基于数据可视化的DCS系统日志分析方法研究及应用

Research and Application of DCS System Log Analysis Method Based on Data Visualization

★北京广利核系统工程有限公司 康炤旭

摘要：DCS系统是核电站的神经中枢，DCS系统的安全稳定运行对核电站的安全具有重要意义。DCS系统本身具有丰富的自诊断功能，当系统出现异常也会记录大量的在线和离线日志。对于离线日志目前的分析手段比较单一，只能通过人工方式进行日志的提取与分析。本文描述了一种提高离线日志的提取与分析效率的方法，利用Windows自带的API函数，文件传输命令等自动获取DCS系统离线日志并统一收集到指定节点。利用Python语言对收集到的离线日志进行分析，结合DCS系统特点，从离线日志中提取出影响DCS系统稳定运行的信息。这些信息采用JSON异步读取的方式，通过ECharts图表工具将分析结果以html格式进行在线展示，辅助操作员对DCS系统的健康状态进行辅助分析及判断，为DCS的智能化运维提供了理论与技术参考。

关键词：Python；ECharts；DCS日志；IIS；JSON

Abstract: DCS system is the nerve center of normal operation of nuclear power plant. The safe and stable operation of DCS system is of great significance to the safety of nuclear power plants. The DCS system itself has rich self-diagnosis functions, and a large number of online and offline logs will be recorded when the system is abnormal. At present, the analysis method of offline log is relatively simple, and the log can only be extracted and analyzed manually. This paper describes a method to improve the efficiency of offline log extraction and analysis of offline logs, using the API function and file transfer command of windows to automatically obtain the offline log of DCS system and collect it to the designated nodes. Python language is used to analyze the collected offline logs. According to the characteristics of the DCS system, the information that affects the stable operation of the DCS system is extracted from the offline logs. The information is read asynchronously by JSON, and the analysis results are displayed online in HTML format through Ecarts chart tool, which helps operators to analyze and judge the health status of DCS system, and provides theoretical and technical reference for intelligent operation and maintenance of DCS.

Key words: Python; ECharts; Log of DCS; IIS; JSON

1 引言

核电DCS系统指的是核电数字化控制系统（Digital Control System），是核电厂的神经系统和大脑，DCS系统本身的健康对整个核电厂的安全至关重要。DCS系统由硬件系统、软件系统及相关网络构成，每一部分出现异常均会记录相应的运行日志，日志分为在线日志和离线日志两个部分。在线日志可以直接在DCS系统的人机界面进行展示，提示操作员对当前实时出现的系统故障进行及时处理。离线日志则是系统运行期间记录在设备本身的运行日志，用于对故障的根本原因进行分析。但是当前离线日志只能手动拷贝，受限于核电现场的特殊性，对设备的操作需要遵循特殊规定，等待时间窗口，时效性大打折扣，往往会出现日志被覆盖的情况。同时，日志信息的分析也只能依靠人工，效率较低，不能满足DCS现场运行的需要。本文基于Windows的API程序接口，结合Python、ECharts等工具实现CPR1000项目非安全级DCS系统日志的自动获取、归集与分析，并将结果以html格式进行展示。

2 DCS系统日志系统组成

2.1 DCS系统架构

CPR1000项目非安全级DCS系统由三层网络、两级设备组成，分别为监视控制级设备，包括操作站、服务器、工程师站等设备，连接MENT和SNET网络，实现DCS系统设备监视与控制功能；过程控制级设备，包括就地控制柜、网关柜、远程I/O柜等设备，连接CNET网络，实现就地工艺系统设备的控制。如图1所示。

