AIOps深度案例分享

Elastic Meetup 成都分享会

leiwang@orientsoft.cn

1

案例介绍

系统简介
客户介绍
系统规模
数据规模

2

实时采集引擎

实时采集
流计算
异常检测
根因推断

3

AI算法调度

算法与ES协同工作
ES内部完成数据清洗和数据聚合
为时序处理算法提供规整数据

4

总结

优势
劣势

案例介绍

案例介绍

客户

客户是股份制商业银行，资产在3000亿以上，同类金融机构中排名第七。

客户的痛点：有监控，没分析；有告警，量太大。

我们主要完成的是AIOps在银行生产业务系统上的一个工程实践。技术亮点主要体现在我们借助机器学习和Elasticsearch实现故障快速定位和对潜在问题的洞察。

案例介绍

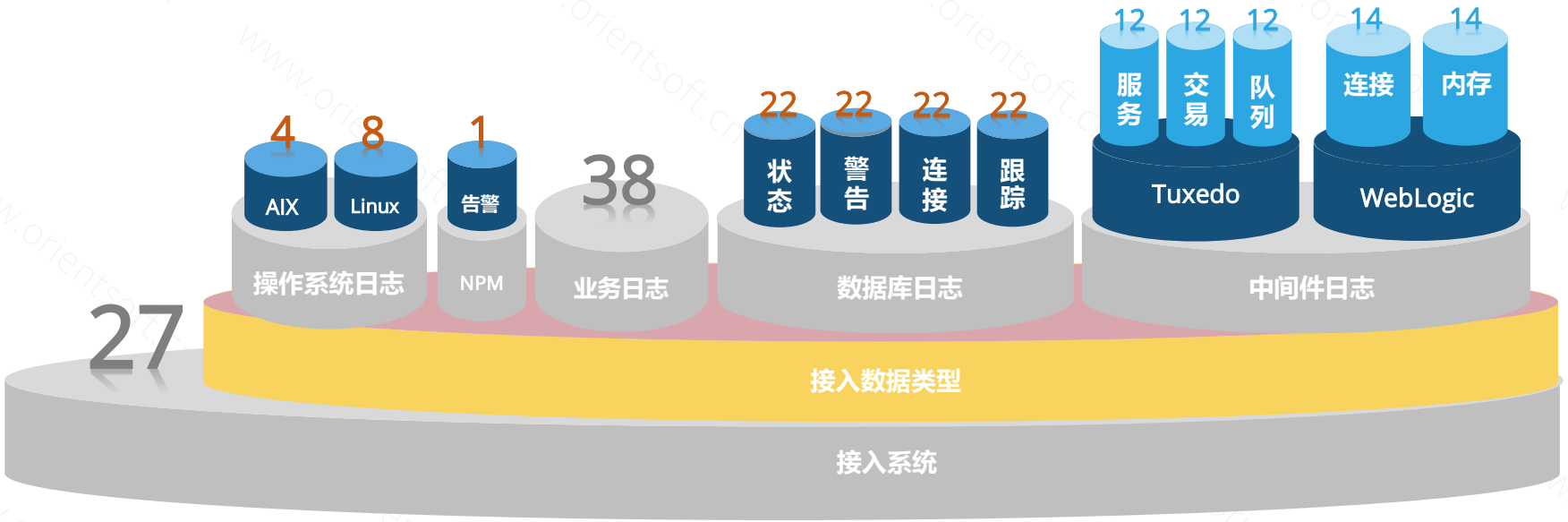
客户系统规模

完成27个一类系统的接入工作

(2019.04.25)

