# 绪 论

## 课题背景及选题意义

随着4G、Wi-Fi等移动通信技术的飞速发展，移动互联网行业的时代已经到来。作为用户进入互联网的重要入口之一，手机应用程序在移动互联网中已经占据了举足轻重的位置。根据iiMedia Research（艾媒咨询）2016年7月29日发布的《2016上半年中国手机桌面行业研究报告》显示截止2016年第二季度，中国智能手机用户已达6.308亿人。庞大的用户基础推动了手机应用程序的快速发展。手机应用程序承载了各种便捷的移动服务，已经成为人们日常生活的一部分。手机应用程序的性能监控与应急处理已经受到移动互联网行业的重视。

为了更好的服务用户，了解用户的访问特征及用户需求，了解手机应用程序的运行情况以及性能情况手机应用厂商迫切需要对用户的使用情况进行分析研究。现在各大手机应用程序公司的普遍做法是记录用户使用情况，记录产品运行情况，然后根据产品运行数据和用户使用行为数据进行分析，进而改进产品，给用户带来更好的体验。应用程序新版本在发布之前的性能测试也已经成为另一个非常重要的工作，通过产品的总体性能测试数据和各个模块分别的性能测试数据，开发人员可以快速发现并定位产品中存在的问题和相关模块的性能瓶颈。另一方面，产品性能的测试数据也是衡量产品是否可以发布的重要指标。

产品日志是手机应用软件在运行过程中对自身运行状态进行实时记录的文件，通常这些文件还包括系统的所有操作、软件自身行为以及用户的行为。这些数据通常用于软件故障排查和软件性能分析，在数据量不多时，日志的处理和分析由经验丰富的产品开发工程师人工查找完成，但是随着系统规模的扩大和用户数量的增加，日志的数量正以前所未有的速度快速增长，这种情况下继续采用人工的方式进行处理将是一件不可能完成的任务。因此，为了高效的处理大规模日志，解决单机系统上的存储和处理瓶颈，提高处理效率，各大公司都在开发研究分布式日志管理系统和相应的软件性能分析系统。

使用软件性能分析系统可以对大量的日志数据进行智能分析，自动计算产品性能，方便诊断和优化产品的缺陷。软件性能分析系统可以量化的衡量整个产品的质量，通过分析产品中各个模块的数据，可以直观得到每个模块的性能数据进而分析出产品的性能短板；通过分析产品不同流程不同模块的数据，可以更快的发现和定位产品的问题，使开发工程师有的放矢；也可以分析用户数据，分析用户使用习惯，分析用户喜好，对用户进行分类，并针对每种不同类型的用户推送其喜欢的话题和功能，这样可以为下一步产品的改进提供建设性的意见。

因此，在微软小娜产品发布之初，我们开发了微软小娜性能分析系统。目的是采用大数据处理的方法更高效更便捷的分析产品日志，为微软小娜产品的发布和新功能的开发提供可以量化的性能指标，并且将这些指标实时的以可视化的方法展示给开发人员。

## 国内外技术现状

随着互联网的迅猛发展，如今的互联网企业每天需要处理数以万计甚至百万千万的用户访问请求。随着系统规模的扩大和用户数量的增加，用户的日志数量更是成几何倍数增加，这种高并发大数据的日志发送接收和处理需要用到分布式软件系统(Distributed Software Systems)。分布式软件系统是在由通信网络互联的多处理机体系结构上支持分布式处理任务的软件系统。分布式软件系统包括分布式操作系统、分布式程序设计语言及其编译系统、分布式文件系统和分布式数据库系统等。在日志收集中主要使用的是分布式文件系统，分布式文件系统主要解决了数据的存储和管理问题：将原本固定在某一个地点的某一个文件系统扩展到任意多个地点或多个文件系统，将这些节点组成一个方便统一管理的文件系统网络。每个节点可以分布在不同的地点，节点间通过网络进行通信和数据传输[3]。在使用分布式文件系统时，不需要关心数据是存储在哪个节点哪个地点上、或者是从哪个节点那个地点获取的，只需要和使用本地文件系统一样管理和存储文件系统中的数据。

为了方便快速收集处理大规模并发数据，分布式日志收集系统已经得到了广泛应用。其中，Scribe是Facebook公司的开源日志收集处理系统，主要采用“分布式收集，统一处理”的可扩展高容错的方案，从各个日志源上收集日志并将日志存储到分布式文件系统上然后再统一处理。Chukwa是Apache公司的一个开源的大型分布式系统主要用于日志的收集和分析[4]。Chukwa是构建在Hadoop的HDFS和MapReduce框架之上的数据采集分析框架，其继承了Hadoop的可伸缩性和鲁棒性[5]。该框架提供采集数据的客户端（Client），由客户端发送数据给服务器（Server），然后通过定期运行脚本分析数据。Flume是由Cloudera公司提供的一个分布式可靠的海量日志采集、聚合和传输系统[6]。Flume支持在日志系统中定制各类数据发送方用于收集数据，同时对数据进行简单处理并且可以输出到各种数据接收方。Aria是微软（Microsoft）公司开发的Cosmos之上的日志收集系统，作为共享数据平台的一部分，Aria可以支持平均每秒5百万次日志收集。将日志导入Cosmos之后，采用Scope脚本更加详细的计算所需数据。Aria支持实时多维的大数据分析，用户可以创建展示页面，展示多维立体数据并且可以实时查看原始日志。

