见微知著, APM在tutorabc的应用

杨大鹏
iTutorGroup集团云平台(TGOP)负责人

2017-08-11

iTutorGroup集团, 让学习成为一种享受

tutorabc
真人在线 英语外教

tutorabc由美国硅谷技术团队研发创立，全球首个在线教育“独角兽”公司。阿里巴巴集团、新加坡淡马锡基金、启明创投和日本软银SBI集团一亿美元B轮融资。

GIC、中俄基金、高盛、银翎资本等世界级投资方近两亿美元C轮融资。

vipjr
青少儿在线教育

vipjr是由姚明代言的青少儿在线教育品牌，为5-18岁青少儿提供包括英语、数学、托福雅思等多元化教学服务。

2017年4月，vipjr联合哈佛大学、宾夕法尼亚大学等最高学府的顶尖思想领袖，成立美国学术顾问委员会。

TutorABC

tutorJr

TutorMing

LiveH2H

大纲

- 如何应对性能问题
- tutorabc对APM的一些实践
- TGOP私有云平台

关注性能，从现在开始！

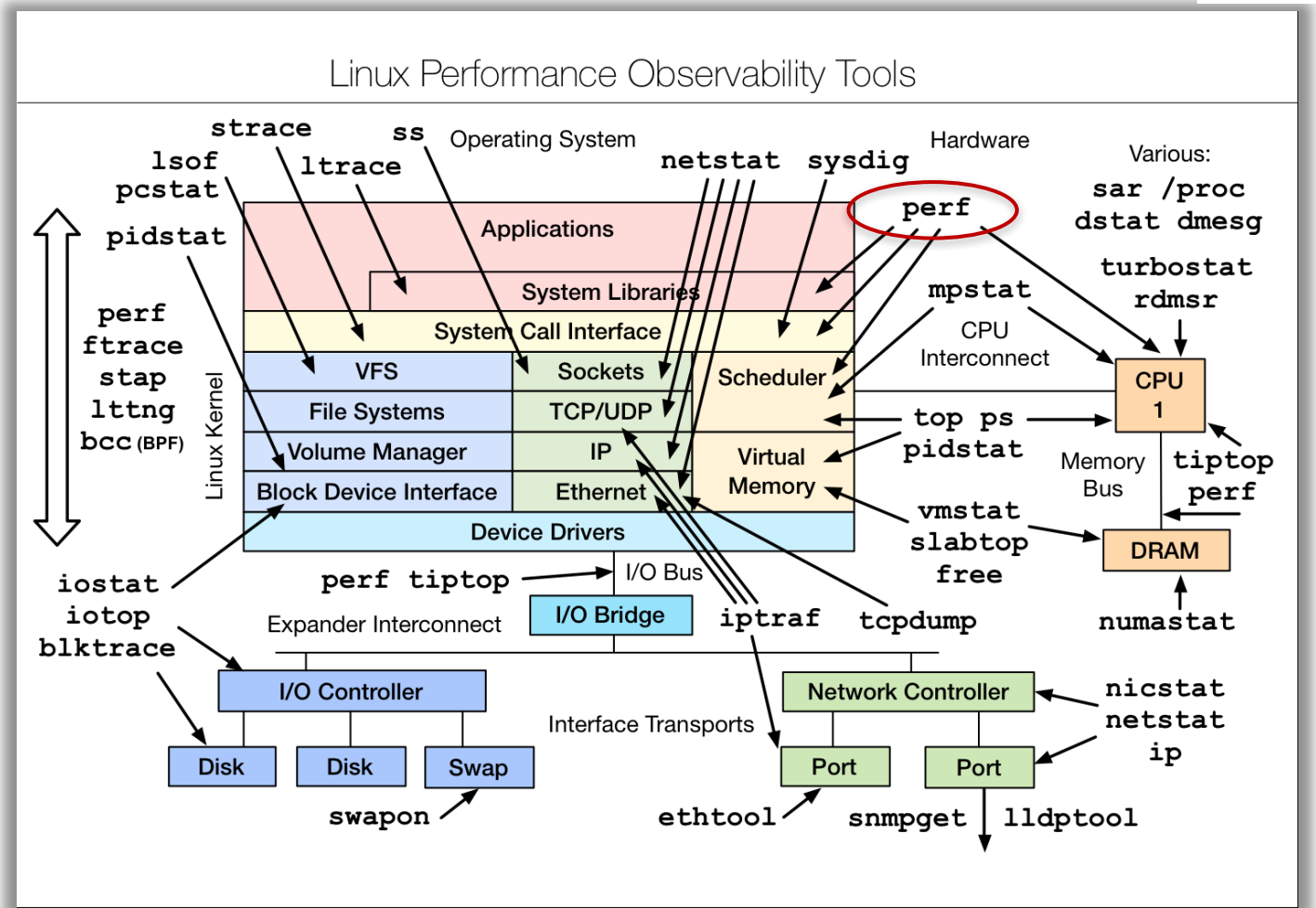
一个真实的调优案例

线上某API服务高峰时段CPU用量超过95%.
期间API响应变慢.
重启服务后，问题依旧存在.

之前通过添加新服务器缓解问题.
考虑到近期业务量有大幅增长，同时为节约计算资源，需要对服务进行优化.
PS: 本地环境可以重现该问题.

如何优化?

选择工具


Linux Performance Observability Tools
strace
ss
Operating System
Hardware
Various:
lsof
ltrace
netstat
sysdig
sar /proc
pcstat
perf
dstat dmesg
Applications
pidstat
turbostat
System Libraries
mpstat
rdmsr
perf
ftrace
System Call Interface
CPU
Interconnect
stap
VFS
Sockets
Scheduler
CPU
1
lttng
File Systems
TCP/UDP
top ps
bcc (BPF)
Linux Kernel
Volume Manager
IP
pidstat
Virtual
Memory
Memory
Bus
tiptop
Block Device Interface
Ethernet
perf
vmstat
Device Drivers
slabtop
DRAM
free
iostat
perf tiptop
I/O Bus
iotop
I/O Bridge
iptraf
tcpdump
numastat
blktrace
Expander Interconnect
I/O Controller
Network Controller
nicstat
Interface Transports
netstat
Disk
Disk
Swap
Port
Port
ip
swapon
ethtool
snmpget
lldptool

