Website：ycyk.brit.com.cn

基于容器云的地面测控资源池日志管理系统的研究

马潇潇[1]，孙甲琦[2]，朱宏涛[1]，张宇东[1]
（1 北京遥测技术研究所 北京 100076；
2 中国航天电子技术研究院 北京 100094）

摘要：新型航天地面测控装备基于商用服务器集群，采用虚拟资源池架构和容器云技术，按需进行测控资源调度和测控业务能力生成。和众多服务设备一样，日志是系统故障诊断的主要途径，由于海量日志可能分散在资源池系统中的不同位置，传统的流式日志采集和分析模式难以大批量管理信息，经常导致系统异常发现不及时、排查效率低、定位难度大等问题，日志的检索和统计缺乏行之有效的手段，弊端明显。针对这种地面测控资源池系统的运维管理需求，本文设计了一种基于容器云的日志管理系统，将平台日志和应用日志聚合管理，根据用户需求和运维目的分类检索并定制可视化日志显示，为使用者精准分析系统异常、快速进行系统维护提供有效支撑。

关键词：容器云；日志；采集；检索；资源池

中图分类号：TP311.5；V556　　**文献标志码：**A　　**文章编号：**2095-1000(2023)06-0057-07

DOI：10.12347/j.ycyk.20230206001

引用格式：马潇潇，孙甲琦，朱宏涛，等. 基于容器云的地面测控资源池日志管理系统的研究[J]. 遥测遥控，2023，44（6）：57–63.

Research of ground TT&C resource pool log management system based on container cloud

MA Xiaoxiao[1]，SUN Jiaqi[2]，ZHU Hongtao[1]，ZHANG Yudong[1]
（1. Beijing Research Institute of Telemetry, Beijing 100076, China;
2. China Academy of Aerospace Electronics Technology, Beijing 100094, China）

Abstract: The new aerospace ground TT&C equipment is based on commercial server clusters and adopts virtual resource pool architecture and container cloud technology for on-demand TT&C resource scheduling and TT&C business capability generation. Like many service equipment, logs are the main way of system fault diagnosis. Since a large amount of logs may be scattered in different locations in the resource pool system, the traditional streaming log collection and analysis mode is difficult to manage information in large quantities, which often leads to problems such as untimely discovery of system abnormalities, low troubleshooting efficiency and difficulty in locating, etc. The lack of effective means for log retrieval and statistics has obvious drawbacks.To meet the operation and maintenance management needs of such ground-based TT&C resource pool system, this paper designs a container cloud-based log management system, which aggregates and manages platform logs and application logs, retrieves and customizes visual log display according to user requirements and maintenance purposes, and provides effective support for users to accurately analyze system abnormalities and quickly perform system maintenance.

Keywords: Container cloud; Logs; Collection; Search; Resoure pool

Citation: MA Xiaoxiao, SUN Jiaqi, ZHU Hongtao, et al. Research of ground TT&C resource pool log management system based on container cloud [J]. Journal of Telemetry, Tracking and Command, 2023, 44(6): 57–63.

引　言

随着我国航天工程从太空走向深空，航天测控网建设进入了全面发展新阶段。新型航天地面测控装备采用天线组阵技术接收更遥远的深空信号，采用资源池架构和容器云技术按需进行测控资

基金项目：2021年航天科技集团青年技术人才支持工程
收稿日期：2023-02-06　修回日期：2023-03-14

源调度和测控业务能力生成[1]。其中，资源池系统作为地面站所有测控数传设备信号处理与站内管控的统一平台，基于商用服务器集群，采用资源池技术和容器云平台相结合的系统架构，主要承担航天器测控数传信号采集合成处理、站内测控资源调度及运行管理任务。其逻辑架构如图1所示。