案例介绍

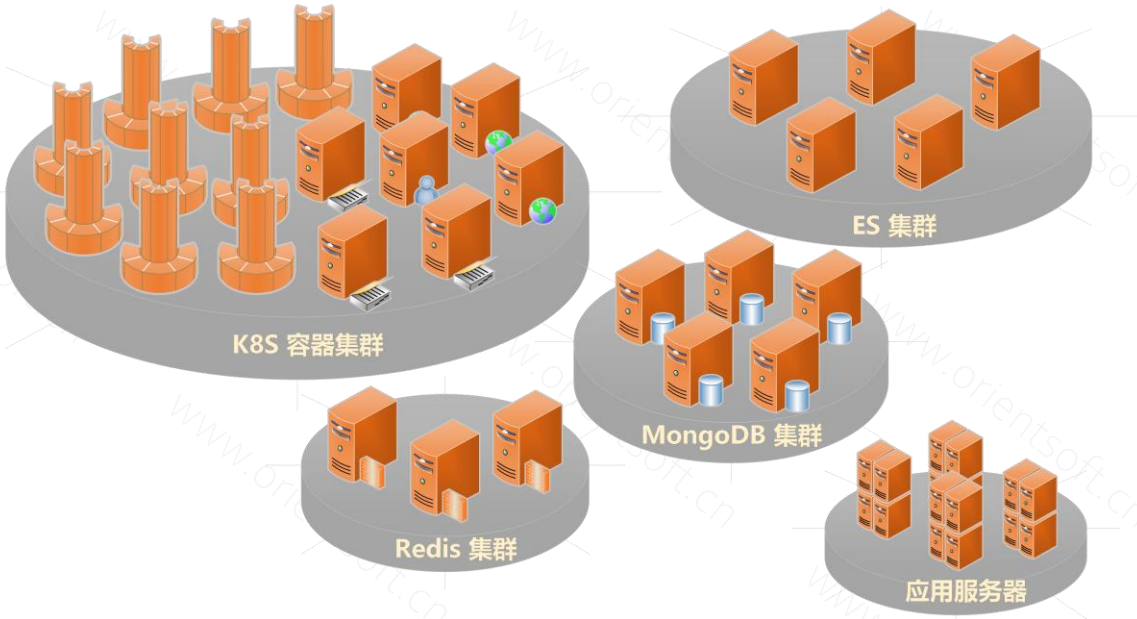
接入数据类型汇总



案例介绍

系统规模

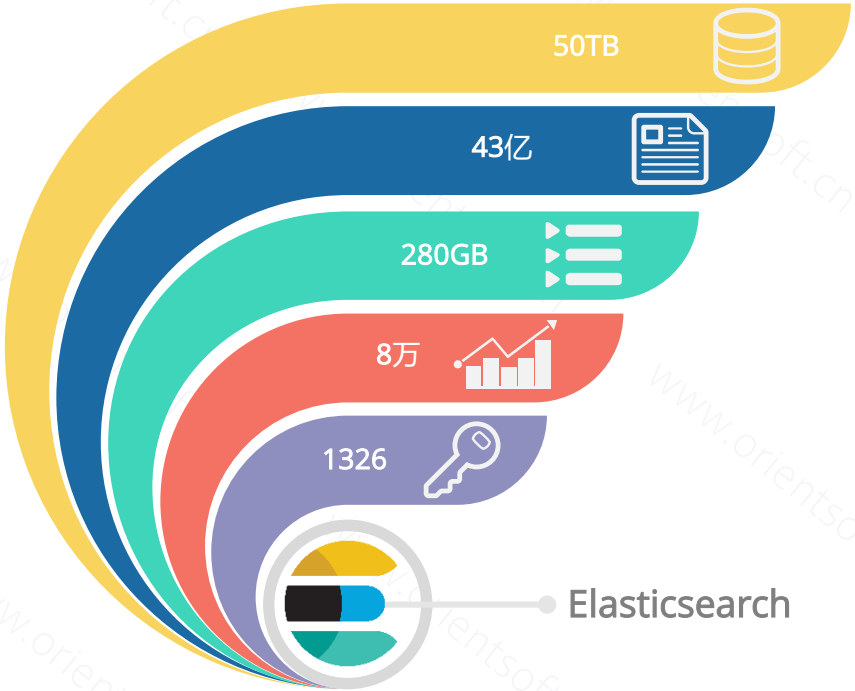
行大数据智能运维系统		
服务实例说明		
符号	计数	说明
	3	运维数据分析平台
	3	NSQ消息服务
	5	服务集群
	9	Conalog采集实例
	5	MongoDB服务
	1	APIBus节点
	5	ES服务
	4	应用服务
	3	Redis服务



ES版本：6.2.2；6个ES节点（4个data+master，2个coordinator），2个Kibana节点，15个Conalog采集节点，2个APIBus流计算节点，3个Redis节点，5个MongoDB节点，2个Prophet-server节点，2个AIDefender节点

