



2023 | APACHE • SkyWalking
SUMMIT | CHINA · SHANGHAI

2023 · 上海

SkyWalking Summit





陈修能

江苏纵目信息科技有限公司 创始人

“基于 SkyWalking 实现
全域一体化观测”

目录

CONTENTS

01. SkyWalking 可以干哪些事？
02. SkyWalking + ClickHouse
03. 打通 SkyWalking 和 Zabbix
04. Argus 全域监控深度解析

01

SkyWalking 可以干哪些事 ?

1. 接入 Prometheus Exporter 取数

OpenTelemetry Metrics Format

The OpenTelemetry receiver supports ingesting agent metrics by meter-system. The OAP can load the configuration at bootstrap. If the new configuration is not well-formed, the OAP may fail to start up. The files are located at `classpath/otel-rules`.

Supported handlers:

- `otlp`: OpenTelemetry gRPC service handler.

Notice: Set `SW_OTEL_RECEIVER=default` through system environment or change `receiver-otel/selector=${SW_OTEL_RECEIVER:default}` to activate the OpenTelemetry receiver.

The rule file should be in YAML format, defined by the scheme described in [MAL](#). Note: `receiver-otel` only supports the `group`, `defaultMetricLevel`, and `metricsRules` nodes of the scheme due to its push mode.

To activate the `otlp` handler and relevant rules of `istio`:

```
receiver-otel:  
  selector: ${SW_OTEL_RECEIVER:default}  
  default:  
    enabledHandlers: ${SW_OTEL_RECEIVER_ENABLED_HANDLERS:"otlp-metrics"}  
    enabledOtelMetricsRules: ${SW_OTEL_RECEIVER_ENABLED_OTEL_METRICS_RULES:"istio-controlplane"}
```

The receiver adds label with key `node_identifier_host_name` to the collected data samples, and its value is from `net.host.name` (or `host.name` for some OTLP versions) resource attributes defined in OpenTelemetry proto, for identification of the metric data.

2. Zabbix Agent Metrics 采集

Configuration file

```
# initExp is the expression that initializes the current configuration file
initExp: <string>
# insert metricPrefix into metric name: <metricPrefix>_<raw_metric_name>
metricPrefix: <string>
# expPrefix is executed before the metrics executes other functions.
expPrefix: <string>
# expSuffix is appended to all expression in this file.
expSuffix: <string>
# Datasource from Zabbix Item keys.
requiredZabbixItemKeys:
  - <zabbix item keys>
# Support agent entities information.
entities:
  # Allow hostname patterns to build metrics.
  hostPatterns:
    - <regex string>
  # Customized metrics Label before parse to meter system.
  labels:
    [- <labels> ]
# Metrics rule allow you to recompute queries.
metrics:
  [ - <metrics_rules> ]
```

```
10 # Unless required by applicable law or agreed to in writing, software
11 # distributed under the license is distributed on an "AS IS" BASIS,
12 # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 # See the License for the specific language governing permissions and
14 # limitations under the License.
15
16 metricPrefix: mc_zabbix_meter_vm
17 expSuffix: service(['host'], Layer.OS_LINUX)
18 multiColumn: true
19 entities:
20   hostPatterns:
21     - zabbix-skwtest
22   labels:
23   requiredZabbixItemKeys:
24     # cpu
25     - system.cpu.load[all,avg15]
26     - system.cpu.load[all,avg1]
27     - system.cpu.load[all,avg5]
28     - system.cpu.util[,guest]
29     - system.cpu.util[,guest_nice]
30     - system.cpu.util[,idle]
31     - system.cpu.util[,interrupt]
32     - system.cpu.util[,iowait]
33     - system.cpu.util[,nice]
34     - system.cpu.util[,softirq]
35     - system.cpu.util[,steal]
36     - system.cpu.util[,system]
37     - system.cpu.util[,user]
38   # memory
39   - vm.memory.size[available]
40   - vm.memory.size[pavailable]
41   - vm.memory.size[total]
42   - vm.memory.size[pavailable]
43   # swap
44   - system.swap.size[,free]
45   - system.swap.size[,total]
46   - system.swap.size[,used]
47   # file
48   - vfs.fs.inode[,used]
49   - vfs.fs.size[,total]
50   - vfs.fs.size[,used]
51   - vfs.dev.read[,ops,avg1]
52   - vfs.dev.write[,ops,avg1]
53
54 metrics:
```

3. 日志数据采集与LAL语言

Enforcer

Enforcer is another special sink that forcibly samples the log. A typical use case of enforcer is when you have configured a sampler and want to save some logs such as to save error logs even if the sampling mechanism has been configured.

```
filter {
  // ... parser

  sink {
    sampler {
      // ... sampler configs
    }
    if (parsed.level == "ERROR" || parsed.userId == "TestingUserId") { // sample error logs or testing users' logs (userId == "TestingUser")
      enforcer {
      }
    }
  }
}
```