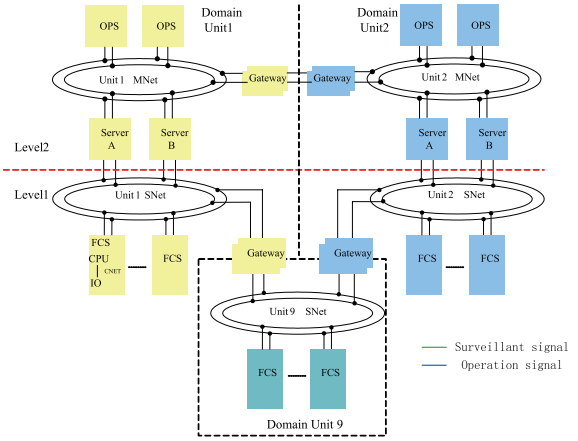

图1 CPR1000非安全级DCS网络结构

2.2 DCS系统日志系统

日志系统负责记录和存储核电站DCS系统发生的所有事件和信息，并根据时间类型对日志进行分类，按照时间进行存放。日志分为操作系统日志及平台日志两大类，其中平台日志又分为在线日志和离线日志两大类。在线日志系统记录的信息主要包括开关量和模拟量的值和状态的变化情况、设备故障和恢复信息，以及操作员的操作记录。日志信息可以输出到显示器、打印机，并以文件形式永久保存于硬盘上，这些文件能够导出，并能通过专用工具进行解析。离线日志系统记录的信息主要包括DCS系统各子软件系统的运行状态信息：服务器实时数据管理服务、历史库服务、报警及事件管理服务、内部计算服务、I/O数据采集服务等子进程的运行状态信息（超时、故障、数据对齐等）、操作站在线显示任务运行状态信息、控制器运行状态信息（同步、IEC运算任务、数据同步等）、通讯站运行状态信息（建立连接、数据包收发等）。操作系统日志信息主要针对Windows操作系统，利用系统本身的日志记录功能记录操作系统运行状态信息：如各种服务的启动、运行、关闭等信息，包括系统日志、应用程序日志、安全日志等[1]。

3 DCS系统日志获取

3.1 DCS操作系统日志获取

3.1.1 Windows API接口

应用程序接口（Application Programming Interface，API）是一些预先定义的函数，或指软件系统不同组成部分衔接的约定。[2]

API用来提供应用程序与开发人员基于某软件或硬件得以访问的一组例程，通过API函数，应用程序可以很方便地访问操作系统的各种服务，而无需访问源码，或理解内部工作机制的细节。通过调用Windows系统的API函数即可实现对操作系统日志的获取。

3.1.2 DOS命令

DOS是Disk Operating System的简称，在Windows图形界面操作系统出现之前，DOS系统几乎是人与计算机进行交流的唯一选择。DOS命令分三大类：内部命令、外部命令和批处理命令。常见的ping、copy、dir等都是DOS命令。其中wevtitul命令即为操作系统日志操作命令，通过该命令可以将操作系统日志获取到指定目录，如

wevtitul epl System c:\a.evtx.

但是该命令有两个应用限制：（1）只有在Windows7及以上操作系统上才支持该命令；（2）导出的日志信息为Windows系统特殊的evtx格式，不便于后续的分析。

3.1.3 方案实现

通过Python语言编程，调用Windows操作系统的API函数，实现操作系统日志的获取，具体实现方式如下（示例）：

```
import win32con
import win32evtlog
import win32evtlogutil  #引入API接口库
def SysLogs (server,logtype, logPath):#定义读取函数
log.write("\n%s Log of %s Events\n" % (server, logtype))
log.write("Created: %s\n\n" % time.ctime())
log.write("\n" + line_break + "\n")
log.write("Event Date/Time: %s\n" % the_time)
log.write("Event ID / Type: %s / %s\n" % (evt_id, evt_type))
log.write("Record #%s\n" % record)
  log.write("Source: %s\n\n" % source)
log.write(msg)
log.write("\n\n") #定义日志记录格式，便于统计分析
if __name__ == "__main__": #调用函数
server = None  # None = local machine
logTypes = ["System", "Application", "Security"]
SysLogs (server, logTypes, r'C:\Users\DCS\Desktop')
```

运行该程序后，对应节点的三个重要操作系统日志“系统”、“应用程序”、“安全”即以log格式文件（localhost_System_log.log）获取至本地C:\Users\DCS\Desktop目录，文件内容如图2所示。

