# ICE库性能以及技术选型

一、技术选型

出发点：综合设计系统由多个子模块系统构成，每个子模块是由不用语言环境所开发的，例如JAVA，C++，C#、脚本语言等等。子模块之间的数据传递和通讯需要有某种机制交互并且数据传递是可靠且安全的，这时需要考虑有网络通讯进行数据处理。由于系统有远程计算功能，这时需要考虑用到分布式框架。服务端有可能部署到任何操作系统平台上，这时需要跨平台处理。客户端处理数据时必须是多任务的，这时就需要异步多线程处理。

在数据处理和消息分发上最关心的就是通讯效率问题，但考虑效率问题的同时还需要考虑耦合性、网络负载大、异构性和扩展性等问题。

综上述系统特点考虑因此选择了网络通信引擎（Internet Communications Engine，ICE）。

ICE优点：

1、面向对象的语义

支持面向对象的分布式计算技术。

2、适合在异种环境中使用

客户和服务器可以用不同的编程语言编写，可以运行在不同的操作系统和

机器架构上，并且可以使用不同的网络技术进行通信。

3、支持同步和异步的消息传递

Ice 提供了同步和异步的操作调用和分派。

4、支持多个接口

可以通过给已经存在的Ice 对象增加新的facet，轻松地扩展某个服务器的

功能，而不会破坏已有的、已经部署的客户。

5、线程支持

Ice run time 完全是线程化的，其API 是线程安全的。

6、透明的定位机制

Ice run time 会负责定位对象，并管理底层的传输机制。

7、Ice Storm 的发布/订阅机制

Ice Storm 是一种发布－订阅服务，能够解除客户与服务器的耦合。在本质

上，Ice Storm 充当的是事件分发交换机。发布者把事件发给这个服务，由它发

给订阅者。这样，发布者发布的单个事件就可以发送给多个订阅者。Ice Storm

适用于构建高效的事件转发机制。

8、利用 Glacier 机制，可以方便地实现穿越防火墙

Glacier是Ice防火墙服务：它能让客户与服务器通过防火墙安全地进行通信，

且不牺牲安全性。

9、采用了 TCP/UDP 协议

ICE 提供了一种RPC 协议，既可以把TCP/IP、也可以把UDP 用作底层传

输机制。（可以通过配置一个参数来选择需要的传输机制）。

10、可靠的安全机制

ICE 采用了SSL 加密技术，使得客户机和服务器哪怕是在不安全的网络中，

也可以安全地进行通信。

11、内建的持久机制

使用 Freeze，可以创建持久的对象实现。

12、开放源码(GPL)

Ice 提供了一个开放的平台，用户可以根据需要来开发应用程序。

二、综合设计子系统中ICE性能优化

简要说明：早期项目中各系统子模块之间相互调用大部分使用WebServices，而web Service的主要作用是通过Internet提供远程服务，以XML作为载体。

优点: 兼容性、跨语言、适用性、可扩展性等。

缺点: 解析速度相对缓慢，对内存、带宽消耗大。

从而在数据对象较大、频繁调用的时候就会感觉耗时时间过长、数据传输慢的问题。

综合设计子系统中涉及ICE服务的功能点优化：

1、建模环境中与Procss引擎进行数据交互

比如：解析XPDL信息，存取模型实体信息、UI需要展现的数据信息等。



2、本地客户端与远程ICE服务Process引擎执行交互(异步双向交互)

比如：建完模以后XPDL需要在远端执行查看执行状态，执行进度、执行日志、执行结果等数据信息。