4. SkyWalking OAP 自观测

Telemetry for backend

The OAP backend cluster itself is a distributed streaming process system. To assist the Ops team, we provide the telemetry for the OAP backend itself, also known as self-observability (so11y)

By default, the telemetry is disabled by setting `selector` to `none`, like this:

```
telemetry:
  selector: ${SW_TELEMETRY:none}
  none:
  prometheus:
    host: ${SW_TELEMETRY_PROMETHEUS_HOST:0.0.0.0}
    port: ${SW_TELEMETRY_PROMETHEUS_PORT:1234}
    sslEnabled: ${SW_TELEMETRY_PROMETHEUS_SSL_ENABLED:false}
    sslKeyPath: ${SW_TELEMETRY_PROMETHEUS_SSL_KEY_PATH:""}
    sslCertChainPath: ${SW_TELEMETRY_PROMETHEUS_SSL_CERT_CHAIN_PATH:""}
```

You may also set `Prometheus` to enable them. For more information, refer to the details below.

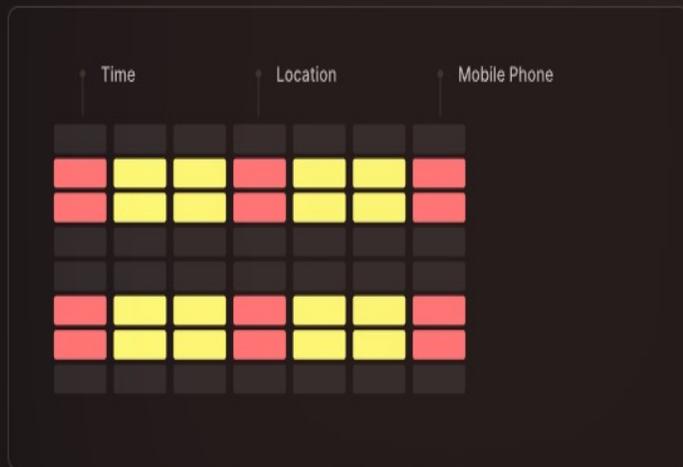
Self Observability

SkyWalking supports exposing telemetry data representing OAP running status through Prometheus endpoint. Users could set up OpenTelemetry collector to scrap and forward telemetry data to OAP server for further analysis, eventually showing up UI or GraphQL API.

02

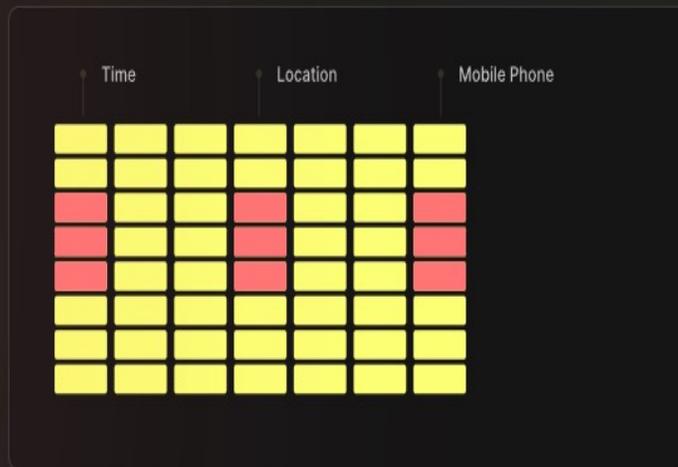
SkyWalking + ClickHouse

1. 为什么选择 ClickHouse



Row-oriented databases

In row-oriented databases, data is stored in rows, with all the values related to a row physically stored next to each other.