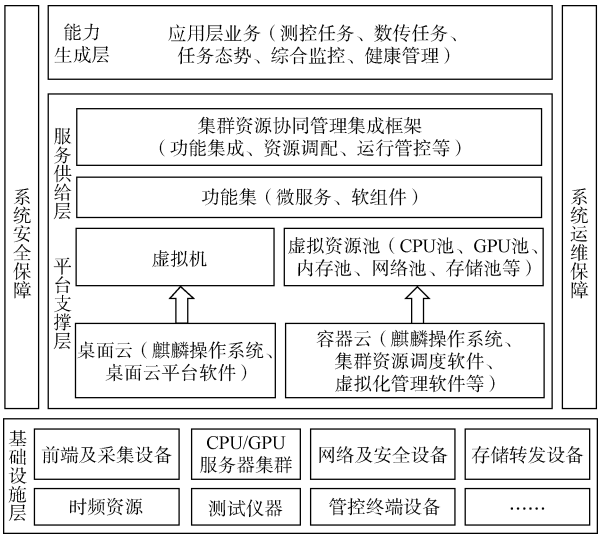

图 1 资源池系统逻辑架构

Fig. 1 Resource pool system logical architecture

资源池系统分为基础设施层、平台支撑层、服务供给层、能力生成层和系统保障层，其中系统保障层包括系统安全保障和系统运维保障两类功能，日志管理功能是系统运维保障功能的重要组成部分。

为了满足资源池系统基础设施层CPU/GPU服务器集群弹性管理以及各类测控资源虚拟化管理的需求，同时便于为能力生成层提供服务化功能和灵活调配架构，在平台支撑层建立容器云平台[2]。容器云以容器为资源分割和调度的基本单位，封装整个软件运行时环境，为开发者和系统管理员提供用于构建、发布和运行分布式应用的平台[3]。资源池系统依托Kubernetes(容器调度管理工具)平台对服务供给层、能力生成层和系统保障层等提供服务，该工具平台能够管理上千个节点的集群，在Kubernetes内可以通过命令去获取容器日志信息，但容器提供的本地功能通常不足以支撑完整的日志记录解决方案[4]。当系统出现故障需要排查时，若仅靠命令行的方式去操作几乎是不可行的。因此，日志功能应该具有独立于节点、Pod(Kubernetes中能够调度和管理的最小单元)或者容器的单独存储和生命周期。

本文设计的基于容器云的日志管理系统是集群级的日志平台，负责采集来自资源池系统各层的工作日志，涵盖集群资源的状态上报信息、支撑平台的系统日志信息以及各容器业务功能的工作日志信息等，将其存储到本地数据库中，供不同使用者基于角色进行检索查询，便于用户进行可视化系统运维、基于历史大数据进行趋势预测等。

1 总体架构设计

基于容器云的日志管理系统在应用场景上应充分考虑集群节点和云平台的部署应用、资源池管理和业务能力动态生成过程的运行监测、基于设备及业务历史数据的健康管理和运维保障等，在系统功能上应重点具备以下三方面：

① 采集汇聚集群节点和云平台的运行管理日志；

② 采集汇聚各容器业务软件的运行管理日志；

③ 基于角色的可视化查询报表功能。

结合地面测控资源池系统特点和上述功能需求，本文对日志管理系统进行了分层架构设计，将日志管理系统分为采集代理层、处理服务层和界面展示层等，如图2所示。

资源平台层是日志管理系统的日志源以及部署、运行所依赖的环境。采集代理层以JSON文件方式与系统资源层对接，从资源层采集到系统硬件、平台以及软件等产生的日志传递给处理服务层。采集代理层根据资源池系统节点容器应用部署情况部署配置不同数量的日志采集代理，采集所负责节点的日志信息。处理服务层主要提供存储和检索服务，负责接收各采集代理采集到的原始日志信息，并进行格式处理，将处理后的日志写入到本地数据库中存储，同时对界面展示层提供日志实时分析检索服务。界面展示层提供了可视化人机交互界面，用户能够实时搜索和查看系统状态，便于系统的运行、维护和管理。

本文将日志管理系统分成三层进行设计和实现，有利于实现各层之间组件模块的解耦，各层的组件模块只需要关注实现自身的功能和性能需求，也有利于日后的维护和改良。日志采集代理层关注实现低延迟高并发的日志采集请求，并将日志数据上报到处理服务层；处理服务层关注实现对海量日志数据的合理存储，根据用户需求进