在大型软件系统中，由于系统环境设置和软件规模等因素，错误（bug）重现和调试都十分困难。百万用户级别的访问量每天会产生上百乃至上千兆字节大小的日志，为了能够高效快速的处理大量日志分布式大数据处理平台广泛应用于日志处理系统。微软主要采用Scope脚本语言对大规模用户日志进行处理。Scope语言是运行在Cosmos上的查询语言。Cosmos是微软研发的大数据处理平台。Cosmos主要有以下两大基本功能：（1）使用分布式文件系统CFS(Cosmos File System)存储大量数据的功能；（2）使用Scope语言处理大数据的功能。Cosmos的只要特点如下：（1）大规模数据处理能力，Cosmos支持PB级别的数据处理，拥有大量可信赖的计算机集群；（2）Cosmos可以使处理大数据更加容易，Scope语言简单易学，允许我们在不懂得MapReduce细节的情况下，使用简单SQL和C#知识就可以进行大数据的运算处理；（3）Cosmos也是一个共享数据系统 (Shared Data Ecosystem)，在工作中不同的组可以在系统提供的平台上相互共享数据。

SQL Server是微软推出的关系型数据库，是一个可扩展的、高性能的、为分布式客户机/服务器计算所设计的数据库管理系统，实现了与Windows NT的有机结合，提供了基于事务的企业级信息管理系统方案。它最初是由Microsoft、Sybase和Ashton-Tate三家公司共同开发的，并于1988年推出了第一个OS/2版本。Microsoft SQL Server近年来不断更新版本，1996年，Microsoft 推出了SQL Server 6.5版本；1998年，SQL Server 7.0版本和用户见面；SQL Server 2000是Microsoft公司于2000年推出，目前最新版本是2012年3月份推出的SQL SERVER 2012[7]。

本系统网站是采用IIS服务部署的，IIS（Internet Information Services，互联网信息服务）是由微软公司提供的基于Microsoft Windows操作系统的互联网基本服务。IIS是一种网页服务组件，其中包括FTP服务器、NNTP服务器、HTTP/HTTPS服务器、Web服务器和SMTP服务器，分别用于文件传输、新闻服务、超文本传输服务、网页浏览和邮件发送服务。

## 本文研究内容

通过课题背景和国内外研究现状可以看到，我们必须要研发一套性能分析系统对微软小娜产品所产生的日志数据进行智能分析、计算性能、诊断产品和优化产品。本文的目的就是开发一套针对微软小娜产品的日志性能分析系统，该系统通过使用微软公司的Aria分布式日志收集系统和Cosmos分布式文件系统来收集和存储日志，系统使用Scope脚本语言计算产品性能，系统使用SQL Server数据库存储产品性能数据，使用ASP.NET框架开发网站，使用IIS部署网站。小娜性能分析系统包括从用户使用产品产生日志开始，经过Aria收集日志并将日志导入到Cosmos分布式文件系统中，再使用Scope语言计算产品性能，最后将性能数据展示在网页上的整个日志采集及分析过程。本文的研究内容将从以下几个方面进行入手：

（1）Aria平台上日志的收集：日志收集主要通过Aria SDK实现，通过Tenant Token将日志数据与微软小娜产品关联起来，日志的内容除基本内容外扩展数据通过Key-Value的方式插入；在此方面还包含日志的设计，为了使日志计算更加合理使日志扩展性更强，设计了一系列措施保证数据的完整性、准确性和可扩展性。

（2）Cosmos平台上日志的计算：本系统采用Scope脚本语言进行性能的计算，主要包括各个模块的成功率和延迟时间。通过计算模块开始日志和结束日志之间的时间差计算模块延迟时间；计算模块结束日志数量占模块开始日志数量比例计算产品模块的成功率。其中，模块延迟时间不是简单的计算每个用户延迟时间的平均值，而是采用求分位数的方法将每个用户的延迟时间从小到大排序，取第75个、第90个、第95个百分位数作为模块的延迟时间。

（3）自动化工具实现：为了能自动的提交脚本并将计算结果导入到数据库中，本系统实现了一个定时提交脚本并向数据库导入数据的工具，并且该工具可以对提交脚本和导入数据过程中可能出现的所有异常进行了相应处理，保证工具的健壮性和灵活性。

（4）数据库和网站实现：微软小娜性能分析系统使用SQL Server数据库存储产品性能数据；采用ASP.NET MVC框架实现展示网站，将数据库中数据以可视化的图表和曲线图的方式展示出来，这样可以直观地看到产品各个功能模块的性能参数。