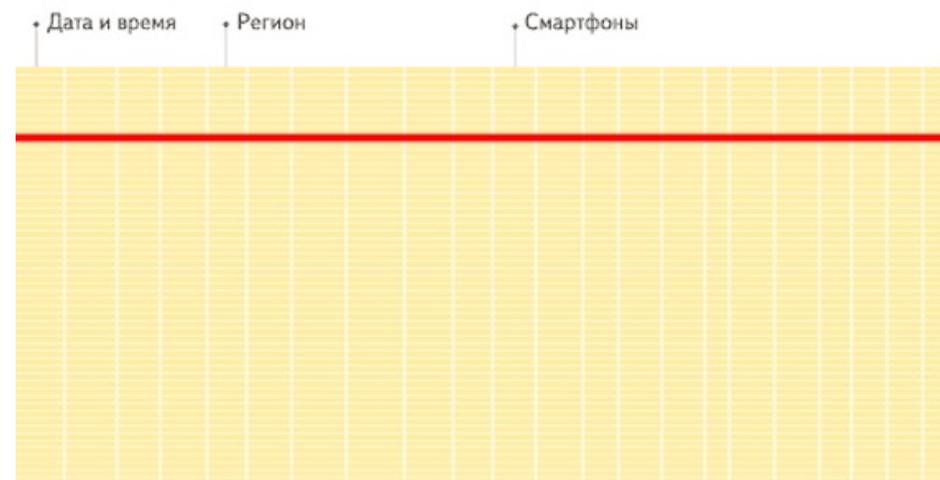
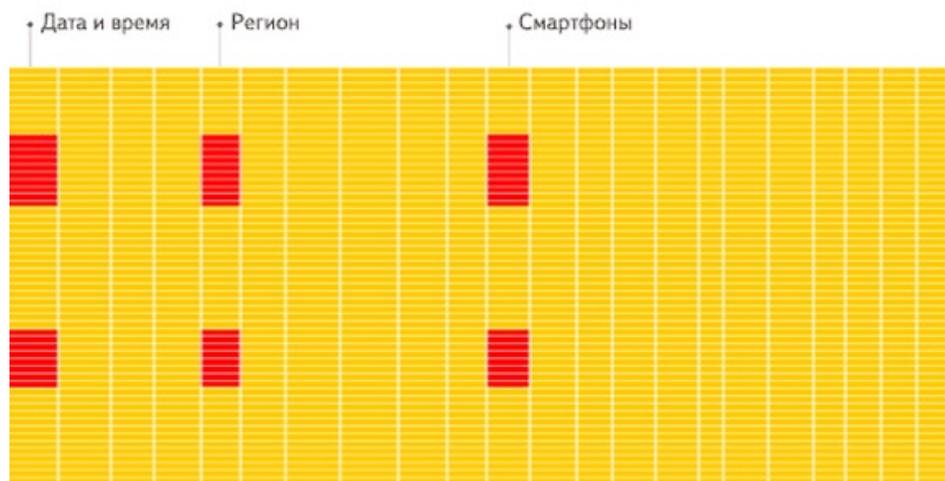


Column-oriented databases

In column-oriented databases, like ClickHouse, data is stored in columns, with values from the same columns stored together.

- 1 列式存储
- 2 高压缩率
- 3 支持数据合并
- 4 低基数值的优化

2. 列式存储



3. 基于列式存储重新设计指标存储

```
skywalking
├── public
│   └── 表
│       ├── instance_jvm_class_loaded_class_count
│       ├── instance_jvm_class_total_loaded_class_count
│       ├── instance_jvm_class_total_unloaded_class_count
│       ├── instance_jvm_cpu
│       ├── instance_jvm_memory_heap
│       ├── instance_jvm_memory_heap_max
│       ├── instance_jvm_memory_noheap
│       ├── instance_jvm_memory_noheap_max
│       ├── instance_jvm_normal_gc_count
│       ├── instance_jvm_normal_gc_time
│       ├── instance_jvm_old_gc_count
│       ├── instance_jvm_old_gc_time
│       ├── instance_jvm_thread_blocked_state_thread_count
│       ├── instance_jvm_thread_daemon_count
│       ├── instance_jvm_thread_live_count
│       ├── instance_jvm_thread_peak_count
│       ├── instance_jvm_thread_runnable_state_thread_count
│       ├── instance_jvm_thread_timed_waiting_state_thread_count
│       ├── instance_jvm_thread_waiting_state_thread_count
│       ├── instance_jvm_young_gc_count
│       └── instance_jvm_young_gc_time
│           └── meter_oap_instance_jvm_gc_count
│               ├── meter_oap_instance_jvm_memory_bytes_used
│               ├── meter_oap_instance_jvm_old_gc_time
│               └── meter_oap_instance_jvm_young_gc_time
```



```
mc_javaagent_instance_jvm 671
├── 列
│   ├── id (String)
│   ├── metric_prefix (LowCardinality(String))
│   ├── entity_id (LowCardinality(String))
│   └── service_id (LowCardinality(String))
├── 123 class_total_loaded_class_count_value (Int64)
├── 123 class_total_loaded_class_count_count (Int64)
├── 123 class_total_loaded_class_count_summation (Int64)
├── 123 class_total_unloaded_class_count_value (Int64)
├── 123 class_total_unloaded_class_count_count (Int64)
├── 123 class_total_unloaded_class_count_summation (Int64)
├── 123 class_loaded_class_count_value (Int64)
├── 123 class_loaded_class_count_count (Int64)
├── 123 class_loaded_class_count_summation (Int64)
├── 123 thread_timed_waiting_state_thread_count_value (Int64)
├── 123 thread_timed_waiting_state_thread_count_count (Int64)
├── 123 thread_timed_waiting_state_thread_count_summation (Int64)
├── 123 thread_waiting_state_thread_count_value (Int64)
├── 123 thread_waiting_state_thread_count_count (Int64)
├── 123 thread_waiting_state_thread_count_summation (Int64)
├── 123 thread_blocked_state_thread_count_value (Int64)
├── 123 thread_blocked_state_thread_count_count (Int64)
├── 123 thread_blocked_state_thread_count_summation (Int64)
├── 123 thread_runnable_state_thread_count_value (Int64)
├── 123 thread_runnable_state_thread_count_count (Int64)
├── 123 thread_runnable_state_thread_count_summation (Int64)
├── 123 thread_peak_count_value (Int64)
├── 123 thread_peak_count_count (Int64)
├── 123 thread_peak_count_summation (Int64)
├── 123 thread_daemon_count_value (Int64)
├── 123 thread_daemon_count_count (Int64)
├── 123 thread_daemon_count_summation (Int64)
├── 123 thread_live_count_value (Int64)
├── 123 thread_live_count_count (Int64)
├── 123 thread_live_count_summation (Int64)
├── 123 normal_gc_count_value (Int64)
├── 123 old_gc_count_value (Int64)
├── 123 young_gc_count_value (Int64)
├── 123 normal_gc_time_value (Int64)
├── 123 old_gc_time_value (Int64)
├── 123 young_gc_time_value (Int64)
├── 123 memory_noheap_max_value (Int64)
├── 123 memory_noheap_max_count (Int64)
├── 123 memory_noheap_max_summation (Int64)
├── 123 memory_heap_max_value (Int64)
├── 123 memory_heap_max_count (Int64)
├── 123 memory_heap_max_summation (Int64)
├── 123 memory_noheap_value (Int64)
├── 123 memory_noheap_count (Int64)
├── 123 memory_noheap_summation (Int64)
```