Windows平台下对应的工具

程序日志

JProfiler(Java), top, netstat等

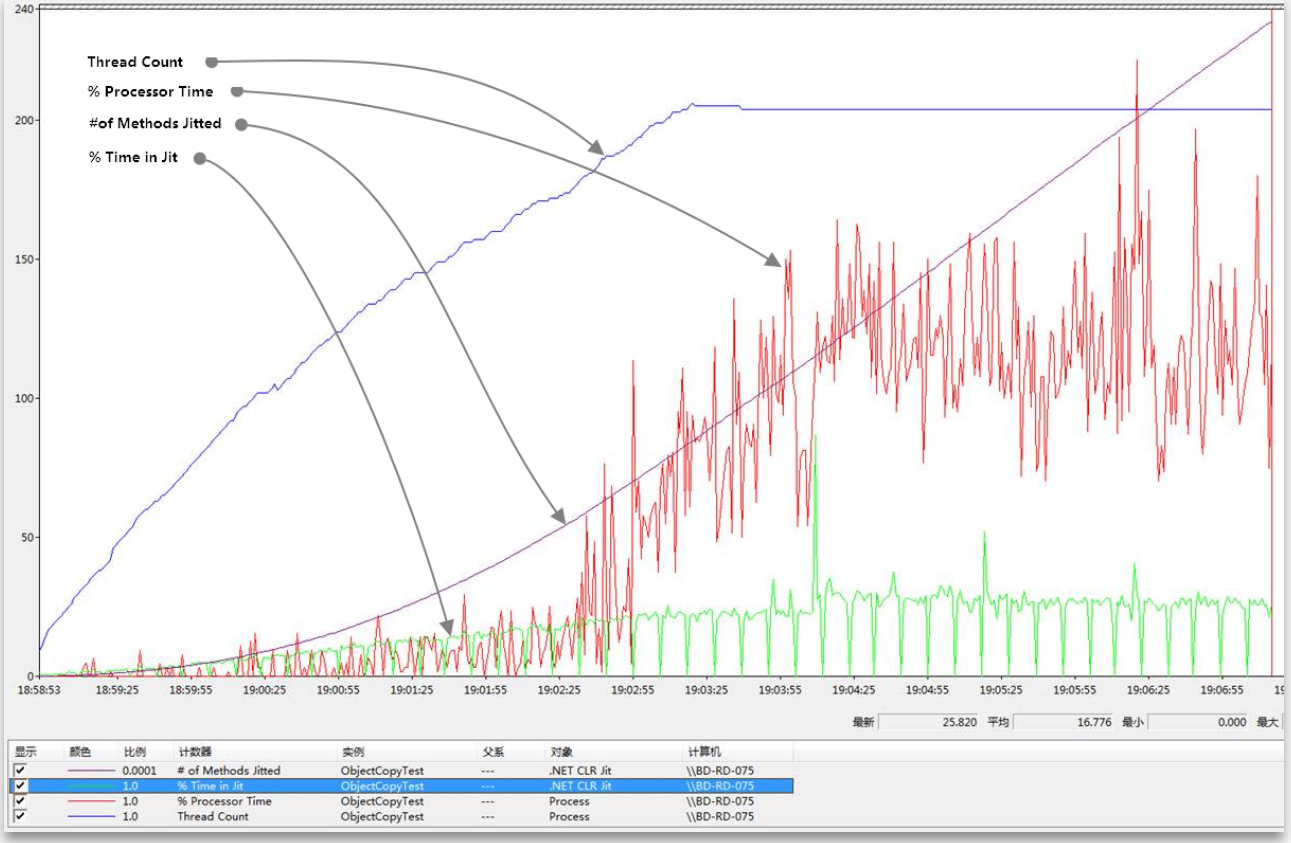