## 论文的组织安排

# 关键技术介绍

采用合理的软件开发框架和技术可以使微软小娜性能分析系统结构合理且具有很高的鲁棒性和可扩展性。本章主要介绍开发微软小娜分析系统时使用的几个关键技术框架。

1. Cosmos平台和Scope脚本语言介绍

Cosmos平台是微软为存储和处理大规模数据而开发的，主要包括两大基本的功能：Cosmos文件系统（CFS）是一个具有存储大规模数据能力的分布式文件系统（Distributed File System），可以支持PB级别数据计算；使用Cosmos平台提供的Scope脚本语言可以处理存储在Cosmos文件系统中的数据，Scope脚本语言可以使我们非常容易的处理大规模数据，只需要基本的SQL和C#语法就可以使用Scope语言，而且现在我们可以使用Visual Studio方便的开发Scope程序。Cosmos平台的大规模数据存储和处理能力主要体现在以下几个方面：

（1）Cosmos平台大约有11万台处理器分布在4个不同的物理集群上；

（2）Cosmos文件系统存储了1.2EB的数据（）；

（3）Cosmos平台每天可以运行10万多个Scope程序。

Cosmos也是一个数据共享系统（Shared Data Ecosystem），不同的团队可以通过Cosmos平台共享数据集。

Scope是Cosmos平台的查询脚本语言，它的许多基本概念和句法表达式都和SQL类似，这样使得程序简单易写，而且Scope还可以调用C#函数，这使得Scope更加的灵活和强大。在Visual Studio中，用户可以使用Scope的开发插件方便快捷的编写程序和提交到Cosmos平台上运行。

1. ASP.NET MVC框架

ASP.NET是.NET Framework的一部分，它可以使嵌入网页中的脚本在因特网服务器短执行，它也可以动态的在Web服务器上被创建。ASP .NET的前身是ASP（Active Server Pages）技术，与ADO （ActiveX Data Objects）1.0在IIS2.0上首次推出，在IIS 3.0 时发扬光大，成为服务器端应用程序的热门开发工具。ASP.NET支持三种不同的开发模式：Web 页面（Web Page）、模型-视图-控制器（MVC）和Web 窗体（Web Forms）。

MVC框架将一个交互式应用程序分成3个组件：模型（Model）包含核心功能和数据，表示应用程序核心，是应用程序中用于处理应用程序数据逻辑的部分；视图（View）负责向用户显示信息，是应用程序中负责数据显示的部分；控制器处理用户输入，是应用程序中处理用户交互的部分。

模型封装了内部的功能和数据，对于用户来说是不可见的（模型和视图相互独立），模型独立于特定的输出表示或者输入方式（模型与控制器独立），用户只能通过控制器操作模型（控制器是模型和视图的桥梁）。

视图向用户显示信息，不同的视图使用不同的方法呈现信息，每个视图组件都有一个更新函数，这个函数被模型变更通知激活，将使得视图和模型一致。视图通过状态查询函数从模型获取数据，如定时刷新。

每个视图都有一个相关联的控制器组件，它们是一一对应的关系。控制器组件接受事件，并翻译成对模型或者视图的请求，事件是如何发送到控制器由用户界面平台决定。如果控制器的行为依赖于模型的试图，那么控制器也需要向模型登记请求变更通知。用户仅仅通过控制器与系统交互。

1. 通用 Windows 平台应用（UWP）

通用 Windows 平台（UWP，Universal Windows Platform）应用是一种基于通用Windows平台生成的可在所有Windows上运行的应用，它在Windows 8操作系统中被首次引入。UWP的核心是使用户在所有设备上的体验相同，并且希望使用现有的最方便最高效的设备完成任务。

通过Windows 10操作系统，用户可以更加轻松地开发UWP应用，只需一个API集合、一个应用和一个应用商店，即可访问所有Windows 10设备包括电脑、平板、手机、Xbox、HoloLens等。对各种屏幕大小以及各种交互模型（无论是触控、鼠标、键盘还是游戏控制器）的支持也更加方便。UWP应用的主要特性如下：

（1）开发人员的目标设备是整个Windows系列，而非某个操作系统。设备系列可标识在其中的设备上所需的API、系统特性和行为。它还可以确定在其上安装应用商店应用的设备集。

（2）UWP应用使用AppX格式进行打包和部署。这提供了值得信赖的安装机制，并且确保应用可以无缝进行部署和更新。

（3）应用商店中的应用程序适用于所有Windows设备。应用开发人员可以向应用商店提交应用，并使其在所有设备系列或仅在所选设备系列上可用。应用开发人员将在一个位置上提交和管理适用于所有Windows设备的应用。

（4）在整个Windows系列上有一套相同的应用程序编程接口（API）。所有Windows设备系列均使用相同的API，这样开发出来的应用将可以在任何Windows 10设备上运行。

（5）在特殊的设备上使用扩展SDK可以为应用增添色彩。扩展SDK可以为每个设备系列添加专用API，如果应用面向特定的设备系列，开发人员就可以使用这些API。在调用扩展API之前需要检查应用在什么设备系列上运行，开发人员仍然可以开发一个可以在所有设备上运行的应用程序。

（6）UWP应用具有自适应输入控件，它们与多种输入类型（如键盘、鼠标、触摸、笔等）配合良好。UWP应用的UI元素使用有效像素，它们会基于设备的屏幕像素数自动自行调整。如果开发人员需要进一步为特定屏幕大小或设备定制UI，布局面板和工具将帮助开发人员开发出适应运行设备的UI。

