竞品项目优化案例分析

内容大纲

1 串播单排查页面优化

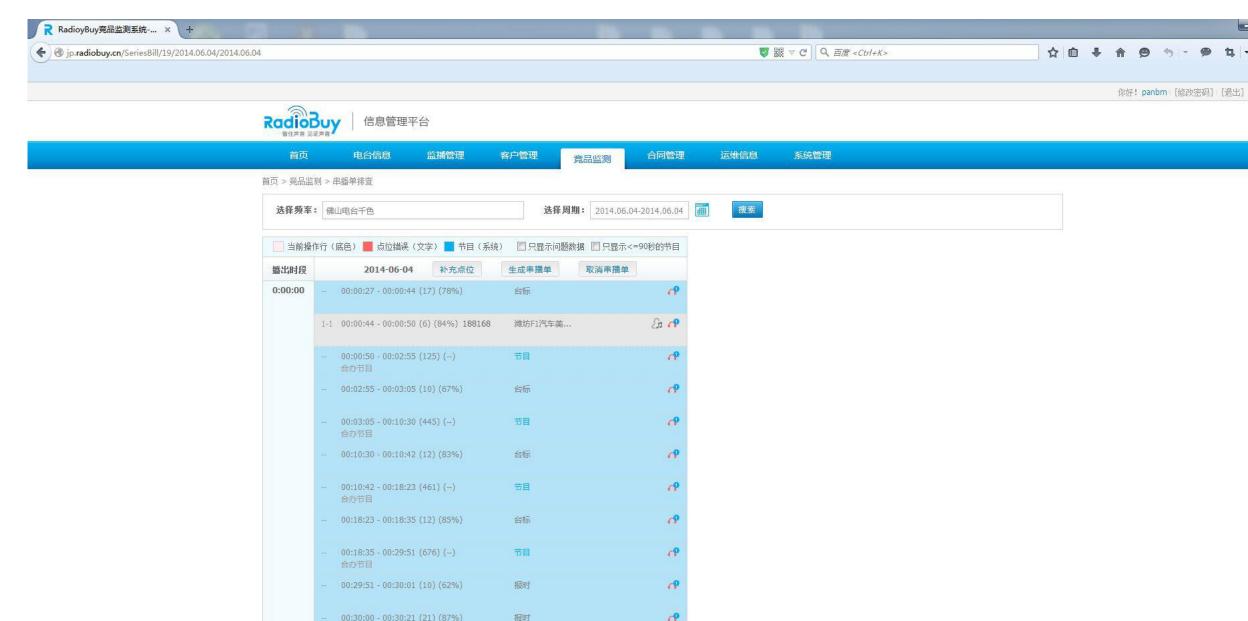
2 广告版本管理 -- 根据播放时间搜索的优化

网站优化流程

浏览器->IIS服务器->ASP.NET->数据库

二八原则

1 串播单排查页面



1.1 查找慢SQL

工具: SQL Profiler 监控Duration大约3000毫秒以上的SQL

1.2 监控这次请求暂用的时间

工具: HttpModule

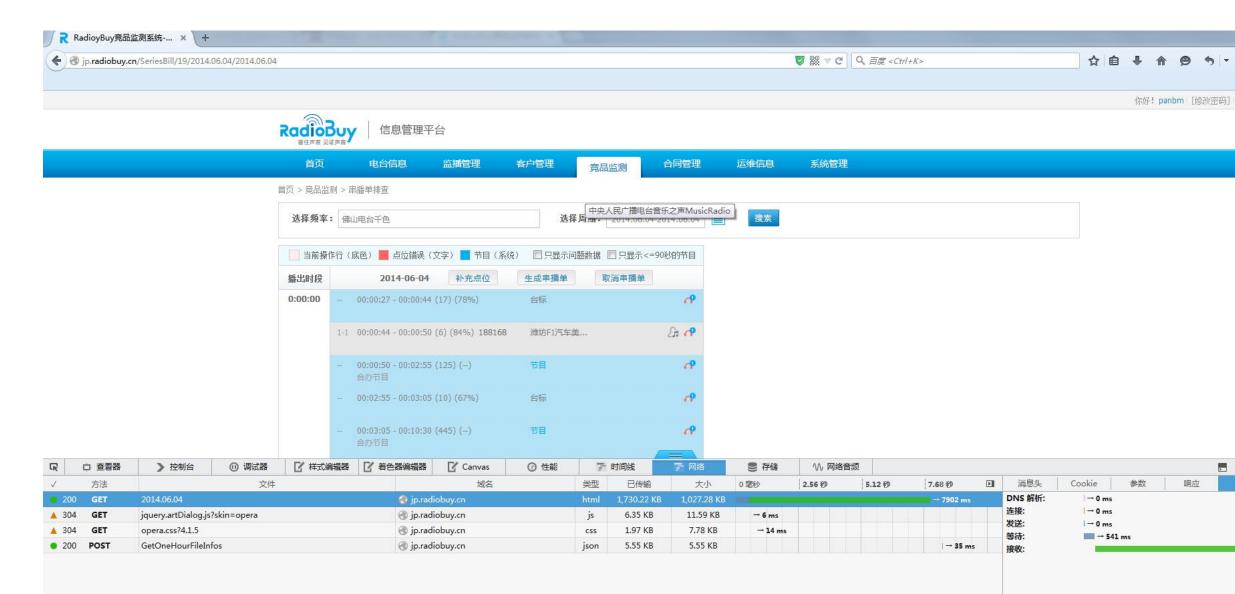
```
public class PerformanceCounterModule: IHttpModule
    private static readonly object lockKey=new object();
    public void Init(HttpApplication context)
        context.BeginRequest += context_BeginRequest;
        context.EndRequest += context EndRequest;
   void context BeginRequest(object sender, EventArgs e)
        HttpApplication httpApplication = (HttpApplication) sender;
        httpApplication.Context.Items["beginTime"] = DateTime.Now;
    void context_EndRequest(object sender, EventArgs e)
        HttpApplication httpApplication = (HttpApplication)sender;
        var beginTime=Convert.ToDateTime(httpApplication.Context.Items["beginTime"]);
        TimeSpan timeSpan = DateTime.Now - beginTime;
        string message = string.Format("url : {0} time : {1}", httpApplication.Request.Url, timeSpan.TotalMilliseconds);
    public void Dispose()
```