gdb, Perf {PMU, TracePoint}

程序日志

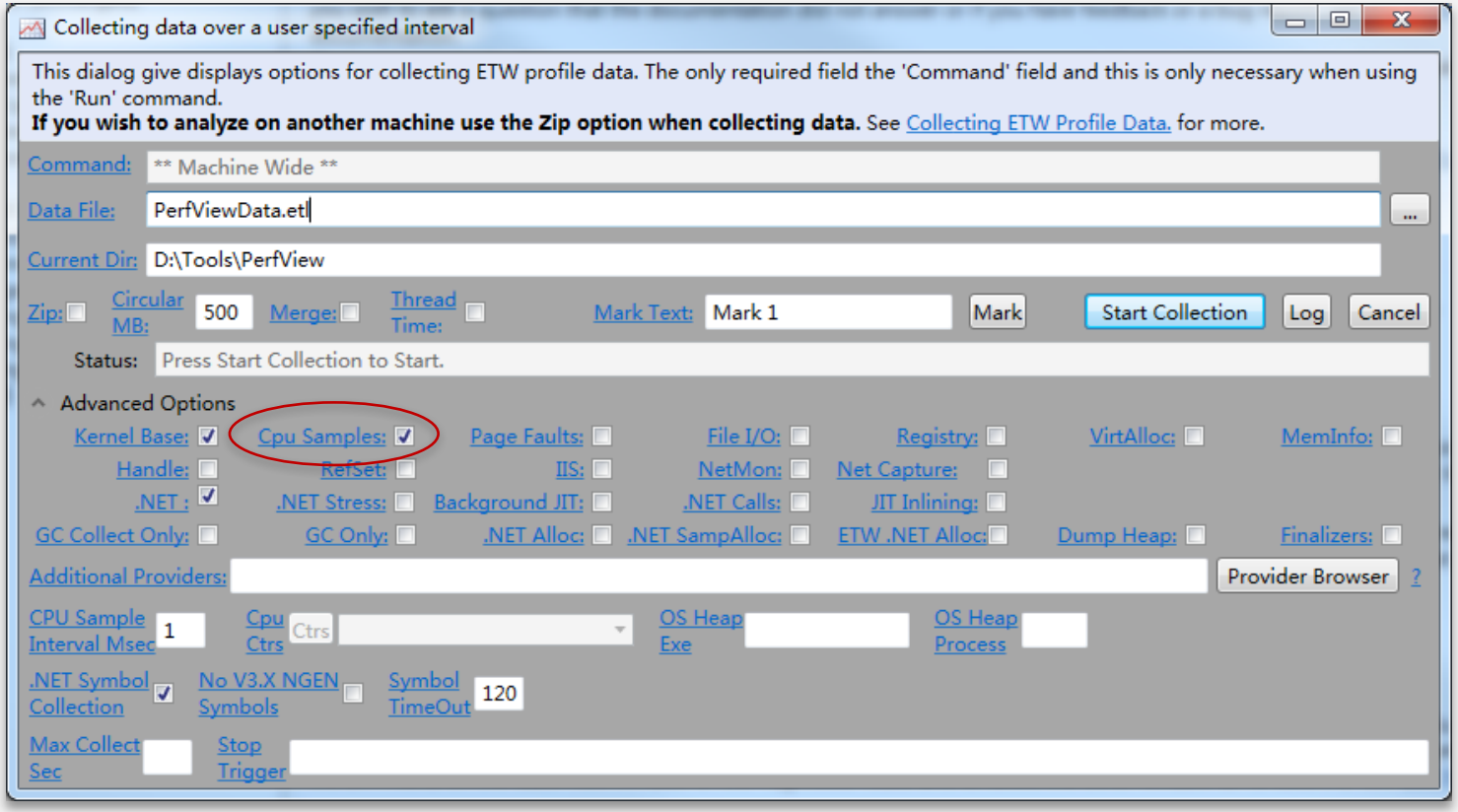
VS Profiler, Perfmon, Sysinternals Suite等