Instance_jvm 一个指标一张表

Instance_jvm 合并一张表

03

打通 SkyWalking + Zabbix

1. Zabbix对Java等中间件监控的痛点

- 1 Zabbix 监控 jvm 的信息必须要通过 JMX 协议来获取，这个强依赖JavaGateway 网关
- 2 对于多个 JVM 实例监控的配置比较麻烦
- 3 JVM 信息采集的不够全，很多涉及到接口性能的数据获取不到

2. Zabbix 基于PromQL采集指标数据

Item Tags Preprocessing

* Name

Type **HTTP agent**

* Key

Type of information

* URL

Query fields

Name	Value
<input type="text" value="name"/>	<input type="text" value="value"/>

Request type

* Timeout

Request body type

Request body

Headers

Name	Value
<input type="text" value="name"/>	<input type="text" value="value"/>

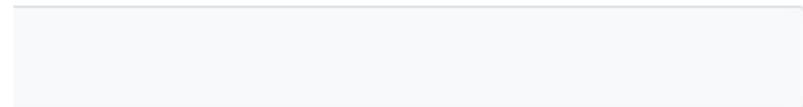
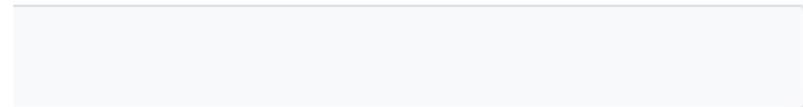
Required status codes