1. 数据库连接池

网站使用C#数据库连接池来管理维护数据库的连接。建立一个数据库连接是一件非常耗时耗力（消耗时间消耗资源）的事情，连接到数据库服务器通常由几个需要较长时间的步骤组成: 建立物理通道（例如套接字或命名管道）；与服务器进行初次连接；服务器分析连接字符串信息；服务器对连接进行身份验证等。当应用程序访问次数数量很大的时候，许多相同的连接将反复地被打开和关闭，从而会引起数据库服务器效率低下甚至引发程序崩溃，又因为大部份的应用程序都是使用一个或几个固定的连接配置，因此我们可以在ADO.NET中使用称为数据库连接池的优化方法来管理维护数据库连接。

数据库连接池存放了一定数量的与数据库服务器相连的物理连接。当程序需要连接数据库服务器的时候，数据库连接池就会检查池中是否有可用的空闲连接，如果发现有空闲连接可用，会将该连接返回给调用者，而不是创建新的连接；当应用程序需要关闭该连接时，数据库连接池会判断该连接是否在最小连接数之内，如果是的话会将连接回收到数据库连接池中而不是真正关闭连接，否则将销毁连接。这样就可以大大减少连接数据库的开销，从而提高网站的性能。高效的使用数据库连接池一般需要遵循以下几个原则：

（1）在最晚的时刻申请数据库连接，在最早的时候释放数据库连接；

（2）关闭连接时先关闭相关用应用程序定义的事务；

（3）确保并维持数据库连接池中至少有一个打开的连接；

（4）尽力避免池碎片的产生，包括集成安全性产生的池碎片以及使用许多数据库产生的池碎片。

1. 本章小结

本章主要介绍微软小娜性能分析系统的设计开发实现过程中，所使用的关键技术框架及其特点，包括Cosmos平台和Scope脚本语言、ASP.NET MVC框架、通用Windows平台应用、数据库连接池等。采用Cosmos平台和Scope脚本语言能够方便高效的计算大规模日志数据；采用ASP.NET MVC框架搭建网站可以使程序容易维护，增强程序可扩展性，能够进行单元测试，保证功能的实现；采用通用Windows平台应用可以开发一套适用所有Windows平台的应用程序；采用数据库连接池可以大大减少网站连接数据库时引起的开销，从而提高网站的性能。接下来对微软小娜性能分析系统的需求进行分析。

# 微软小娜性能分析系统需求分析

详细梳理微软小娜性能分析系统的内部流程，能够全面地指出微软小娜性能分析系统的具体需求；有针对性地梳理系统的业务处理模块，能够归纳总结出系统的功能性需求，基于对系统性能提升的考虑，可以对非功能性需求提出要求。根据微软小娜性能分析系统的主要功能将该系统分为4个模块包括Aria上日志的收集、Cosmos上日志的计算、自动化工具实现、数据库和网站实现，下面将详细针对每一模块实现其需求分析。

1. 日志收集需求分析

微软小娜性能分析系统的第一步就是获取用户操作数据和产品运行数据，获取产品客户端的日志，即日志的收集。当用户操作产品时，开发人员希望记录产品的版本号，产品的应用平台，用户操作的具体功能，用户操作的准确时间，用户所在地区，用户使用的语言以及区分日志的编码字段等关键信息。根据日志收集的需求设计日志结构如下表所示：

|  |  |
| --- | --- |
| 属性 | 值 |
| Event Name | CoA App Load |
| Impression ID | 17b84a2c-e4e7-4925-8ab3-9c24d5ebd368 |
| App Version | 1.0.0.662-release |
| OS Version | Android 4.4.4 |
| Time Stamp | 188469142 |
| Language | En-US |
| Key | Value |
| … | … |
| Key | Value |

其中每个字段的含义如下：

Event Name:该字段主要为区分产品中不同模块不同功能的日志，目的是使日志的计算更加方便。产品初始加载日志的Event Name可以设为CoA App Load，闹钟模块日志该字段可为CoA Alarm。

Impression ID:该字段为日志序列号，主要是为区分不同用户不同操作的日志数据，便于性能数据的计算，如果两条日志的Impression ID相同那么就表示这两条日志发生在同一客户端。例如要计算产品加载的时间延迟就需要在数据集中获得Impression ID相同的产品加载开始日志与加载结束日志，通过这两条日志的时间差就可以计算出该用户使用产品时产品加载的时间。Impression ID的编码可以采用UUID（Universally Unique Identifier）标准，其目的是使所有日志都有唯一辨识的资讯，UUID标准可以保证编码的唯一性。

App Version：该字段记录当前操作发生时产品的版本号，主要目的是区分不同版本产品性能，进行纵向对比。版本号的具体规定为正式版本以-release结尾，测试版以-dev结尾，微软小娜产品的版本号码由4个十进制数组成，数之间以点号分隔，如1.0.0.662-release。

OS Version：该字段记录当前操作发生时手机系统的版本号，如Android 4.4.4，主要目的是对比不同手机系统之间的性能差异，使开发人员可以针对某一操作系统进行测试优化。

Time Stamp：该字段记录当前日志的生成时间，用来计算产品模块或功能的延迟，精确到毫秒。在此Time Stamp是指格林尼治时间1970年01月01日00时00分00秒起至现在的总毫秒数。