案例介绍

数据规模（6个月）



50TB ES落盘数据量

43亿 ES落盘文档数

280GB 日均日志数据和状态数据总量

12万 ES文档落盘每秒峰值

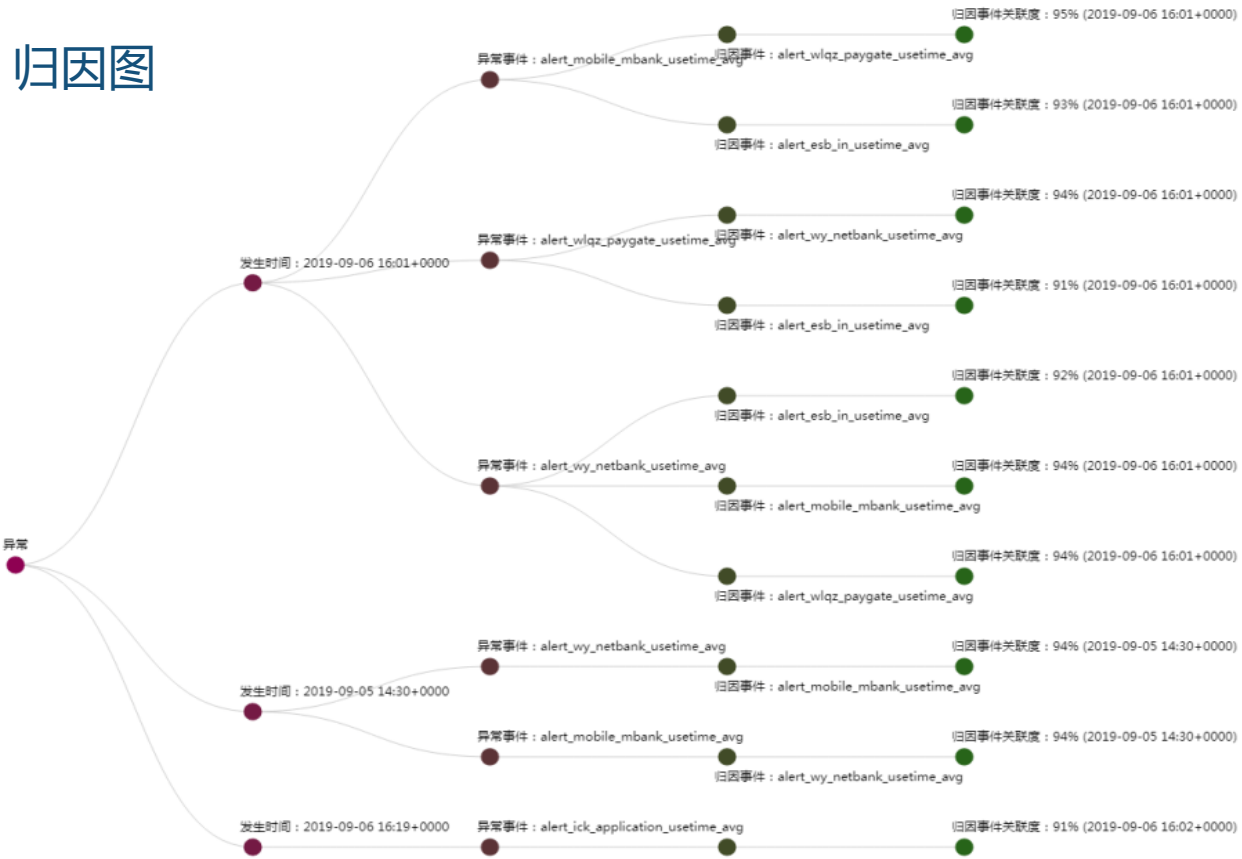
1326 ES索引总数

异常告警


全局监控
全局告警
核心业务-应用-失败率告警
核心业务-应用-平均耗时告警
核心业务-Tuxedo-平均队列长度告警
图形前端-应用-失败率告警
图形前端-应用-平均耗时告警
二代支付-应用-失败率告警
二代支付-应用-平均耗时告警
二代支付-WebLogic-平均队列长度告警
二代支付-WebLogic-丢失连接总数告警
二代支付-WebLogic-最大等待连接数告警
网上银行-应用-失败率告警
网上银行-应用-平均耗时告警
网上银行-WebLogic-平均队列长度告警
手机银行-应用-失败率告警
手机银行-应用-平均耗时告警
手机银行-WebLogic-平均队列长度告警
手机银行-WebLogic-丢失连接总数告警
手机银行-WebLogic-最大等待连接数告警
综合理财-应用-失败率告警
综合理财-应用-平均耗时告警
综合理财-WebLogic-平均队列长度告警
网联前置-应用-失败率告警
网联前置-应用-平均耗时告警
网联前置-WebLogic-平均队列长度告警
综合前置-应用-失败率告警
综合前置-应用-平均耗时告警
综合前置-Tuxedo-平均队列长度告警
一户通-应用-失败率告警
一户通-应用-平均耗时告警
一户通-WebLogic-平均队列长度告警