Follow redirects

Retrieve mode

Convert to JSON

HTTP proxy



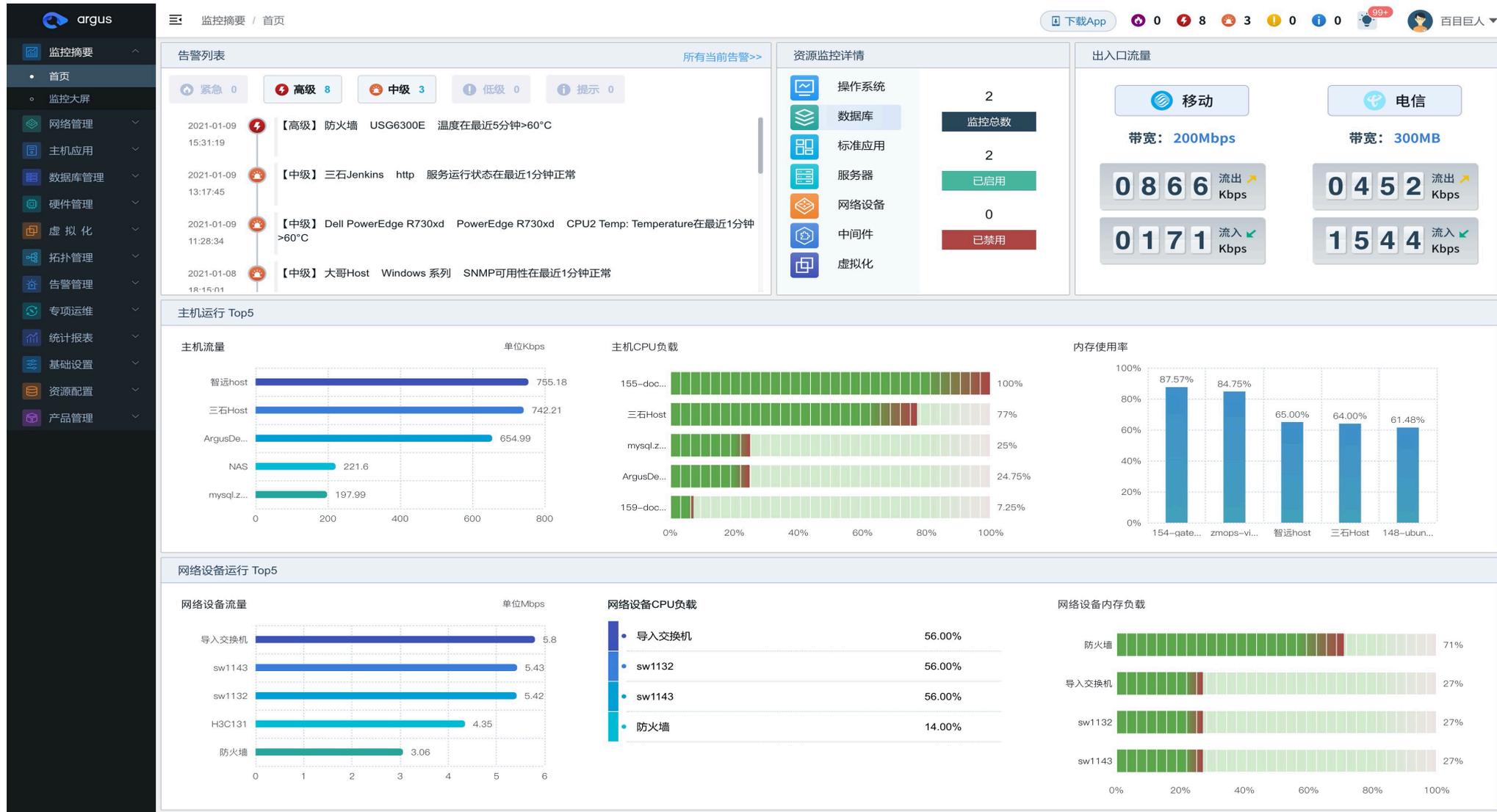
04

Argus 全域监控深度解析

Zabbix Plus = ArgusOMS

SkyWalking 二开 = ArgusAPM

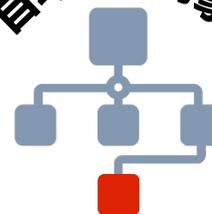
1. Argus 概览页



2. 全自动的网络动态拓扑



自动关联对象



实时展现

流量趋势

端口状态

CPU 负载

内存负载

关联告警

...

3. 服务主动异地拨测

全部拨测

状态速览

正常	故障	暂停
3	7	1

状态历史

名称	状态	日期时间	消息
拨测-win3.25	故障	2022-11-18 16:15:33	
拨测-linux2.147	故障	2022-11-18 16:11:30	
拨测-linux40.145	故障	2022-11-18 16:10:22	
拨测-ND03	正常	2022-11-18 15:55:13	
拨测-ND02	正常	2022-11-18 15:53:48	
拨测-AC	正常	2022-11-18 15:51:54	
拨测PING-3.184	故障	2022-11-18 14:55:06	
拨测PING-2.147	故障	2022-11-18 14:51:42	
拨测PING-3.132	故障	2022-11-18 14:14:00	
拨测-三石web服务	暂停	2022-11-02 18:28:04	
http 服务拨测	故障	2022-10-21 10:27:56	

共 11 条 < 1 > 前往 1 页

全部拨测

请输入关键字搜索

PING服务通用 拨测-win3.25 在线率: 24 % 故障

PING服务通用 拨测-linux2.147 在线率: 0 % 故障

PING服务通用 拨测-linux40.145 在线率: 0 % 故障

PING服务通用 拨测-ND03 在线率: 100 % 正常

PING服务通用 拨测-ND02 在线率: 100 % 正常

PING服务通用 拨测-AC 在线率: 100 % 正常

PING服务通用 拨测PING-3.184 在线率: 0 % 故障

PING服务通用 拨测PING-2.147 在线率: 0 % 故障

PING服务通用 拨测PING-3.132 在线率: 0 % 故障

HTTP服务通用 拨测-三石web服务 在线率: 0 % 暂停

共 11 条 < 1 2 > 前往 1 页

4. Syslog 分析



4. Syslog 分析

基础信息

名称

原始日志

_host

level

service

source

自动获取

手动输入

_rawMessage

前序步骤解析结果(自动生成)

▶ {}

定义解析规则

类型

字段

错误处理

高级选项

解析结果

▶ {}

定义解析规则

类型

字段

GROK 表达式

错误处理

高级选项

5. 设备配置文件备份

The screenshot displays the Argus network management interface. The main content area shows a table of network devices with columns for object name, IP address, management level, brand, unique ID, model, runtime, CPU usage, memory usage, monitoring method, alarm rules, maintenance personnel, and backup status. Two devices are listed: '纵目核心交换机' and '纵目出口设备'. A modal window titled '已备份配置文件' (Already backed up configuration files) is overlaid on the table, showing a list of backup records with columns for backup time, backup person, file name, and actions (download/delete). The modal also includes a '+ 新增备份' (Add new backup) button and pagination controls.