Language：该字段存放用户使用的语言和所在地区，主要的字段值为：En-US（英语-美国）、En-IN（英语-印度）、Zh-CN（汉语-中国）、Ja-JP（日语-日本）。

Key-Value：该字段存放开发人员为各个模块定制的参数，这些数据以Key-Value的方式存放，可以细分产品的子功能，更加精确的计算各个子功能的性能参数。

整理好日志的基本格式之后，需要对日志的发送和收集储存进行分析。用户的日志收集采用Aria分布式日志收集平台实现，通过在客户端加入Aria SDK，客户端就可以主动发送用户日志到Aria服务器，然后Aria服务器将各个产品的日志数据进行整理，并把微软小娜的日志数据储存在Cosmos分布式文件系统中供下一步计算使用。

1. 日志计算需求分析

在整个性能分析系统中，最关键最重要的模块就是日志计算模块，因为日志计算的准确度直接关系到性能数据的准确度，是整个项目的核心模块。

微软小娜性能分析系统主要负责分析客户端的数据，需要计算不同版本号、不同地区、不同网络环境、不同操作系统、不同功能模块的性能数据，并对其进行纵向比较，因此，在日志计算的过程中需要对以上需求进行分别计算。其中各个需求的划分方法分别为：版本号为每次产品发布时的版本号，具体规定为正式发布版本以-release结尾，开发测试版以-dev结尾，各个功能模块的测试版以-rd结尾，版本号码由4个十进制数限定，数之间以点号分隔，如1.0.0.662，随着发布版本增加版本号会逐渐增大；不同的网络主要是指网络运营商和网络制式的区分，在中国大陆地区网络运行商主要有中国移动、中国联通和中国电信，在美国和欧洲网络运营商主要有AT&T、Sprint、T-Mobile、Verizon Wireless、MetroPCS和Cricket Communications，网络制式主要有HSPA+、LTE、HSDPA、HSPA、UMTS、CDMA-eHRPD、TD-SCDMA、EDGE、GPRS，计算性能数据时，系统会根据网络制式将其划分为4G、3G、2G以及Wi-Fi网络，以便于计算；因为各个地区的语言和习惯不同，微软小娜产品会根据地区不同对产品做出一些功能上的调整，因此需要对不同的地区分别计算产品性能，现在需要计算地区有美国、印度、日本和中国4个地方；在操作系统上系统主要区分iOS和Android，后面可能会有WP等其他操作系统的需求。

微软小娜性能分析系统有两大主要性能指标：成功率和延迟时间。成功率是指每个功能模块在没异常情况下正常运行到结束的概率；延迟时间是指产品的每个功能模块从开始加载到结束的时间，需要精确到毫秒，最关键的是功能模块的延迟时间不应该简单是每个用户延迟时间的平均值，而应该采用求分位数的方法将每个用户的延迟时间从小到大排序，取第75个、第90个和第95个百分位数作为模块的延迟时间，因为求平均值的方法可能会引入较大误差，使用这种计算方法可以降低延迟时间的误差（假设第75个半分位数的延迟时间为2000毫秒，表示有75%的用户的延迟时间小于2000毫秒）。根据产品的特点、产品的功能分析，需要计算的场景举例如下：

SR（Speech Recognition）：语音识别场景，指从用户点击麦克风录音开始到语音识别模块识别完全部文字的过程，需要计算语音识别模块的成功率及其延迟时间，如果用户主动取消了语音识别操作，也认为语音识别模块失败。

LU（Language Understanding）：语义理解模块，指从语音识别完全部文字到客户端收到HTTP回复语义理解结果的过程，需要计算语义理解模块的成功率以及延迟时间，如果用户主动取消了语义理解操作，也认为语义理解模块失败。

Proactive Page Load：客户端主页面加载场景，指从发出加载页面请求开始到主页面被全部加载完成的过程，需要计算其成功率以及延迟时间，如果用户主动取消了页面加载操作，也认为页面加载模块失败。

App Load：产品启动，指从用户在桌面点击小娜图标开始到用户可以使用麦克风或搜索框搜索的过程，需要计算其成功率以及延迟时间，如果用户主动取消了产品启动操作，也认为产品启动模块失败。

DAU（Daily Active User）：指微软小娜每天的活跃用户量，需要计算每天微软小娜产品总体及各个功能、各个模块的分别活跃用户量，也包括每日新用户数量已经新用户次日存留用户量。

1. 自动化工具需求分析

实现自动化工具的目的是自动化工具可以每天定时提交脚本并自动将已经计算的性能数据导入到数据库中，因此该工具主要有定时提交脚本和将性能数据导入到数据库这两大需求。Cosmos分布式文件系统中的日志数据会有延迟时间，一般的延迟时间为2天，但是也可能会有更多天数的延迟，因此自动化工具需要记录两个日期，分别是已经将脚本提交的日期，和已经将性能数据导入到数据库的日期。这样做的目的是防止重复计算降低系统性能，防止重复数据导入数据库。系统每次运行之前，需要检查之前脚本的提交日志和导入数据的日志，以便重新提交延迟的产品日志。此外还可能会有某一天或一段时间的性能数据因为日志问题需要重新计算，自动化工具需要有对指定日期或指定时间段的数据进行重新计算，并把已经导入数据库的数据删除。