WinDbg, Perfview {ETW}

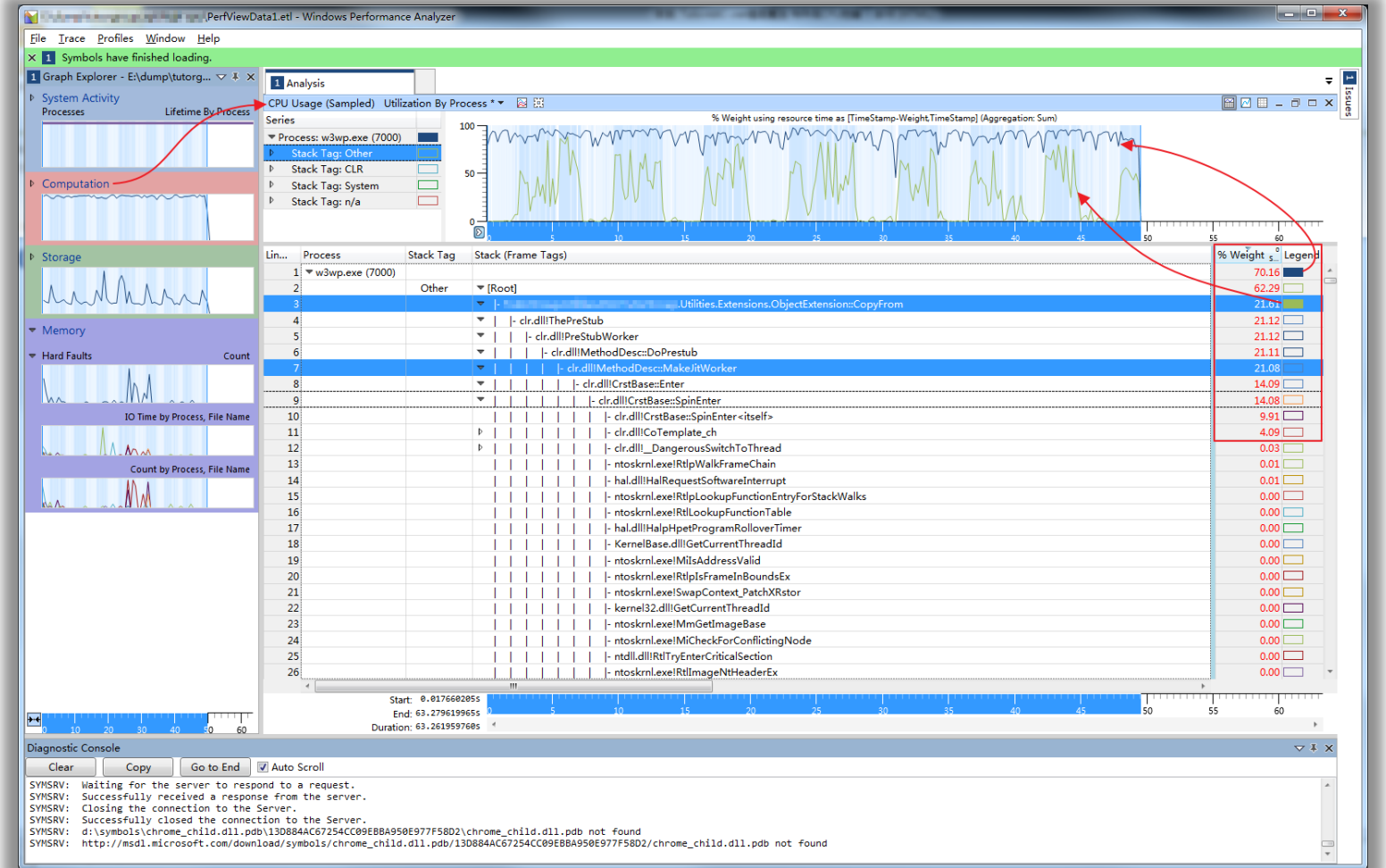
问题分析: 本地测试得到的性能参数



问题分析: PerfView采集ETW数据

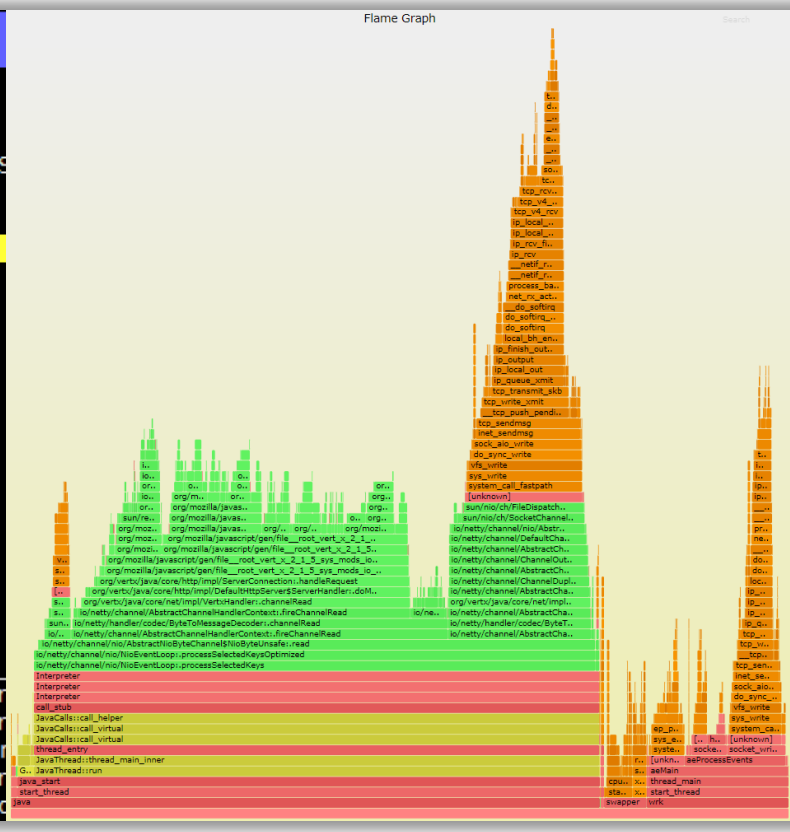


问题分析: XPerf查看CPU时间分布



如果是Perf...

```
Samples: 17K of event 'cpu-clock', Event count (approx.): 4477250000
  Children      Self  Command  Shared Object       Symbol
+   14.95%     0.15%  java     [kernel.kallsyms]   [k] tracesys
+   12.93%     0.04%  java     perf-3884.map       [.] 0x00007f1858392888
+   12.81%     0.00%  java     perf-3884.map       [.] 0x00007f1855472dfc
+   12.75%     0.01%  java     libjava.so          [.] Java_java_io_UnixFileS
+   11.98%     0.00%  java     libjvm.so           [.] java_start
+   11.98%     0.00%  java     libpthread-2.17.so  [.] start_thread
-   11.78%     0.00%  java     libjvm.so           [.] GangWorker::loop
   - GangWorker::loop
      - 11.77% ParNewGenTask::work
         - 9.94% GenCollectedHeap::gen_process_roots
            - 9.64% ConcurrentMarkSweepGeneration::younger_refs_iterate
                 CardTableRS::younger_refs_in_space_iterate
                 CardTableModRefBS::non_clean_card_iterate_possibly_parallel
               - CardTableModRefBS::non_clean_card_iterate_parallel_work
                  - 9.57% CardTableModRefBS::process_stride
                     + 4.92% ClearNoncleanCardWrapper::do_MemRegion
                     + 4.26% CardTableModRefBS::process_chunk_boundaries
         + 1.81% ParEvacuateFollowersClosure::do_void
+   11.77%     0.00%  java     libjvm.so           [.] ParNewGenTask::work
+   11.01%     0.02%  java     libjava.so          [.] canonicalize
+   10.96%     1.70%  java     libc-2.17.so        [.] __lxstat64
+    9.94%     0.00%  java     libjvm.so           [.] GenCollectedHeap::gen_
+    9.64%     0.01%  java     libjvm.so           [.] CardTableModRefBS::nor
+    9.64%     0.00%  java     libjvm.so           [.] CardTableModRefBS::nor
+    9.64%     0.00%  java     libjvm.so           [.] CardTableRS::younger_r
+    9.64%     0.00%  java     libjvm.so           [.] ConcurrentMarkSweepGen
+    9.57%     0.13%  java     libjvm.so           [.] CardTableModRefBS::pro
```