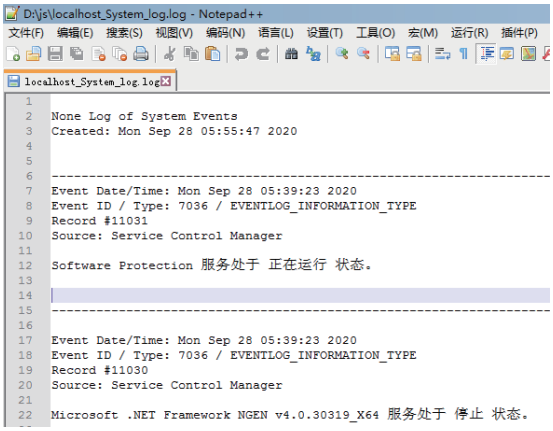

图2 日志文件示例

3.2 DCS系统平台日志获取

DCS系统的平台日志均是以文件的格式存储在对应系统设备的硬盘上，直接拷贝该文件即可实现对平台日志的获取。

3.2.1 Windows系统设备

对于Windows系统设备，所有的平台日志均存储在对应设备指定的目录下，结合操作系统日志的获取方式，统一获取方式，通过Python语言调用操作系统的DOS命令，实现日志文件的拷贝，具体实现方案如下：

```
import os  #引入操作系统库组件
os.system("xcopy /k /y D:\\HOLLiAS_MACS\\Log D:\\log")
```

执行上述命令，即可将对应节点D:\HOLLiAS_MACS\Log目录下的文件拷贝至本地D:\log下。

3.2.2 Linux系统设备

对于Linux系统设备（如网关、通讯站），平台日志存储在对应设备指定的目录下，此类设备已经默认开启了FTP服务，可以通过FTP方式将需要的日志文件下载到本地目录。同时也为了统一，依然采用Python语言调用DOS命令的方式实现，具体实现方案如下：

（1）新建一个批处理文件log.bat,文件内容如下(*后为注释文字，实际文件中不需要)，并保存在FTP启动目录下：

```
open 128.0.0.65          *远端设备IP地址
administrator            *远端设备的登录用户名
abc                      *远端设备的登录密码
cd /MacsRTS/log          *切换到日志文件所在目录
get THREAD_TCP_LOG.log   *下载日志文件
bye                      *退出FTP
```

（2）使用Python语言调用FTP命令，实现日志文件的获取，具体如下：

```
import os  #引入操作系统库组件
os.system("FTP-s:log.bat")
```

执行上述命令，即可将远端设备/MacsRTS/log目录下的日志文件下载到本地使用。

3.2.3 嵌入式系统设备

对于嵌入式系统设备（控制站主控），其运行日志采用专有格式存储在设备上，需要通过专门的读取工具进行读取。日志文件无法用常规软件打开，需要专门解析工具进行格式转换后才能打开，目前不考虑使用自动化手段进行读取。

4 DCS系统日志分析

4.1 运用现有工具

对于操作系统日志，有Splunk、SolarWinds&EventManager等专业工具进行分析，但是对于DCS系统平台日志无现有工具进行统计分析，使用受到限制。

4.2 高级语言编程

结合前面日志的获取使用Python语言实现，而Python语言在数据分析方面具有天然的优势。

Python是一种面向对象、解释型计算机程序设计语言，被称为胶水语言，能够和其他语言如C++、Java等结合。它具有语法简洁、数据结构高效等特点，开源且能够跨Windows、Linux、MacOS平台使用[3]。因其开源的特性，Python具有十分丰富的第三方库，可以很方便进行调用实现特定的程序功能。如Numpy库（Numerical Python）是一个Python科学计算的基础包。它不但能够完成科学计算任务，而且能够作为高效多维数据容器，存储和处理大型矩阵[4]；Matplotlib是一个数学图表库，可以使用其完成数据图表的制作；Scrapy库是一个爬虫库，可以通过它在网络上爬取需要的任何信息。

4.3 方案实现

综合前文两种方案的对比，结合DCS平台日志本身的特点，同时能更好地与日志的获取衔接，选择使用Python语言进行日志数据的简单统计与分析。对于操作系统日志，每一个操作系统事件都有唯一的一个ID号代表，采用事件ID号作为统计关键字避免事件信息过长不便于统计的弊端。具体实现方案如图3所示。