自动化工具应该可以简单的被应用于其他程序，因此需要有动态配置脚本优先级、脚本路径、数据下载路径以及数据库名称以及表名称的功能。除此之外，自动化工具还应该每天定时自动运行，这样可以最大程度上节约开发人员的时间。最重要的是自动化工具需要具有异常处理、上下文恢复的功能，因为在提交脚本时可能会由于超时，脚本编译错误，源文件源目录不存在等问题出现异常，这时应该把所有情况恢复到工具未运行的情况，并需要记录异常日志，特别的，当数据导入数据库发生异常时，自动化工具需要将已经导入的部分数据删除，并记录异常日志。当有异常发生时，自动化工具应该有自动提醒开发人员的功能，可以使开发人员快速收到异常日志或错误记录，以便快速发现问题解决问题。

1. 数据库和和网站的需求分析

计算出的产品性能数据都会被保存在数据库中以供网站读取，因此在数据库的需求分析阶段需要确定基本的表结构，数据库主要存储的数据可以分为三类：功能的成功率、功能的延迟时间和功能的使用人数。

在数据库保存产品功能的成功率时，需要存储日志的产生时间（精确到天），功能的名称，功能的子名称，产品的版本号，手机的网络状况，手机的系统版本号，用户的语言和地区，以及功能的使用次数，功能被成功使用的次数和功能的成功率。在讨论延迟时间的需求时需要定义几个指标PLT75、PLT90、PLT95，PLT75、PLT90、PLT95的含义是假设PLT75等于2000毫秒，表示有75%的用户的延迟时间小于2000毫秒，其计算方法是将每个用户的延迟时间从小到大排序，取第75个、第90个和第95个百分位数即为PLT75、PLT90和PLT95的值。在数据库保存产品功能的延迟时间时，需要存储日志的产生时间（精确到天），功能的名称，功能的子名称，产品的版本号，手机的网络状况，手机的系统版本号，用户的语言和地区，功能的使用次数以及PLT75、PLT90和PLT95的值。在存储功能的使用人数时，需要存储日志的产生时间（精确到天），产品的使用人数，产品当天的新用户，产品昨天的新用户在今天的使用人数，除此之外还要存储各个功能模块的使用人数，此功能当天的新用户，此功能昨天的新用户在今天的使用人数。

网站功能性需求为：用户可以查询每天的产品性能数据，用户可以选择不同的网络数据（Wi-Fi/4G/3G/2G）、不同的产品发布市场（Zh-CN/En-US/En-IN/Ja-JP）、不同的产品系统平台（Android/iOS）查看相关功能模块的性能数据，用户可以指定功能模块查看其几个月之内的性能变化趋势图。根据数据库存储的三类数据：功能的成功率、功能的延迟时间和功能的活跃用户量。网页需要划分为三个区域，分别展示功能模块的成功率、延迟时间和活跃用户量。网站需要根据成功率和延迟的数据大小对数据标注不同的颜色，以求更直观的看出产品各个功能模块的性能，具体标准为当成功率高于99%时标为绿色，当成功率高于90%时标为黄色，当成功率低于90%时标为红色；当延迟时间小于1500毫秒时标为绿色，当延迟时间小于2500毫秒时标为黄色，当延迟时间大于2500毫秒时标为红色。绿色数据表示该功能模块运行良好，黄色数据表示该功能模块性能需要注意，红色表示该模块性能出现较大问题，当某个模块的性能参数出现较大问题时，网站应该又给开发该功能的开发人员邮件提醒的功能。为了纵向比较产品的性能，当用户点击任何一个数据时，网站需要展示相同功能，相同的语言和地区，相同的手机系统，相同的网络状况下此数据两个月以来变化的趋势图。对于每个大功能模块的成功率和延迟，需要画出饼状图展示每个子功能模块的成功率和延迟时间；对于每个功能模块的失败原因进行分析，统计不同失败原因导致功能失败的次数，并画出饼状图。网站的用例图如下：



1. 非功能性需求

微软小娜性能分析系统的非功能性需求主要包括两个方面：性能需求和扩展性需求。因为开发人员要关注当前版本产品的性能情况，因此需要整个分析系统应支持小时不间断工作；如果发现产品性能下降，开发人员需要查看用户的原始日志发现问题，因此用户的原始日志应该永久保存；分布式文件系统中产品的性能数据应保存3个月；数据库中产品的性能数据应永久保存；可扩展性指性能分析系统应支持业务功能的扩展，当增加模块或减少模块时，性能分析系统应该自动计算新增加的模块，并在网页上自动增加相关表格支持新模块的性能数据展示，这样能有效的减少后续系统开发和投入成本。

1. 本章小结

本章从微软小娜性能分析系统的业务流程入手，介绍了性能分析数据的详细流程，引出了微软小娜性能分析系统在计算和展示产品的性能数据时所要解决的问题，并给出了具体解决方案。在功能性需求中，从Aria上日志的收集、Cosmos上日志的计算、自动化工具实现、数据库和网站实现四个功能模块入手，对业务的需求进行详细剖析；在非功能性需求中，对微软小娜性能分析系统的性能需求和扩展性需求进行详细的说明。详细的需求分析是微软小娜性能分析系统的基础环节，保证设计有序展开及最终的实现结果能够满足要求。