项目亮点

归因图


异常
发生时间：2019-09-06 16:01+0000
异常事件：alert_mobile_mbank_usetime_avg
归因事件：alert_wlqz_paygate_usetime_avg
归因事件关联度：95% (2019-09-06 16:01+0000)
归因事件：alert_esb_in_usetime_avg
归因事件关联度：93% (2019-09-06 16:01+0000)
异常事件：alert_wlqz_paygate_usetime_avg
归因事件：alert_wy_netbank_usetime_avg
归因事件关联度：94% (2019-09-06 16:01+0000)
归因事件：alert_esb_in_usetime_avg
归因事件关联度：91% (2019-09-06 16:01+0000)
异常事件：alert_wy_netbank_usetime_avg
归因事件：alert_esb_in_usetime_avg
归因事件关联度：92% (2019-09-06 16:01+0000)
归因事件：alert_mobile_mbank_usetime_avg
归因事件关联度：94% (2019-09-06 16:01+0000)
归因事件：alert_wlqz_paygate_usetime_avg
归因事件关联度：94% (2019-09-06 16:01+0000)
发生时间：2019-09-05 14:30+0000
异常事件：alert_wy_netbank_usetime_avg
归因事件：alert_mobile_mbank_usetime_avg
归因事件关联度：94% (2019-09-05 14:30+0000)
异常事件：alert_mobile_mbank_usetime_avg
归因事件：alert_wy_netbank_usetime_avg
归因事件关联度：94% (2019-09-05 14:30+0000)
发生时间：2019-09-06 16:19+0000
异常事件：alert_ick_application_usetime_avg
归因事件关联度：91% (2019-09-06 16:02+0000)