（1）操作系统日志信息分为三大类：信息类信息，如用户登录，系统服务运行等成功操作的信息记录；警告类信息，

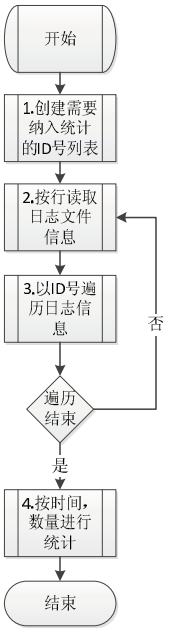

图3 方案实现流程图

不是直接的、主要的，但是会导致将来问题发生的信息记录，如当磁盘空间不足时将会产生一条警告信息；错误类信息，是指需要通知用户知道的重要问题，通常指功能和数据的丢失。并不是所有的信息对DCS系统的健康状态都有意义，因此根据日志对DCS系统重要性，对日志信息进行筛选并创建列表待用。DCS相关ID号（示例）如表1所示。

表1 DCS相关ID号列表（示例）

序号	ID号	描述	备注
1	57	网络适配器硬件出错	提示网络异常，影响DCS网络通讯
2	59	出现了意外的网络错误	提示网络异常，影响DCS网络通讯
3	4202	系统检测到网卡 Realtek...Family PCI Fast Ethernet NIC - 数据包计划程序微型端口与网络断开	提示网络异常，影响DCS网络通讯

（2）使用Python语言中的文件操作函数对日志文件按行进行读取。

（3）使用while，for循环语句实现以（1）中的ID号在日志文件中进行遍历。

（4）对遍历结果进行统计。

实现代码如下（片段）：

```
with open(file,encoding='utf-8') as fileobject:
    lines = fileobject.readlines()
i = 0
while i < n:
    for line in lines:
        if seri["fault"][i] in line.strip() and (sysid[i] in line.strip()):
            seri["num"][i] += 1
    i += 1
```

5 DCS系统离线日志可视化展示

5.1 可视化展示方案选择

在得到统计数据的基础上，选择合适的可视化展示方案进行数据展示。数据可视化方案繁多，有入门级的Excel，有内容更丰富的D3、Visual.ly，有支持用户互动操作、支持高度个性化定制的ECharts、Hicharts，也有专业级的SPSS、SAS。

Excel自带饼图、折线图、条形图等多种图表，能够很方便地创建图表，但是制作图表对原始数据的排列格式要求较高，需要花费精力进行整理，同时不支持多个显示终端的同时查看。Excel图表示例如图4所示。

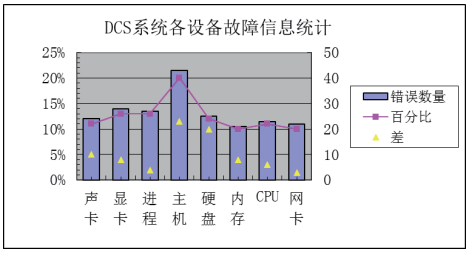

图4 Excel图表示例

SAS工具是专业级的图表绘制工具，可以根据需要绘制各种信息丰富的图表，同时支持强大的编程功能，支持各种复杂的统计分析（回归、生存、方差、多变量分析等）。但是其需要有较强的编程基础才能驾驭，学习成本较高。

ECharts是由百度公司推出的一款可视化图表控件，底层依赖Canvas类库ZRender，可提供直观、交互、个性化定制的数据图表，不但支持柱状、折线、饼图等普通图表，还支持地图、和弦图、数据视图、大规模散点图等特殊图表，主要采用Canvas进行渲染绘图。[5] 它提供丰富的图表实例，根据实际需要替换其中的数据系列即可完成满足需求的数据图表。同时ECharts能够像普通JavaScript库一样很方便地引用到html代码中实现数据图表的绘制，并可以通过Ajax异步请求的方式更新图表数据，实现图表的动态显示。

Ajax是通过支持异步请求技术的XmlHttpRequest对

象向服务器发异步请求，从服务器获得请求数据，然后用JavaScript操作DOM模型进行交互和动态显示，实现页面数据更新的一种异步通信技术[6]。考虑到核电现场使用环境的特殊性、使用成本、部署的便利性，采用ECharts作为可视化工具。