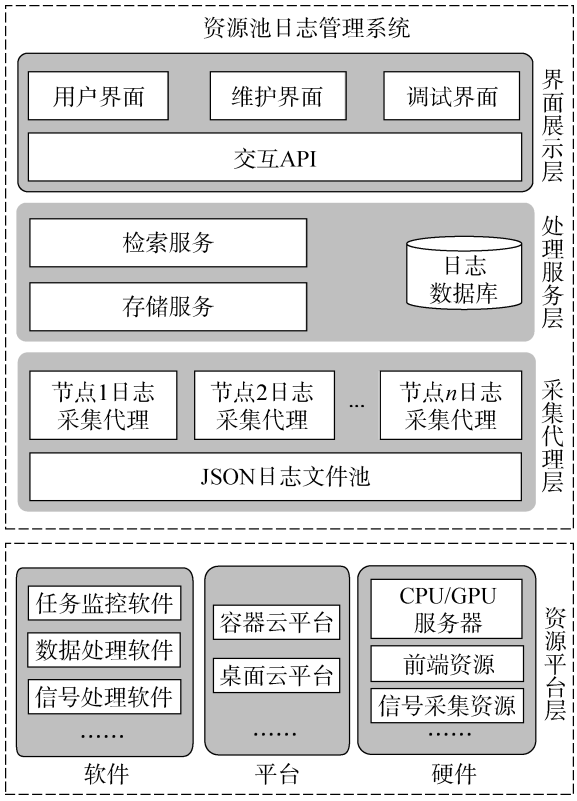

图 2　日志管理系统总体架构

Fig. 2　Overall architecture of the log management system

行高性能实时检索；界面展示层关注实现日志的数据化和信息可视化，便于用户操作管理。

2　功能设计与实现

资源池系统采用Kubernetes工具平台管理测控集群资源和微服务架构业务应用软件。基于Kubernetes的应用日志输出方式一般包括三种[5]：

① 遵循Docker官方建议把日志输出到标准输出(或者标准错误输出)控制台，使用Kubectl logs命令可以看到日志，比如Nginx的日志就是输出到控制台[6]。

② 输出日志文件到容器内指定文件系统的目录中，比如tTomcat的日志就是写入到容器中的文件系统的文件。

③ 应用软件直接将日志发送给日志收集系统。

此外，也可以基于应用系统全新开发一套面向平台日志和容器业务日志的日志管理系统，但这种日志输出方式无太大优势[7]。本文设计的日志管理系统结合资源池特点和Kubernetes平台管理要求[8,9]，采用开源日志组件进行部署配置和性能优化，设计容器业务日志数据规范和交互规范，优化可视化人机交互界面，适用于海量平台日志和容器日志的实时采集、存储管理和高效检索展示。

2.1　采集代理层设计与实现

采集代理层日志采集功能采用节点代理方式。采用这种方式的主要原因包括：① 资源池系统业务软件分别部署在不同节点的pod中的多个容器里，各节点部署日志采集代理可以并行采集所在节点的容器日志。② 耗费资源少，只需部署在节点。③ 独立部署代理，与应用系统隔离，不影响任务应用。

本文选用Fluentd作为日志采集代理核心组件，它是一个用于统一日志采集的开源数据收集器，使用JSON文件来统一日志数据格式。它具有可插拔架构，支持各种不同种类和格式的数据源和数据输出，同时也具备高可靠性和高扩展性。相比Flumed、Logstash等其他采集组件，它具有配置简单、占用空间小、插件机制更灵活的优点，更适用于资源池系统各容器自定义数据源的日志采集[10]。

为了保证资源池系统集群中的每个节点都运行同样Fluentd的pod副本，并行收集每个节点的日志，使用Daemon set控制器部署Fluentd组件并设定日志过滤标准。本文统一定义各类日志信息如表1所示。

本文通过配置Fluentd自带或可适配的插件来实现日志的读取、过滤、格式处理和转发等功能。采用Tail插件来起到读取器的作用，获取服务器本地的日志文件作为输入，同时安装Dstat插件采集集群服务器及平台日志信息。

采用Parse插件按照配置的日志文件路径和文件名称来解析不同信源的日志。在资源池系统中，Parse插件使用正则表达式进行日志信息匹配，插件配置内容示例如图3所示。为了适配处理服务层的存储，提高传输效率，日志输出统一设计为JSON格式。其中，日志类型logType分为调试日志、运维日志、操作日志和运行日志，在系统配置中分别用0、1、2、3表示。

2.2　处理服务层设计与实现

处理服务层主要功能是对多源日志数据进行存储和检索。本文采用Elasticsearch作为日志处理服务核心组件，它是一个开源高扩展的分布式全文检索引擎，可以近乎实时地存储和检索数据。