实时采集引擎

实时采集引擎

系统架构

Conalog实时数据采集引擎

APIBus实时流计算引擎

大数据存储服务集群

多维数据分析平台

智能工作站

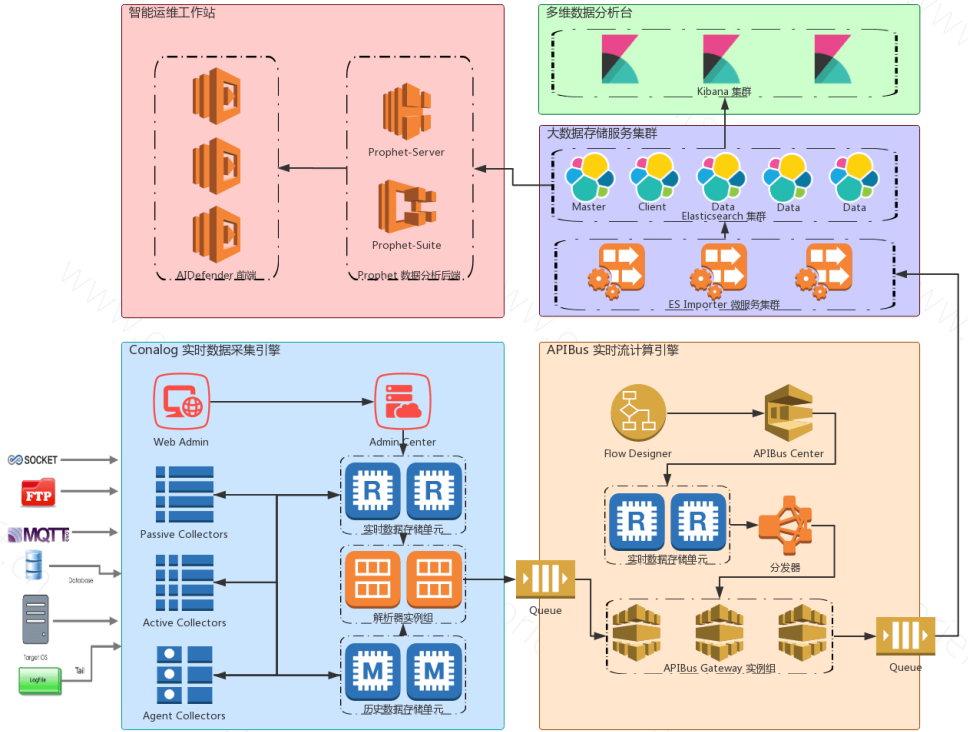
根因推断/故障预测
辅助决策/预测未来。

异常告警
数据价值转换为生产力。

监控展示
数据价值的呈现和展示。

数据处理
保障数据的有效性和正确性。

数据接入
为系统提供大数据运营的数据基础。



实时采集引擎

Conalog实时数据采集引擎

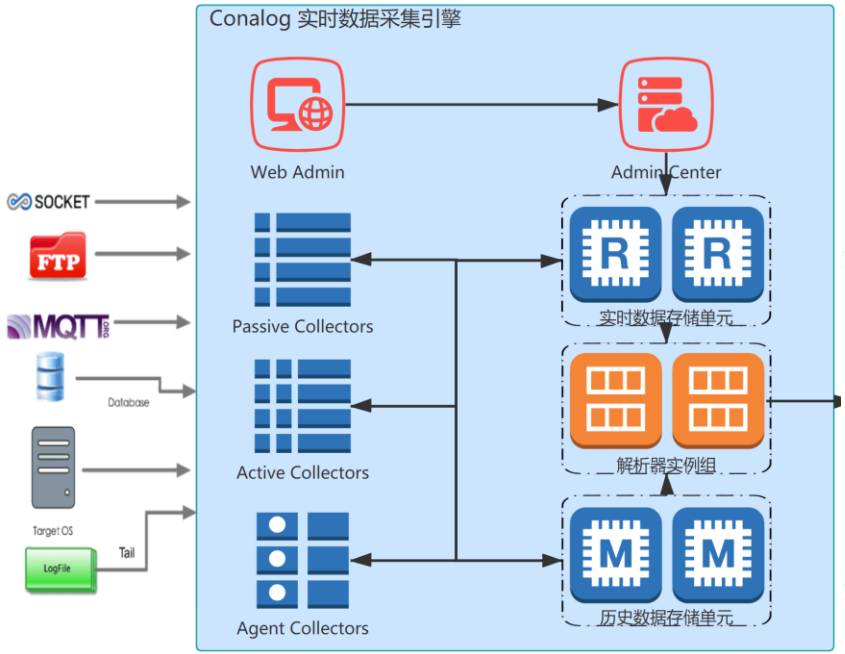
基于NodeJS开发的数据采集引擎；

非侵入式采集

接入灵活
- 被动采集器处理日志等需要被动接收的数据
- 主动采集器处理需要主动获取的状态类数据
- Agent采集器处理第三方的数据接入

实时性高
- SSH
- Redis PUB/SUB
- JS程序实时解析

容器化部署
- 支持K8s



实时采集引擎

APIBus实时流计算引擎

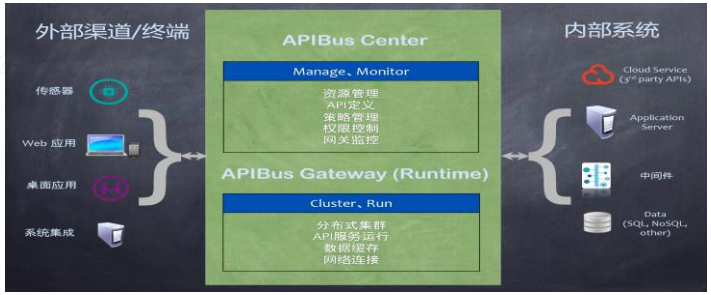
基于服务总线理念开发的实时流计算引擎；

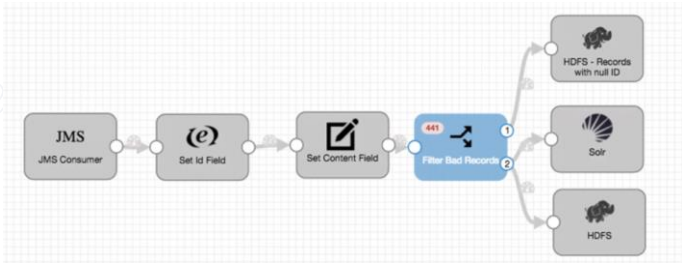
大规模可扩展性
- 2台 x86 PC Server 处理
  - 20w个并发连接
  - 每秒处理1.5w次API
  - 微秒级可预计延迟(在传输压力下)