# 微软小娜性能分析系统总体设计

微软小娜性能分析系统总体设计的主要任务是把需求分析得到的系统功能性和非功能性需求换为软件结构和数据结构。本章将微软小娜性能分析系统按功能进行模块划分、建立模块的层次结构及调用关系、确定模块间的接口及网站界面等。除此之外本章还会对数据结构进行设计，包括性能数据特征的描述、确定数据的组成结构、以及数据库的设计。下面将详细介绍微软小娜性能分析系统的总体设计。

1. 设计目标

微软小娜性能分析系统是为了更有效的实现快速化、规范化、流程化的分析微软小娜产品性能而建立的分析系统。通过微软小娜性能分析系统，可以有效的收集存储大规模产品日志数据，快速有效的计算产品性能以及更加直观的看出产品性能数据，这样可以极大的提高性能分析工作的效率，提升微软小娜产品的研发能力和综合实力。

微软小娜性能分析系统各个子模块间应该能够相互独立的工作也可以动态的进行组合，根据不同的使用场景系统可以提供不同的操作项。这样既可以增强系统的可维护性，又增加了系统的可扩展性。系统要达到的目标就是提供一个可靠、稳定、快速获得产品性能数据的性能分析系统，为了达到这个目标，制定如下设计目标：

（1）使用模块化、标准化和参数化的设计标准，保证系统的灵活性、实用性、稳定性、开放性、先进性和可扩展性，使任何新的业务需求和产品需求都可以快速实现；

（2）微软小娜性能分析系统需提供数据统计和决策支持的功能，帮助开发人员快速获得产品性能参数和市场投放效果，并根据用户行为用户习惯完善产品的功能；

（3）微软小娜性能分析系统应该具有一定的容错能力，在日志数据出现异常的情况下也不会崩溃，而是可以出现合理的解决提示。

1. 设计原则

为了实现一个可靠、稳定、可以快速获得产品性能数据的性能分析系统，特制定如下设计原则：

（1）可靠性原则：对于性能分析系统来说，数据源的准确性和完整性是整个系统的基础，对于整个性能分析系统至关重要，这涉及到后面的计算环节，如果用户日志数据不准确、不完整将直接影响到产品性能数据的准确性，影响到整个系统的可信度，这就要求日志的收集和存储需要有高度的可靠性；

（2）健壮性原则：系统需保证无论在何种优化和调试情况下，无论发生什么异常情况，都要保证系统的稳定运行，获得的数据必须稳定、可靠、可信赖，要保证数据库数据的准确性，所有存储在数据库中的数据必须是准确的，可以作为评估产品指标的；

（3）先进性原则：先进性需要系统能够最大程度适应业务需求变化、产品需求变化以及技术发展变化，因此，在设计系统时需要采用先进的设计模式和开放的技术框架，使用先进的管理方式，保证系统的先进性；

（4）可扩展性原则：在设计系统时需要采用参数化、规范化的配置方式，模块化的设计理念使系统的各项指标都可以灵活设置，使系统具有良好的可移植性，另外数据存储结构的设计也需要满足先进、可扩展、易维护的要求。可扩展性是评价一个系统性能优劣的重要指标之一，良好的可扩展性必然可以提高系统的可维护性，这是软件项目开发过程中很重要的部分。

1. 性能分析系统业务逻辑和功能设计

微软小娜性能分析系统涉及到众多子模块的交互，Aria收集日志平台与Cosmos分布式文件系统存储模块的交互，Cosmos分布式文件系统存储模块与自动化运行脚本工具的交互，自动化工具与数据库模块的交互，网站与数据库的交互。主要的几个子系统模块业务逻辑如下：

（1）日志收集存储：收集用户日志、整理用户日志并将用户日志数据保存在Cosmos分布式文件系统中；

（2）日志计算：根据业务需求和产品需求计算日志数据获得产品的性能参数，主要包括功能模块的成功率，功能模块的延迟时间和功能模块的活跃用户量；

（3）自动化工具：自动化工具需要每天定时提交脚本并计算日志数据，然后将计算的产品性能数据保存在数据库中；

（4）网站：网站将保存在数据库中的性能数据展示在网页上，根据不同的业务需求以不同的方式展示不同的数据。

为了提高微软小娜性能分析系统功能的扩展性，在系统以功能划分的基础上，微软小娜性能分析系统总体设计采用平台化设计思想。微软小娜性能分析系统的活动图如下。



1. 性能分析系统架构设计

在对微软小娜性能分析系统进行架构设计时，本文采用模块化的架构设计方法，将性能分析系统主要分为以下几个模块：日志计算模块、自动化工具模块、是数据展示模块。最主要的几个子模块架构设计如下：

（1）日志计算：本模块采用Scope + .NET框架实现，采用面向过程的实现方法，分析出解决问题所需要的步骤，然后用函数把这些步骤一步一步实现，使用的时候一个一个依次调用。