表 1 日志信息定义

Table 1 Log message definition

日志项	定义	说明
序号	id	标记日志序号的逻辑数值，该条日志的主键
来源	logSrc	日志信息来源，标记该条日志是来自服务器、平台或具体的容器应用软件
生成日期	logDate	生成该条日志的当前日期
生成时间	logTime	生成该条日志的当前时间，根据测控数传任务需求，精确到毫秒
日志等级	logLevel	日志分为三个等级：提示、警告、错误
日志类型	logType	日志分为类型：调试日志、运维日志、操作日志、运行日志，用于不同角色用户进行检索使用，其中管理员可以查看全部信息，操作员可以查看操作日志和运行日志等任务信息
当前用户	curUser	当前登录系统的用户
日志内容	logContent	具体日志文本信息

```
<source>
  @type tail
  path /root/fluentd.demo/logs/appname.log


  <parse>
   @type multiline
   format_firstline /\d{4}.\d{1,2}.\d{1,2}/
   format /^(?<id>\d{4})(?<logSrc>[^\s]*)(?<logDate>\d{4}.\d{1,2}.\d{1,2})(?<logTime>\d{4}\d{1,2}:\d{1,
2}:d{1,2}\.\d{3})(?<logLevel>[^\s]*)(?<logType>\d{1}(?<curUser>\d{1})(?<logContent>[\s\S]*)

</source>
```

图 3 Parse 插件配置内容示例

Fig. 3 Example of parse plugin configuration content

该组件扩展性好，可以扩展处理大数据时代的 PB 级体量的数据[11]。另外，资源池系统长期运行，日志数据量大但日志数据结构不复杂，查询条件多样化，在故障诊断时还将用到模糊匹配，所以采用 Elasticsearch 比较适宜[12]。

本文采用 Docker 技术[13]部署由多个节点组成的 Elasticsearch 集群，拉取 Elasticsearch 镜像，上传到集群每个节点，通过 docker load 指令解压，在 yaml 文件中注入环境变量、pod 模板、容器参数等，实现各节点的配置，包括节点名称、集群名称、端口号和资源限制等。在服务器中启动容器，用 kubectl apply 指令将 yaml 文件启用，即可完成集群部署。

Elasticsearch 集群是日志采集代理转发信息的接收端，它具有面向文档的存储方式，可以存储整个对象或文档，每个文档的内容都可以通过索引被搜索[14]。作为存储组件，Elasticsearch 本身是一个非关系型数据库，以 JSON 格式作为存储文档序列化格式。本文中将日志信息写入数据库的时间戳作为 RowKey，可确保主键的唯一性。采集到的日志信息作为 Value，通过键值对格式进行信息存储，一方面符合 JSON 的格式要求方便数据录入，另一方面也便于后续日志信息检索和归类。日志信息采集存储示例如图 4 所示。

2.3 界面展示层设计与实现

界面展示层主要功能是与处理服务层交互，提供可视化界面实时显示用户检索的日志信息。本文采用可与 Elasticsearch 无缝集成的 Kibana 作为日志界面展示核心组件，它是一个开源分析及可视化平台，可以搜索、查看交互存储在 Elasticsearch 索引中的数据，并通过各种图表进行高级数据分析及展示。基于浏览器的用户界面可以快速创建仪表板实时显示 Elasticsearch 查询动态[15]。

通过 Kibana 整合资源池系统各节点容器中业务软件日志信息，便于针对一个计划、一项任务、一次事件等基于时间轴进行索引处理，也可以针对关键词进行模糊匹配处理，并据此创建柱形图、折线图、散点图和地图等，便于用户查看。Kibana 通过在 yaml 文件中注入环境变量、类型设置的方