对开发者友好
- 拖拽式开发、可见即可得的API,服务编排

与大数据预集成
- 直接访问Hadoop HDFS，HBase
- 将Event直接发给Spark Stream
- 作为生产者或者消费者使用Kafka
- 对接Elasticsearch和Solr等多种全文存储DB

容器化部署
- 支持K8s





实时采集引擎

ESImporter数据写入服务

基于微服务框架

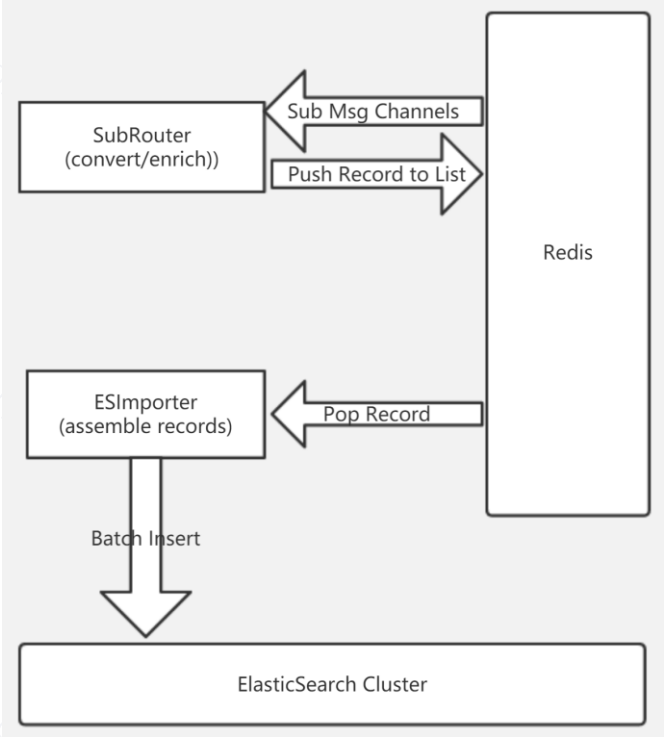
将数据流以秒级写入到Elasticsearch时序数据库

边缘计算的轻量级版本

同时支持其他第三方时序数据库，比如KDB和DolphinDB

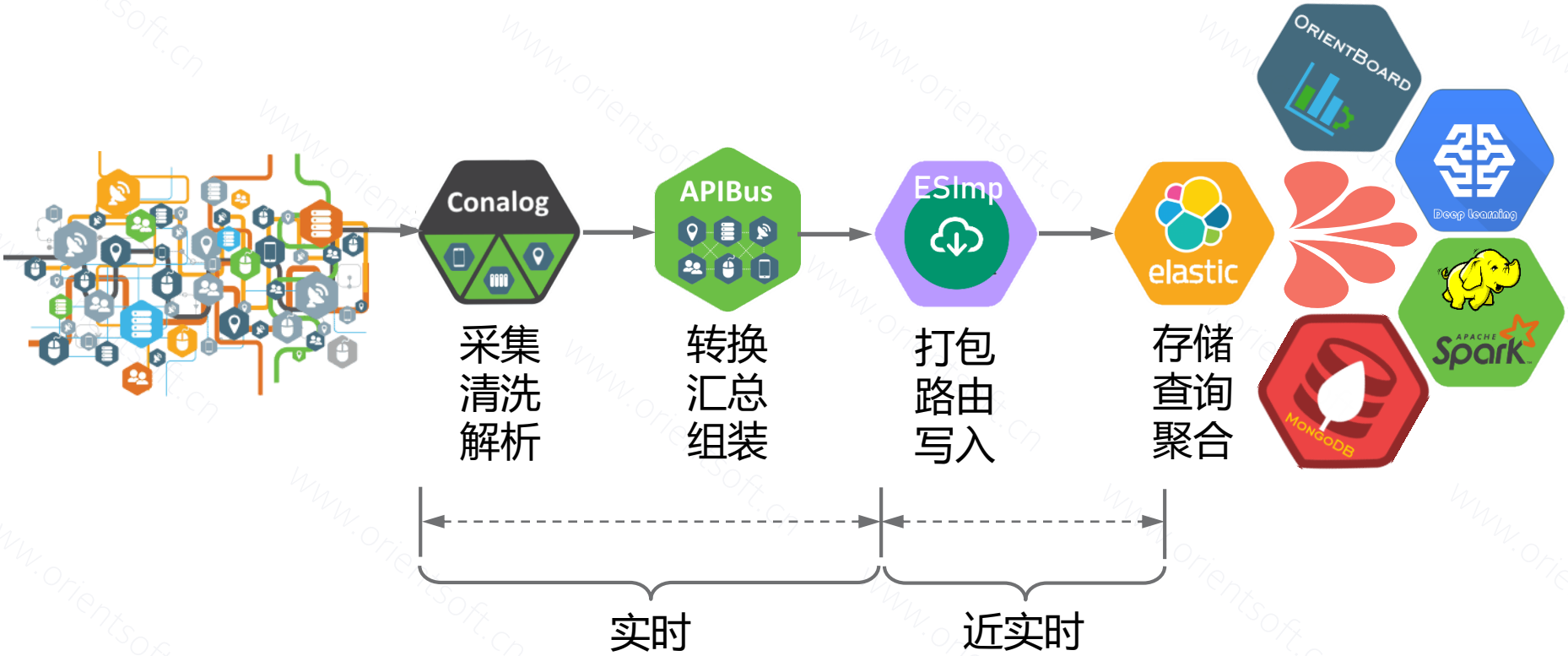
支持水平扩展

容器化部署



实时采集引擎

数据通路


Conalog
APIBus
ESImp
elastic
ORIENTBOARD
Deep learning
APACHE Spark
MONGODB
采集
清洗
解析
转换
汇总
组装
打包
路由
写入
存储
查询
聚合
实时
近实时