5.2 可视化方案介绍

整体数据可视化方案为通过Python语言实现DCS操作系统日志、平台日志的读取并进行数据统计。统计结果由Python语言写入到JSON文件保存。前端采用ECharts工具读取JSON文件进行可视化图表绘制与分析，借用Windows系统本身自带的互联网信息服务（Internet Information Services，IIS）发布在局域网上，从而支持多个终端同时查看。方案架构如图5所示。

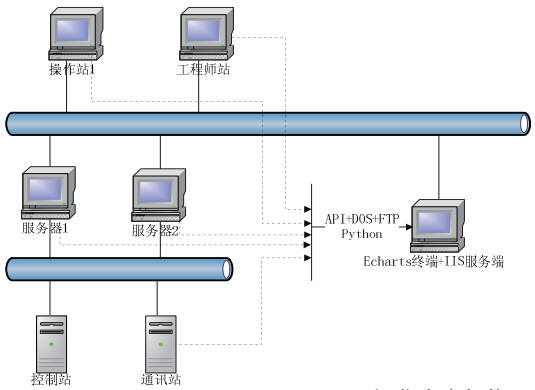

图5 可视化方案架构图

方案详细介绍：

（1）选择一台ECharts终端，可根据核电现场实际选择，备份服务器、配置服务器等。

（2）ECharts终端同时作为网页发布服务器，配置IIS服务，如图6~10所示。

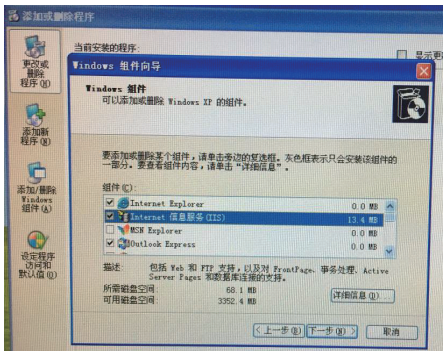

图6 安装IIS服务

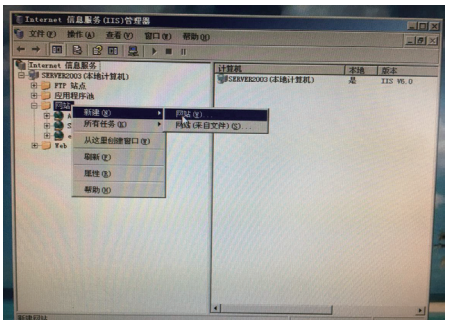

图7 创建网络站点

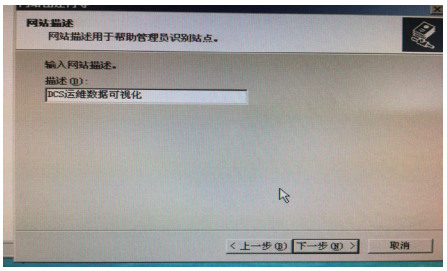

图8 定义网站描述

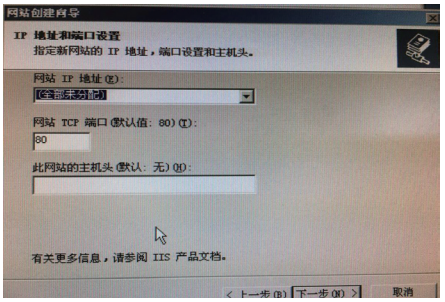

图9 指定网站IP及端口号

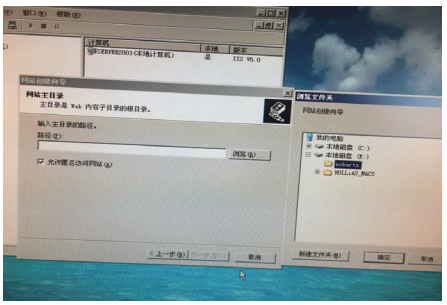

图10 指定网页路径

（3）将Python语言编写的日志获取、日志分析两部分代码整合并打包为可执行文件放置在ECharts终端，并配置为周期运行，其运行结果为一个包含统计数据的JSON格式文件。

