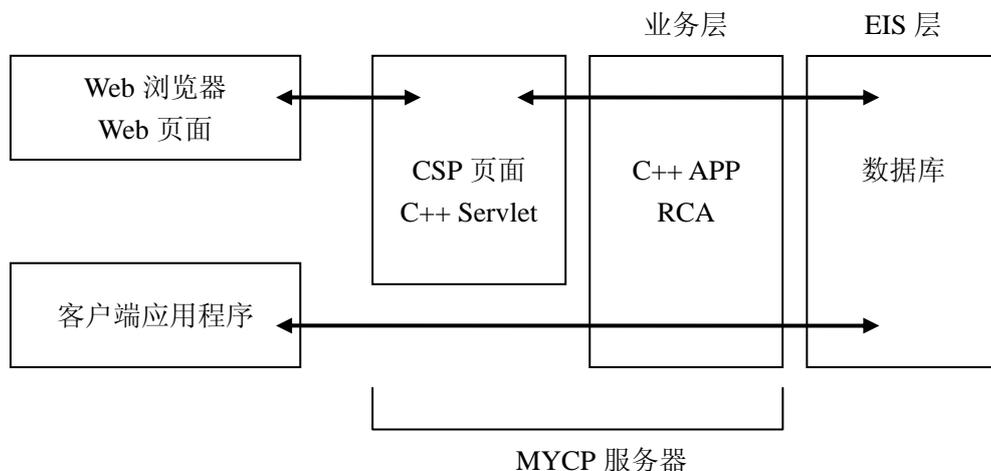
MYCP web 层组件可以是 CSP 页面或者 C++ Servlet。

如下图所示，web 层可能包含某些 C++ APP 对象来处理用户输入，并把业务数据发送给运行在业务层上的 RCA（或者本地其他 C++ APP）进行处理。



8.5. 业务层组件

业务层代码的逻辑用来实现各种业务需要，由运行在业务层上的 RCA 进行处理。下图表明了一个业务层组件的应用模型：



RCA 是一种远程的 C++ APP 应用组件。

8.6. 企业信息系统层

企业信息系统层处理企业信息系统软件，包括企业基础建设系统（例如企业资源计划 ERP、客户关系管理 CRM 等），大型机事务处理，数据库系统，和其它信息系统等。

9、 关键技术点

- 跨平台：

为了实现跨平台，必须使用标准的 C++ 开发库和标准，不能使用类似 MFC 之类的类库，基于功能和性能的综合考虑，选用 boost 开发库协助开发。

主要包括有 shared_ptr 智能指针、asio 网络通讯、thread 多线程等等。

原则上具标准 C++ 编译器的各种硬件、平台、操作系统等环境都支持 MYCP。

- 智能指针：

C++ 开发的最大难度就在于内存管理，所以 MYCP 一开始设计就采用了智能指针技术，从根本上避免内存泄露，提高系统安全性和运行效率。

- 组件化设计：

为了方便开发者开发、调试和部署管理，MYCP 对业务逻辑进行抽象分类、组件化设计，并采用 XML 配置；甚至包括 MYCP 核心组件在内的所有组件，都是可替换更新。

- CSP+容器：

利用 HTTP 解析封装组件，实现 HTTP 的接入，就可以实现静态 HTML 页面输出；通过定义、解析实现 CSP 标签技术，CSP 页面跟 C++ APP 应用组件交互，实现动态 HTML 页面输出；同时配合 CDBC 数据库等其他接口，实现完整 CSP+容器功能。

- 远程组件技术 RCA：

定义并实现 SOTP 协议，比 SOAP 协议简单易用，实现远程组件调用、输入/输出参数，传输数据，传输附件等远程访问技术。

10、 功能模块设计

MYCP 采用组件化设计，下图详细列出 MYCP 各功能模块：

TCP 通讯 CommTcpServer		UDP 通讯 CommUdpServer		RTP 通讯 CommRtpServer		LOG 日志管理组件	
HTTP 解析封装 ParserHttp			SOTP 解析封装 ParserSotp				
组件管理、多线程管理、任务调度 CGCP							
HTTP 服务器、CSP+容器 HttpServer				C++ APP (SOTP) 组件	用户开发的其 他 C++ APP C++ Servlet RCA		
CDBC 组件	C++ APP 组件	C++ Servlet 组件	CSP 页面				

11、 相关软件分析及总结

从整体功能上来讲，目前市场没有类似的软件，MYCP 是第一个集 HTTP 服务器、C++ Web 应用服务器于一身的整套解决方案。