对象名称	IP地址	管理等级	品牌	唯一识别码	型号	运行时长	CPU使用率(%)	内存使用率(%)	监控方式	告警规则	运维人员	备份	操作
纵目核心交换机	172.1.4	二级	华为	21980108522S	S5720SV2-28P-LI-AC	66天,05:09:54	6	27	SNMPv2	Z	超级管...	2	命令行
纵目出口设备	172.1	二级	华为	1019...	USG6331E	66天,05:08:16	7	71	SNMPv2	Z	超级管...	0	命令行

备份时间	备份人	文件名称	操作
2023-11-02 13:48:05	超级管理员	20231102134805_...	下载 删除
2023-11-02 13:47:54	超级管理员	20231102134749_...	下载 删除

6. 原生内置了200+模板，开箱即用

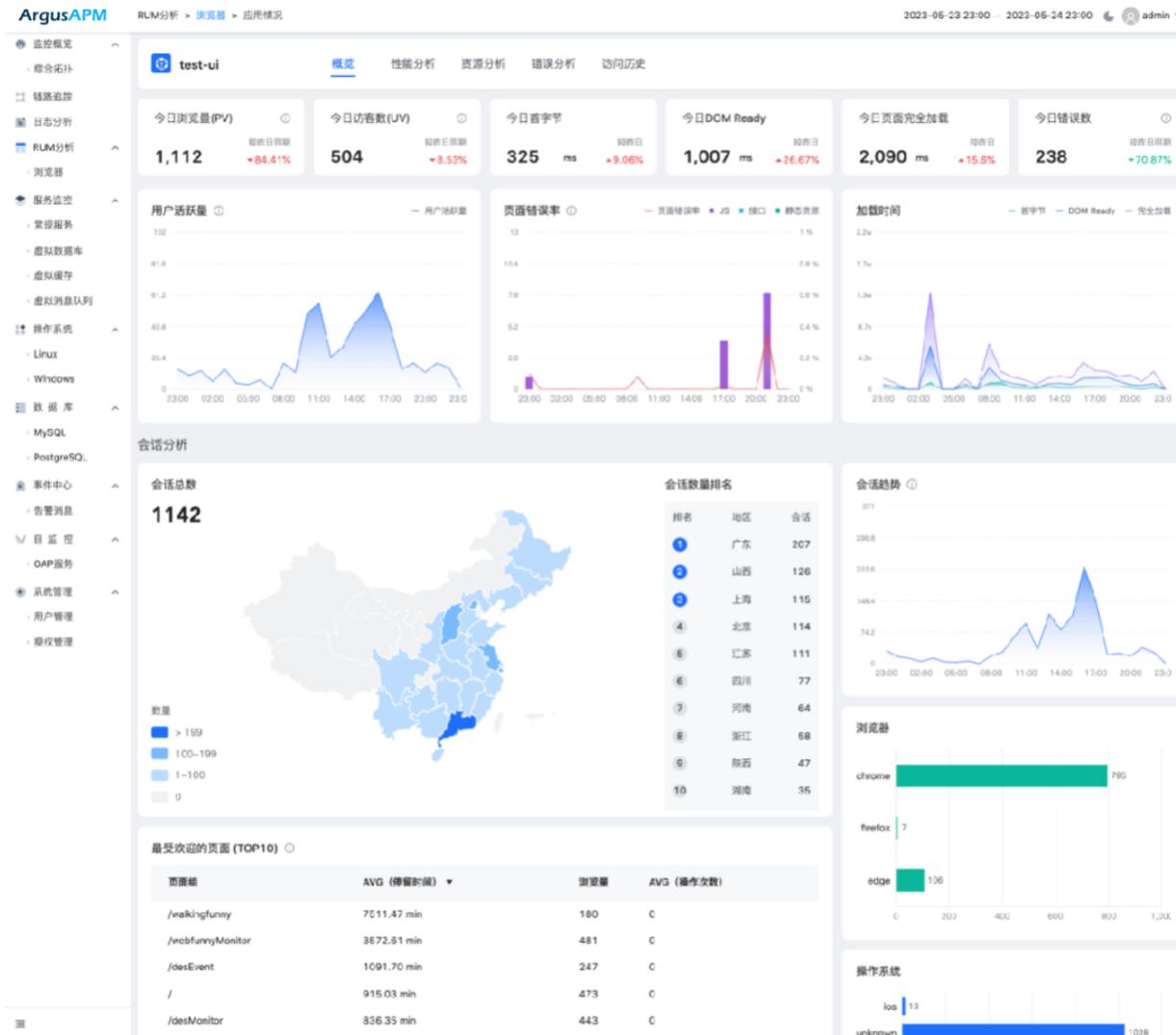
The screenshot shows the 'Resource Configuration / Add Resource' page in the Argus dashboard. The interface is divided into three main steps: 'Model/Version Selection', 'Monitoring Method Configuration', and 'Refining Maintenance Information'. A red box highlights a grid of device categories: Network Equipment (selected), Wireless Equipment, Servers, Storage Devices, Operating Systems, Virtualization, Databases, Middlewares, Standard Applications, and Terminal Devices. Below this is a search bar and a table of network devices.