（4）编辑ECharts代码，引入JSON格式数据，实现数

据的可视化展示，代码实现如图11、12所示（片段）。

（5）在显示终端打开IE浏览器，输入

http://130.0.0.89/omc.html（地址根据实际情况输入），即可显示可视化图表如图13所示。

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>DCS日志可视化展示</title>
    <!-- 引入 echarts.js -->
    <script src="D:\js/echarts.js"></script>
    <!-- 引入 jquery-3.js -->
    <script src="jquery-3.5.1.min.js"></script>
</head>
    <style type="text/css">
        #bar,#bar1,#bar2,#bar3{width:900px;height:600px;borde
    </style>
<body>
    <!-- 为ECharts准备一个具备大小（宽高）的Dom -->
    <div id="bar"></div>
    <div id="bar1"></div>
    <div id="bar2"></div>
    <div id="bar3"></div>
    <script type="text/javascript">
        // 基于准备好的dom，初始化echarts实例
        var myChart_bar = echarts.init(document.getElementBy

        // 指定图表的配置项和数据
        var option = {
            title: {
                text: '故障日志数量'
            },
            tooltip: {
            trigger: 'axis',
            axisPointer: {
                type: 'cross'
```

图11 代码示例1

```
// 基于准备好的dom，初始化echarts实例
var myChart = echarts.init(document.getElementById("bar"));
myChart.setOption({
title:{text: '故障日志数量'}, //图标题
legend:{data:['数量']}, //图例
xAxis:{data:[]}, //x轴属性
yAxis:{},//y轴属性
//图属性
series:[{
name:'数量',
type:'bar',//图类型，柱状图
color:['#C0FF3E'],//设置图像颜色
data:[]//图表的数值
}]
});
//使用jQuery中的$.get()获取data.json文件，使用done函数；
//done(function(data))中data表示调用的函数返回的数据
$.get('data.json').done(function(data){
    console.log(data)
myChart.setOption({
xAxis:{data:data.category},
series:[{
name:'数量',
data:data.data}]
});
});
</script>
```

图12 代码示例2

图13 可视化图表

如图13，左上图：故障数量统计图，以柱状图的形式直观展示各种故障的统计数量，支持鼠标左右拖动。右上图：设备故障趋势图，与折线图的形式直观展示各种设备故障数量的变化趋势，提示维护人员对变化趋势异常的设备提前进行干预。左下图：故障种类分析图，内环展示设备种类，外环展示对应设备故障数量及百分比。右下图：故障原因气泡图，横纵坐标分别代表操作系统日志异常数量、DCS平台日志异常数量，并以气泡面积大小表异常数量总数情况。

6 结论

为将DCS运行过程中产生的离线日志进行可视化展示，辅助操作员对DCS系统的健康状态进行分析。本文采用Python语言调用Windows系统的API接口及批处理命令对DCS系统的离线日志进行自动化读取并进行数据统计。统计的结果以JSON格式进行存储供前端的ECharts进行异步调用，实现可视化图表的展示。本方案为核电DCS系统离线日志的获取与展示提出了一种可行的方案，为DCS系统的智能化运维拓展了一个方向。未来还需要在数据分析方面深入研究，借助大数据分析方法，将收集到的日志信息进行多维度、更精细的分析，由预防性维修向预测性维修转变。同时考虑将现场带有日志记录信息功能的设备日志均自动化地提取出来参与分析，如控制器、交换机等，丰富展示结果。AP

作者简介：

康炤旭（1984-），男，山东德州人，学士，现就职于北京广利核系统工程有限公司，研究方向是核电数字化仪控系统设计、维护。

参考文献：

[1] 王照文. 基于事件ID的Windows日志集中审计[J]. 中国金融电脑, 2011, (4) :80 – 82.

[2] 沙晓晨. 应用程序接口版权保护及限制研究[D]. 南京：南京师范大学, 2017.

[3] 陈嘉发. Python数据可视化的应用研究[J]. 福建电脑, 2019, (5) : 114 - 116.

[4] 黄琪. 基于Python的数据可视化方法和系统实现[J]. 信息与电脑, 2019, (14) : 137 – 140.

[5] 徐欣威. 基于ECharts的科技统计数据可视化设计与实现[J]. 天津科技, 2019, (3) : 66 – 70.

[6] 赵海国. Ajax技术支持下的ECharts动态数据实时刷新技术的实现[J]. 电子技术（上海）, 2018, (3) : 25 -27.