AI算法调度

AI算法调度

Prophet-Server算法调度

配置高度灵活

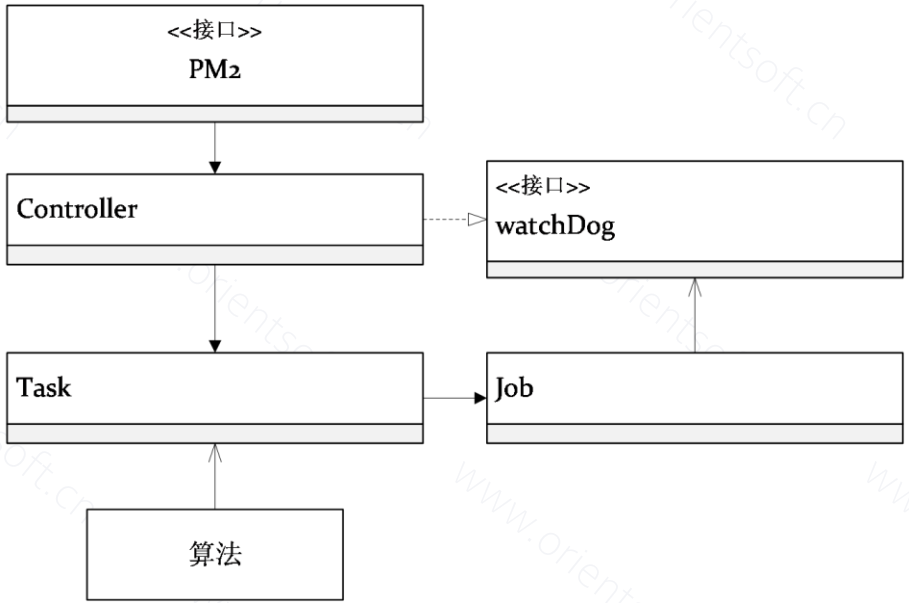
同算法的逻辑充分解耦

使用PM2调度

带算法死活检测

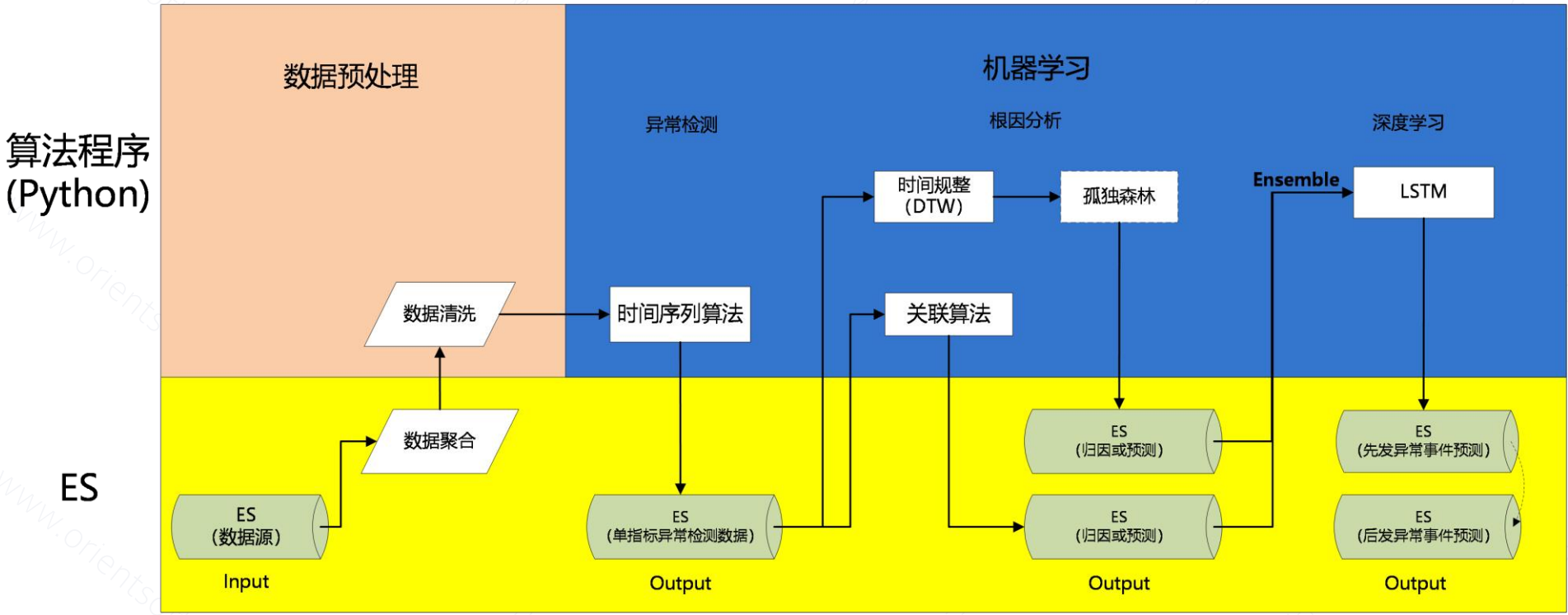
支持水平扩展

调度器本身可以容器化部署



AI算法调度

Prophet-Suite算法和ES的协同工作流



解决深度分页的问题

```
resp = helpers.scan(
    client = Elasticsearch(host=es_host,
      port=es_port,
      http_auth=(es_user, es_pwd)
    ),
    index = data_index,
    preserve_order = True,
    query = {
      "query": {
        "range": {
          "datetime": {
            "gte": start,
            "lt": end
            }
        }
      }
    }
)
```

```
resp = esClient.count(
  doc_type = 'data',
  index = data_index,
  body= {
    "query": {
      "range": {
        "datetime":{"gte":start, "lt" :end}
      }
    }
  }
)
```

```
resp = esClient.search(
  index = data_index,
  body = {
    "size": oneBatchSize,
    "search_after": docCursor,
    "query": {
      "range": {
        "datetime":{"gte":start, "lt" :end}
      }
    },
    "sort": [{"datetime": {"order": "asc"}}]
  }
)
```

总结

总结

ES 聚合能力很棒

- 8星期的数据
- 按每分钟聚合
- 80640条结果数据
- 35~110毫秒

KB 可视化能力超强

- 可视化图表生成
- Dashboard
- 第三方支持 Vega
- Canvas

数据接入有点水土不服

- 采集Agent不让装
- 即使让安装，每台机器要试验
- 对Agent的监控
- firebeat有些行为不能理解

人民中路二段68号中铁瑞城大厦1705室

Contact Us:

028-84118076

www.orientsoft.cn

leiwang@orientsoft.cn

https://github.com/Orientsoft

一起探索数据的更多可能

助力您的生产力，质的改变

了解产品服务

OrientSoft

谢谢大家！