1.3 查看IIS的日志

找出耗时最长的请求

建议: 以后加上日志收集分析工具

1.4 用FireFox跟踪浏览器请求



1.5 排除网络延时

```
PS C:\Users\dell> ping jp.radiobuy.cn -1 1024
|正在 Ping jp.radiobuy.cn [119.254.85.97] 具有 1024 字节的数据
    119.254.85.97 的回复: 字节=1024 时间=4ms TTL=119
  自 119.254.85.97 的回复: 字节=1024 时间=5ms TTL=119
来自 119.254.85.97 的回复:字节=1024 时间=4ms TTL=119
来自 119.254.85.97 的回复:字节=1024 时间=4ms TTL=119
|119.254.85.97 的 Ping 统计信息:
   数据包:已发送 = 4,已接收 = 4,丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
   最短 = 4ms, 最长 = 5ms, 平均 = 4ms
```

1.6 测试下载速度

- 公司服务器进行了下载速度限制
- 用360网络速度测试 最大下载速度: 250KB/s

1.7 解决方案1

IIS的动态内容压缩是基于MIME类型的,在IIS的默认设置(applicationHost.config)中,定义了一些mimeType,只有响应内容的Content Type与mimeType匹配时,IIS才会对之进行压缩。

Content-Type是text/html; charset=utf-8,这个mimeType在IIS的默认设置中没有定义,所以没有被压缩。只需加个这个Content Type就能解决问题。

IIS6 没有applicationHost.config

建议:升级IIS6

1.8.1 解决方案2-自己开发压缩模块

尝试了在endrequest编写代码,始终失败。

在staskoverflow找到了解决方案,并对改方案改进。

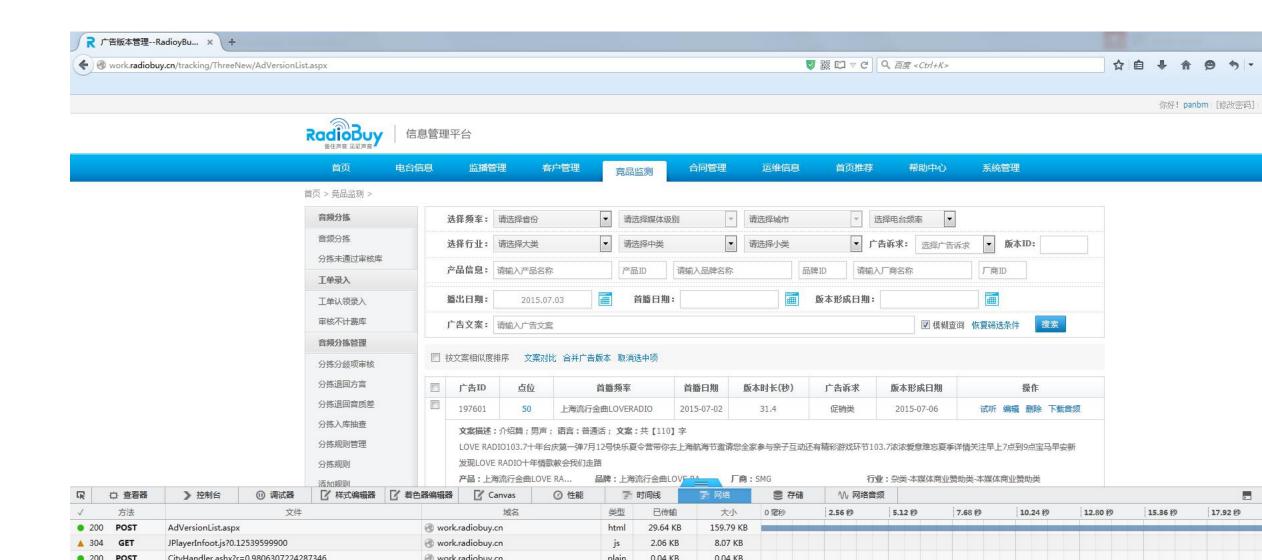
1.8.2 解决方案2一代码

```
public class DynamicCompressionModule : IHttpModule
   //可以进行压缩的类型
   private static HashSet<string> contentTypeSet;
   static DynamicCompressionModule()
       contentTypeSet = new HashSet<string>()
           "text/html",
           "text/plain"
   public void Init(HttpApplication context)
       context.ReleaseRequestState += context_ReleaseRequestState;
   void context_ReleaseRequestState(object sender, EventArgs e)
       HttpApplication context = ((HttpApplication)sender);
       //step1# 不压缩静态资源,静态资源又IIS进行压缩
       if (WebUtil.IsStaticResource(context.Request))
           return;
       //step2# 只压缩文本和html格式的数据。
       string acceptedTypes = context.Request.Headers["Accept-Encoding"];
       if (string.IsNullOrWhiteSpace(acceptedTypes) || !IsCanCompressed(context.Response.ContentType))
           return;
       //step3# 进行gizp压缩
       context.Response.AppendHeader("Content-Encoding", "gzip");
       context.Response.Filter = new GZipStream(context.Response.Filter, CompressionMode.Compress);
```