选择	序号	型号	品牌	类型	系列
<input type="radio"/>	1	H3C E528	H3C	交换机	E552/528
<input type="radio"/>	2	H3C E552	H3C	交换机	E552/528
<input type="radio"/>	3	10508V	H3C	交换机	H3C 10500
<input type="radio"/>	4	10512	H3C	交换机	H3C 10500
<input type="radio"/>	5	10504	H3C	交换机	H3C 10500
<input type="radio"/>	6	10508	H3C	交换机	H3C 10500
<input type="radio"/>	7	1910-24G	H3C	交换机	H3C 1900
<input type="radio"/>	8	1910-24G-PoE-365W	H3C	交换机	H3C 1900
<input type="radio"/>	9	1910-8-PoE+	H3C	交换机	H3C 1900
<input type="radio"/>	10	1910-16G	H3C	交换机	H3C 1900

共 2374 条 | 25条/页 | 1 2 3 4 5 6 ... 95 | 前往 1 页

下一步

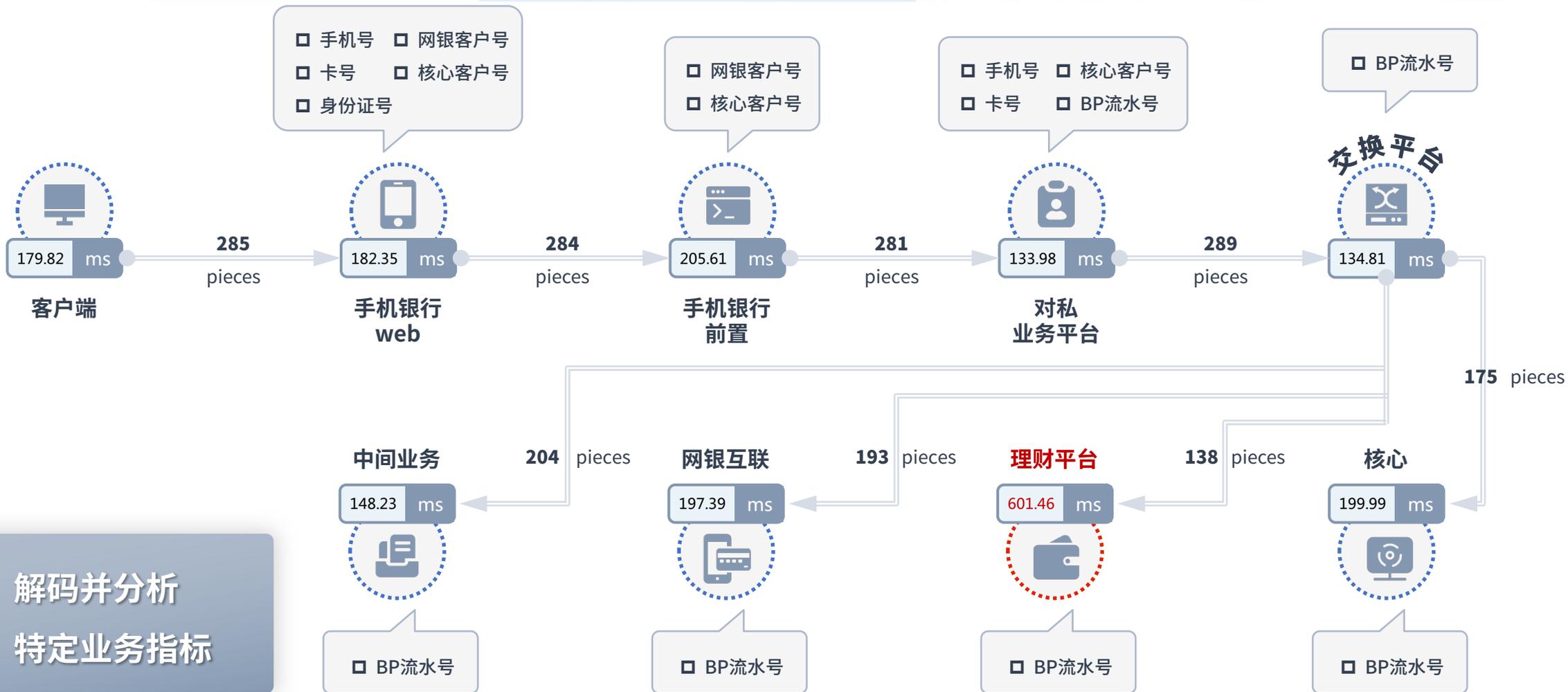
7. 端侧的用户访问体验数据



9. 业务链路 - 业务流探针自定义开发

< 01月10日 >

19:30



解码并分析
特定业务指标

10. 增强了链路追踪的分析

The screenshot displays the ArgusAPM interface for a distributed trace. The left sidebar shows navigation options like '链路追踪' (Distributed Tracing) and '火焰图' (Flame Graph). The main area shows a trace for the endpoint `POST:/system/auth/login` with a status of '正常' (Normal). The trace is visualized as a flame graph and a span list table.

Span ID	Operation	Duration	Percentage
1	index.html	709.34ms	34.67%
2	POST:/system/auth/login	664ms	34.21%
3	Druid/Connection/getConnection	0ms	0%
4	Mysql/JDBC/PreparedStatement/execute	1ms	0.03%
