# C#流媒体服务器系统

# 技术白皮书

## 概述

C#流媒体服务器，采用C#语言开发，功能类似FMS，对视频直播、录播、实时通讯提供支持。并能扩展任意协议支持，扩展任意服务器功能。支持服务器集群，支持跨平台部署。技术上参考开源项目crtmpserver以及openRTMFP，并在原有基础上进行了一些改进和优化。

C#语言自身的特点有许多优势，比如编译和调试的效率非常高，代码可读性强，错误显示易于排查，内存的自动管理，并行和异步编程的好处，LINQ和lambda表达式，扩展方法、特性、动态编译、等等。相比C++语言更为繁琐，IDE智能化较弱，很难在复杂代码间切换查找，头文件容易出错，本地编译有很多环境因素容易出错，内存管理容易造成内存泄漏等。Visual studio还为.net框架程序提供许多强大的工具，例如性能分析器，可以查找代码中消耗内存和CPU的代码或函数，以及函数调用次数，调用关系，分配比例等等。Visual studio还有代码图功能，可以用图形展示代码逻辑，代码之间的联系。可以计算代码覆盖率，代码度量值。在重构方面提供非常便捷的操作和提示。总体可以提升非常卓著的编写效率。

## 改进部分

* 融合两大框架，实现rtmfp协议和rtmp协议之间的互通互联，即采用不同协议客户端可以进入相同的房间进行交互，观看视频，发布视频等等。实现方法是通过改造原有rtmfp协议的IO部分适应原有crtmpserver所采用的协议链的框架。
* 网络IO部分采用C#内建的异步socket编程模型，该模型实现了完成端口的socket通讯方式。
* 数据读写部分使用内存流、文件流等统一读写模型。
* 使用C#代码代替引用openssl的功能，使得程序不依赖本地库，更好的跨平台部署。
* 对频繁创建的对象采用池化技术，回收资源，重复使用，提高性能。
* 使用T4模板功能，生成常量代码，提高性能。
* 对框架一些匹配功能和配置功能，采用C#的特性，声明式编程，可读性和可维护性增强。
* 对服务器应用扩展，提供实时编译功能，动态编译C#代码和加载。
* 使用C#索引器改造部分类的表达，代码更直观。
* 使用.net内建的集合类（列表，链表，字典，哈希表），和支持并发访问的集合类优化数据组织和访问。
* 使用LINQ表达式，优化数据查询和遍历访问。
* 增加服务器集群功能。

## 程序架构

Application

配置文件

应用扩展逻辑脚本

应用程序管理器

协议管理器

协议工厂

IO管理器

IO处理器

协议链

协议链

协议处理器（多个）

共享对象管理器

流管理器

## 集群架构

源服务器

边缘服务器

边缘服务器

服务器之前采用私有协议TCP连接，支持断线重连。

共享对象全部同步，文件录制会同时在所有服务器录制，方便点播分流。发布流可以发布到任意一台服务器