1.9 压缩后的效果



2 广告版本根据播放时间搜索优化--优化前

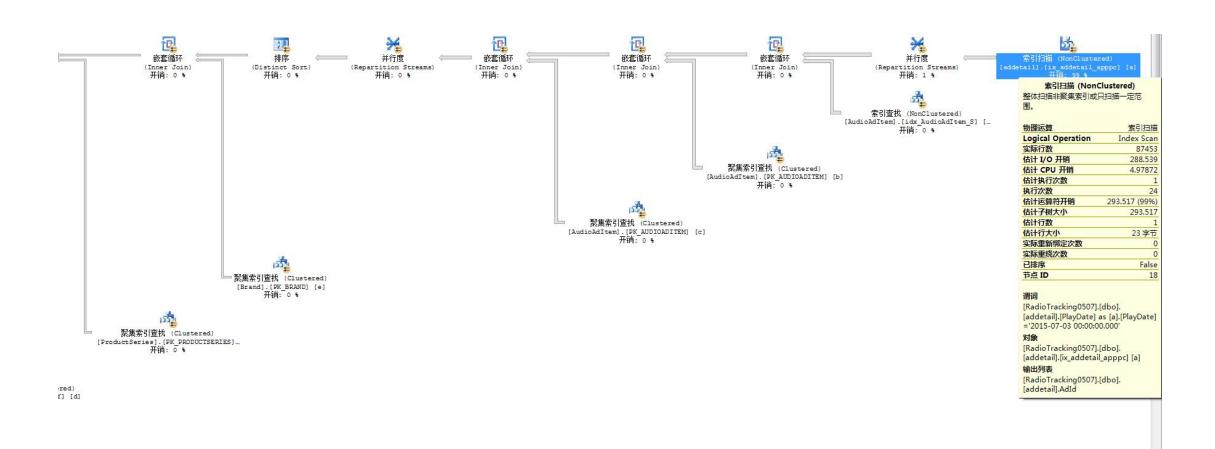


2.1 监控慢SQL

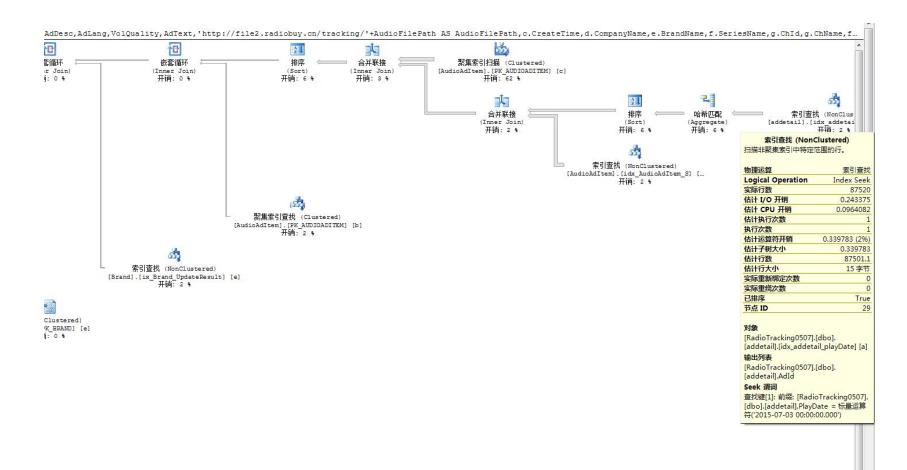
监控到存储过程的问题。

[p_GetAudioAdItemList] 这个存储过程

2.1 查看执行计划



2.2 添加索引



2.3 添加索引后的效果