5	Druid/Connection/close	0ms	0%
6	Druid/Connection/getConnection	0ms	0%
7	Mysql/JDBC/PreparedStatement/execute	136ms	4.54%
8	Druid/Connection/close	0ms	0%
9	Druid/Connection/getConnection	0ms	0%
10	Mysql/JDBC/PreparedStatement/execute	1064ms	63.74%
11	Druid/Connection/close	0ms	0%
12	Redisson/SETEX	0ms	0%
13	Mysql/JDBC/Connection/commit	239ms	7.98%
14	Druid/Connection/close	0ms	0%

Service Execution Time Ratio:

Service	Ratio
esb-webui	40.35%
ESB-system	89.15%
MySQL	64.11%

链路详情 (Trace Details):

```
{ 21 items
  "traceId": "5be7ffeb-68ad-4edf-955f-76dc56ee3b59"
  "segmentId": "2467cbeb0e3b41f7918994da4cfc939.1623.16845014415947394"
  "spanId": 8
  "parentSpanId": 0
  "refs": [] 0 items
  "serviceCode": "ESB-system"
  "serviceInstanceName": "5610aa8e9fcd46af9cd2eefe9e8e0c566172.17.0.1"
  "startTime": 1684501441810
  "endTime": 1684501442012
```

11. 新增 ClickHouse 存储

数据优势



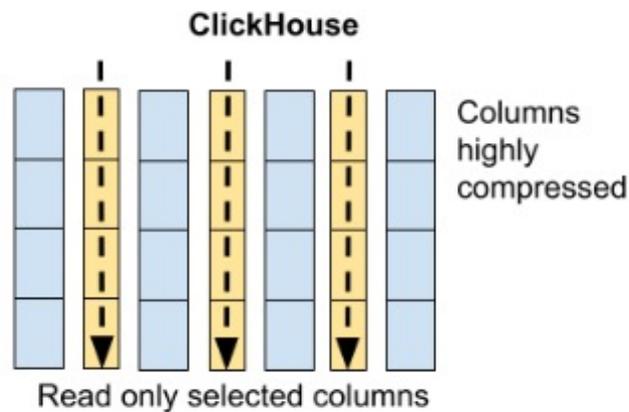
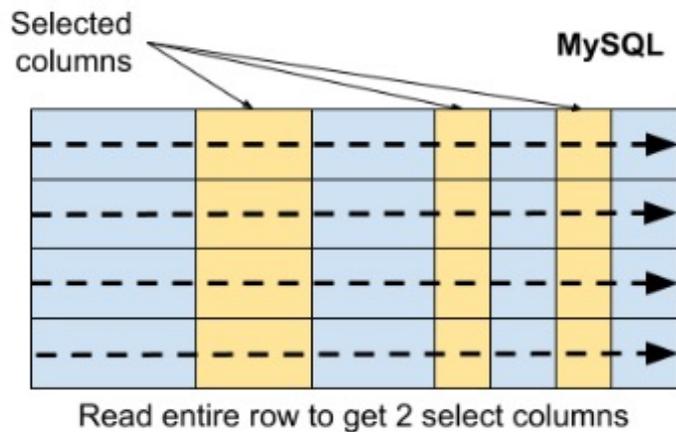
Batch 写入速率提高近 **一倍**，查询延迟全部在 **毫秒级**



单表多列进行压缩后，相较于 ES，存储成本仅需原本的 **10%**



在同等环境下，机器规模可以节省将近 **50%**



Q&A

欢迎提问交流
(仅限2位提问)



南天

中国大陆 上海



扫一扫上面的二维码图案，加我为朋友。



2023 | APACHE • SkyWalking
SUMMIT | CHINA · SHANGHAI

2023 • SkyWalking Summit

感谢您的观看