（2）自动化工具：本模块采用.NET + XML+ [Multithread](https://www.baidu.com/link?url=8DKEiKfGoKkVZmhQXoe9WycO1t475c9EPfUKesaQgSFJfFb7-GmVEb6CgD95dzc837gBZRpAAAj7BrrGpnzHg4YqY7wvrkGvX77GhrrZ3-C7scBAXnrmVKLEwnFR5FCj&wd=&eqid=d9884b5c00009d950000000357a485ad)框架实现，自动化工具对每一个单独的任务都起一个线程单独执行，因为每次提交脚本执行都会耗费较长时间，采用多线程框架可以有效的节约时间，可以使程序的响应速度更快；除此之外还将脚本参数写在XML文件中，当添加脚本删除或脚本更改路径时，只需要更改XML文件而不需要重新部署，可以显著提升模块的可扩展性；

（3）数据展示：本模块采用ASP .NET + MVC框架实现，模型包含核心功能和数据，是数据展示模块中用于处理应用程序数据逻辑的部分；视图部分即网站页面负责向用户显示微软小娜功能模块的性能数据；控制器处理处理用户交互，根据用户输入展示出不同的性能。

结合微软小娜性能分析系统采用的技术架构，要完成一个优秀合理的性能分析系统本系统设计上分三层，即数据计算层、任务处理层和数据访问层，每层都能完善自身的功能，最终将每一层整合起来，就可以完成一个优秀的性能分析系统。数据计算层是整个性能分析系统的核心，根据业务需求的不同计算不同的数据；任务处理层是整个系统的中枢，起到承上启下的作用，用户触发任务，处理层进行具体业务处理，读取并展示相应的数据；数据访问层，采用数据库连接池技术，减少数据库连接时间，而且该层实现了系统中设计的数据访问，提高了代码重用性，消除了上层对数据源的直接依赖，隐藏了实现细节，降低了模块间的耦合性。

1. 性能分析系统数据库设计

微软小娜性能分析系统采用SQL Server数据库。SQL Server是微软推出的一个关系型数据库，是一个可扩展的、高性能的、为分布式服务器计算所设计的数据库管理系统，提供优秀的并发控制方案，具有良好的运行效率，有利于提高分析系统的运行效率，提供了基于事务的企业级信息管理系统方案，同时Visual Studio完美支持SQL Server的开发工作，开发人员可以方便的在Visual Studio 上查看、调试和管理数据库。

完善优秀的系统表设计是微软小娜性能分析系统的基础，优秀的系统表能够向系统提供可靠的存储、维护、检索数据功能。微软小娜性能分析系统数据库主要涉及到的信息包括，各个功能模块的用户量，各个功能模块的使用次数，各个功能模块的用户次日存留率，各个功能模块的成功率和各个功能模块的延迟时间。在实现微软小娜性能分析系统数据库的过程中要遵循以下原则：保持规范化，满足范式，减少重复数据；设计合理，结构优化；满足性能分析系统的业务扩展性。

微软小娜性能分析系统的表结构说明，如表所示：

|  |  |  |
| --- | --- | --- |
| 表名 | 描述 | 功能 |
| AvailCoA | Android版微软小娜成功率 | 保存Android版微软小娜每个功能模块的使用次数和成功率 |
| AvailCoAByNetwork | Android版微软小娜按照网络分类后的成功率 | 保存Android版微软小娜按照网络分类后每个功能模块的使用次数和成功率 |
| LatencyCoA | Android版微软小娜延迟时间 | 保存Android版微软小娜每个功能模块的使用次数和延迟时间 |
| LatencyCoAByNetwork | Android版微软小娜按照网络分类后延迟时间 | 保存Android版微软小娜按照网络分类后每个功能模块的使用次数和延迟时间 |
| UsageCoA | Android版微软小娜用户量 | 保存Android版微软小娜每天的活跃用户量和每天的新用户量 |
| AvailCoI | iOS版微软小娜成功率 | 保存iOS版微软小娜每个功能模块的使用次数和成功率 |
| AvailCoIByNetwork | iOS版微软小娜按照网络分类后的成功率 | 保存iOS版微软小娜按照网络分类后每个功能模块的使用次数和成功率 |
| LatencyCoI | iOS版微软小娜延迟时间 | 保存iOS版微软小娜每个功能模块的使用次数和延迟时间 |
| LatencyCoIByNetwork | iOS版微软小娜按照网络分类后延迟时间 | 保存iOS版微软小娜按照网络分类后每个功能模块的使用次数和延迟时间 |
| UsageCoI | iOS版微软小娜用户量 | 保存iOS版微软小娜每天的活跃用户量和每天的新用户量 |

1. 本章小结

本章介绍微软小娜分析系统的总体设计与分析，依照可靠性原则、安全性原则、易用性原则、先进性原则和可扩展性原则进行总体设计，梳理了微软小娜性能分析系统的业务逻辑，整理出性能分析系统的功能设计，进而采用Scope + .NET + MVC的架构构建分析系统，设计出了性能分析系统的整体框架，分为数据计算层、任务处理层和数据访问层，最后根据业务需求以及系统功能设计出数据库的相关表结构。