#### summit架构

**一、summit介绍**

Summit系统是典型的MVC结构系统，其中View层称为SummitFT，基于微软C# .NET技术；Control层分为2部分，一部分为Java开发的通信中间层，另一部分为C/C++编写的Summit主体部分；最后，Model层作为Summit业务数据抽象、存取层，基于ENTITY实现，支持主流的Oracle/SQL Server以及Sybase数据库。下面，我们从前端到后端，对Summit使用的主要技术进行介绍。

**二、summit主要技术**

  SummitFT使用Infragistics的C#控件库作为基础，封装出了一套自己的控件。整个界面风格统一、控件布局合理，操作方便，对用户比较友好。作为对比，Calypso基于Java做的界面；Kondor基于C做的界面，操作体验上来说，跟SummitFT是没法比。

  我们再来说Control层，Summit作为典型的CS程序，客户端与服务端通讯采用的不是TCP/IP直接通讯的方式，而是采用了HTTP协议和WebService的方式。其中，SummitFT通过HTTP协议与通讯中件层通讯；通讯中间层采用WebService与etoolkit进程通讯，达到使用Summit后端服务的目的。这种设计的好处就是Control层不仅可以对接SummitFT，还提供了一套灵活的供其他客户端调用的方式，比如Summit就支持VBA、Java等其他语言的直接调用。由此可以看出，Summit系统在设计时已经考虑到了系统的开放性。通讯中间层采用Java语言编写，负责接收SummitFT的HTTP连接，并负责HTTP协议报文与SOAP报文之间的转换。Summit Business Control层即上文提到的etoolkit，etoolkit使用C/C++开发，实际上就是一个WebService Server，负责处理中间层的请求，并将结果封闭成SOAP报文，返回给通讯中间层。

Summit系统优秀的架构，提供了其技术层面上高可扩展性。一方面，Summit系统的开发工作相对来说较容易，只需要掌握一些API的使用规则即可。大量的精力是花在搞懂业务规则上。另一方面，如果不使用Summit API，自己也可以使用C/C++和其开源库，写出很多好用、性能高的框架，然后套到Summit后端。笔者就写过很多平台类的组件，套到Summit整个框架内。

**三、summit硬件架构**

从硬件架构方面来看，Summit依旧采用的是异构方式，其主CPU来自于IBM Power 9，22核心，主频为3.07GHz，总计使用了103752颗，核心数量达到2282544个。GPU方面搭配了27648块英伟达Tesla V100计算卡，总内存为2736TB，操作系统为RHEL 7.4。从架构角度来看，Summit并没有在超算的底层技术上予以彻底革新，而是通过不断使用先进制程、扩大计算规模来获得更高的性能。

虽然扩大规模是提高超算效能的有效方式，但是为了将这样多的CPU、GPU和相关存储设备有效组合也是一件困难的事情。在这一点上，Summit采用了多级结构。最基本的结构被称为计算节点，众多的计算节点组成了计算机架，多个计算机架再组成Summit超算本身。

1. **节点性能**

Summit的一个完整节点拥有2颗22核心的Power 9处理器，总计44颗物理核心。每颗Power 9处理器的物理核心支持同时执行2个矢量单精度运算。换句话说，每颗核心可以在每个周期执行16次单精度浮点运算。在3.07GHz时，每颗CPU核心的峰值性能可达49.12GFlops。一个节点的CPU双精度峰值性能略低于1.1TFlops，GPU的峰值性能大约是47TFlops。