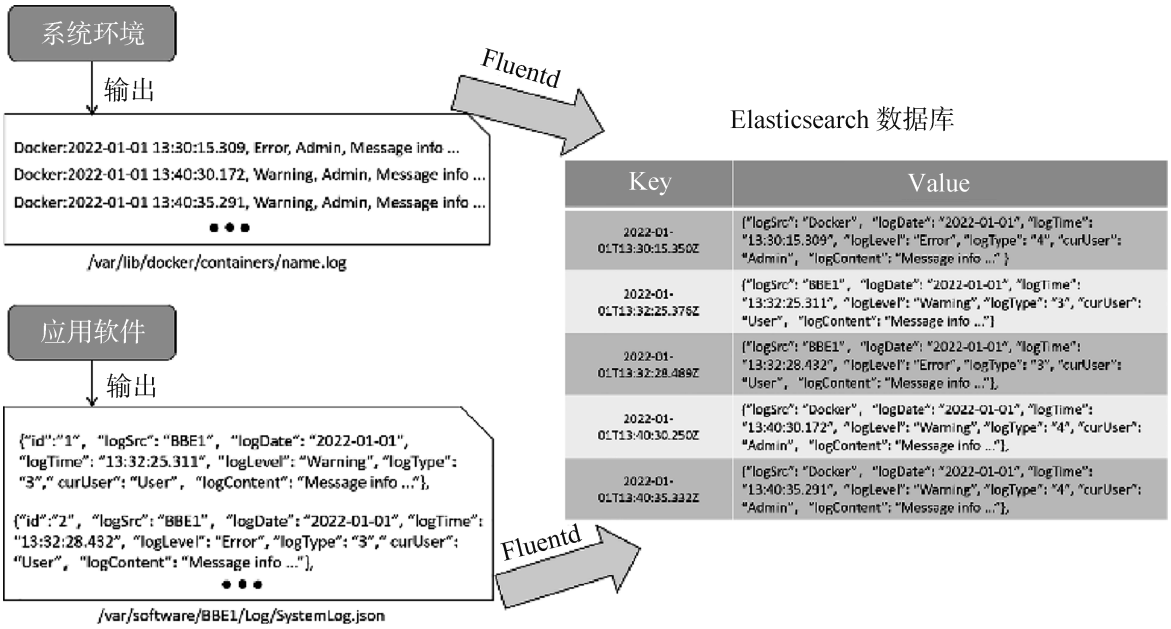

图 4　日志信息采集存储示例图

Fig. 4　Example diagram of log information collection and storage

式进行配置部署。

本文设计的日志主界面在用户登录，登录后根据当前登录用户身份进行一次检索，使用户登录后只能查询到对应类型的日志内容。

操作员登录后的界面如图 5 所示，Kibana 会自动生成时间轴。显示一定时间内产生的日志数据量，可以手动选择时间范围，对于用户来说，通过数据量时间轴的方式可以查看某任务在特定时间内是否成功启动，起到任务监控的作用。日志信息由用户类型筛选过后仅显示操作日志和运行日志，如系统状态和监控内容等。

图 5　操作用户日志可视化界面

Fig. 5　Visual interface for operating user logs

管理员登录后的界面如图 6 所示，管理员可查看包含操作员用户日志在内的所有信息，便于管理员掌握系统状态，进行故障排查、错误分析等工作。

图 6 管理员日志可视化界面

Fig. 6 Visual interface for administrator logs

3 主要性能测试

本文对日志管理系统进行了日志记录实时性测试和日志查询性能测试。日志记录实时性测试结果如图 7 所示，其中图右边为系统信息记录时刻，左边为本系统采集日志信息时刻，可见时间精度可达 ms 级，能够满足地面测控装备对日志信息记录的时间精度性能要求。

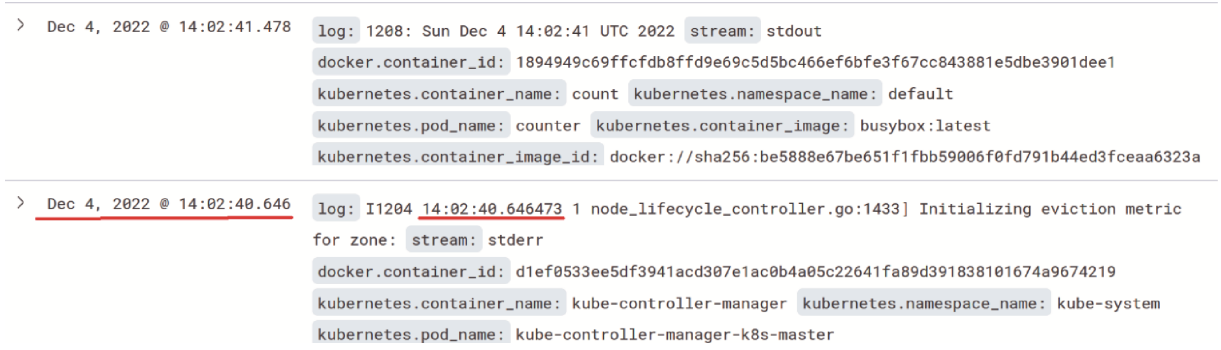

图 7 日志记录实时性测试

Fig. 7 Logging real-time testing

日志查询性能测试结果如图 8 所示，经多次测试，关键字检索时间在 500 ms 左右。测控装备日志查询功能主要用于事后运维使用，该查询时间满足使用需求。

4 结束语

本文设计的基于容器云的地面测控资源池日志管理系统，充分结合资源池技术和容器云技术，聚合了各类日志采集、存储和查询流程，适用于集群系统日志信息的集中和统一管理。该系统解决了当前应用系统面临的日志采集存储方式不统一、检索分析效率低、故障定位难等难题。针对日志记录实时性和查询性能的测试结果表明，该系统性能基本满足资源池系统的日志管理需求。未来的日志管理系统可以在智能化上做进一步研究，探索日志系统自动进行漏洞自查和问题分析处置。

图 8　日志查询性能测试结果

Fig. 8　Log query performance test results

参考文献

[1] 刘云杰, 王彬, 刘友永. 基于资源池的航天测控资源动态重组实现技术[J]. 无线电工程, 2018, 48(8): 699-703.
LIU Yunjie, WANG Bin, LIU Youyong. Research on implementing technologies of dynamic resources restructuring for aerospace TT & C based on resource pooling[J]. Radio Engineering, 2018, 48(8): 699-703.

[2] 王建宇, 苗恕宾, 艾娟. 容器云平台(CaaS)浅析[J]. 信息系统工程, 2017(12): 26.

[3] 刘昱良, 何璐. 基于 Docker 技术的容器云平台浅析[J]. 大众科技, 2021, 23(1): 15-17+20.

[4] 王伟军. 基于 Kubernetes 的容器云平台建设[J]. 电脑知识与技术, 2019, 15(36): 47-48.

[5] 雷惊鹏, 唐雅文, 颜世波, 等. 基于 Docker 和 Kubernetes 的 ELK 日志分析系统的研究与实现[J]. 太原学院学报: 自然科学版, 2021, 39(2): 61-67.

[6] 李灿彬, 甘宏. 基于 Docker 容器的云平台设计与实现[J]. 科技广场, 2017(6): 38-41.

[7] 王晓东. 基于虚拟化的分布式软件日志管理系统 VDSL 研究[J]. 电脑知识与技术, 2015, 11(10): 90-92.

[8] 陈金光. 基于阿里云的 Kubernetes 容器云平台设计与实现[D]. 杭州: 浙江大学, 2018.

[9] 翁涅元, 单杏花, 阎志远, 等. 基于 Kubernetes 的容器云平台设计与实践[J]. 铁路计算机应用, 2019, 28(12): 48-53.
WENG Shengyuan, SHAN Xinghua, YAN Zhiyuan, et al. Design and practice of container cloud platform based on Kubernetes[J]. Railway Computer Application, 2019, 28(12): 48-53.

[10] 张文燕, 张浩, 张进涛. 一种改进的 OpenStack 云平台日志收集系统[J]. 电子技术与软件工程, 2020(22): 153-155.

[11] 范朗. Elasticsearch 海量数据存储查询优化[J]. 工业控制计算机, 2020, 33(10): 85-87.

[12] 李中男, 郭猛, 宋梦也. 基于云计算资源池的运维管理平台[J]. 信息与电脑(理论版), 2020, 32(17): 13-15.

[13] 罗东锋, 李芳, 郝汪洋, 等. 基于 Docker 的大规模日志采集与分析系统[J]. 计算机系统应用, 2017, 26(10): 82-88.

[14] WEI Z J. Elasticsearch 中数据是如何存储的[EB/OL]. (2018-12-07)[2022-12-15]. http://elasticsearch.cn/article/6178.

[15] 韦疆臣. 基于 ELK 的大数据平台运维管理应用研究[D]. 西安: 西安电子科技大学, 2019.

[作者简介]

马潇潇　1999 年生，硕士研究生。

孙甲琦　1976 年生，博士，研究员。

朱宏涛　1977 年生，硕士，研究员。

张宇东　1993 年生，硕士，工程师。

(本文编辑：潘三英)