https://github.com/brendangregg/FlameGraph

问题分析: 定位问题代码

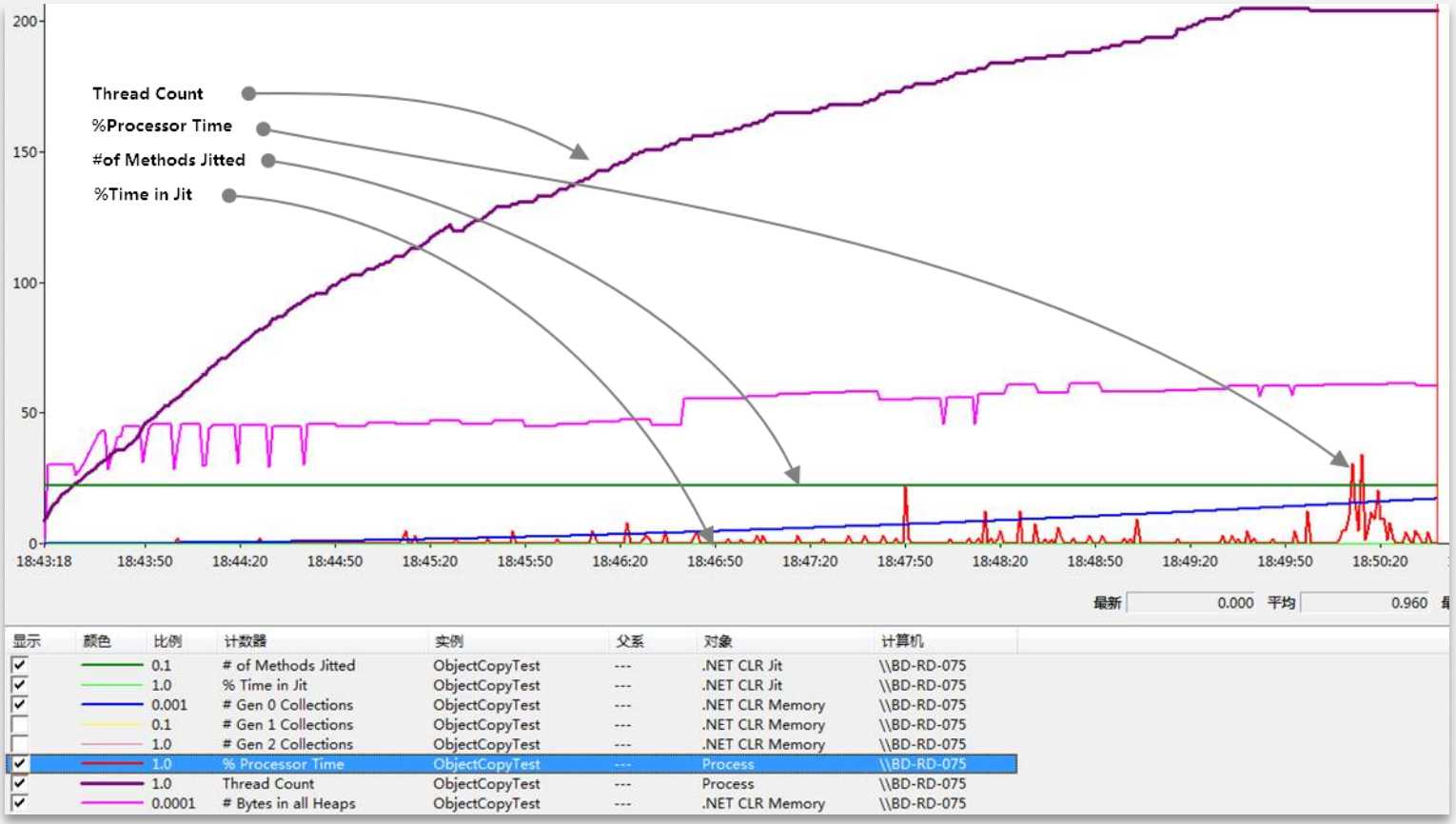
```
36          protected CallInfo CallInfo;
37          protected DynamicMe
38          protected EmitHelper
39
            8 references
40          protected BaseEmitte
44
            23 references
45          internal Delegate GetDelegate()
46          {
47              var action = cache.Get( CallInfo );
48              if( action == null )
49              {
50                  Method = CreateDynamicMethod();
51                  Generator = new EmitHelper( Method.GetILGenerator() );
52                  action = CreateDelegate();
53                  cache.Insert(CallInfo, action, CacheStrategy.Temporary);
54              }
55              return action;
56          }
```

```
95          public void Insert(TKey key, TValue value, CacheStrategy strategy)
96          {
97              entries[key] = strategy == CacheStrategy.Temporary
98                  ? new WeakReference(value)
99                  : value as object;
100         }
```

解决方案

```
36          protected CallInfo CallInfo;
37          protected DynamicMethod Method;
38          protected EmitHelper Generator;
39
            8 references
40          protected BaseEmitter( CallInfo callInfo )...
44
            23 references
45          internal Delegate GetDelegate()
46          {
47              var action = cache.Get( CallInfo );
48              if( action == null )
49              {
50                  Method = CreateDynamicMethod();
51                  Generator = new EmitHelper( Method.GetILGenerator() );
52                  action = CreateDelegate();
53                  cache.Insert(CallInfo, action, CacheStrategy.Permanent); //cache.Insert(CallInfo, action, CacheS
54              }
55              return action;
56          }
```

验证方案



显示	颜色	比例	计数器	实例	父系	对象	计算机
☑		0.1	# of Methods Jitted	ObjectCopyTest	---	.NET CLR Jit	\\BD-RD-075
☑		1.0	% Time in Jit	ObjectCopyTest	---	.NET CLR Jit	\\BD-RD-075
☑		0.001	# Gen 0 Collections	ObjectCopyTest	---	.NET CLR Memory	\\BD-RD-075
☐		0.1	# Gen 1 Collections	ObjectCopyTest	---	.NET CLR Memory	\\BD-RD-075
☐		1.0	# Gen 2 Collections	ObjectCopyTest	---	.NET CLR Memory	\\BD-RD-075
☑		1.0	% Processor Time	ObjectCopyTest	---	Process	\\BD-RD-075
☑		1.0	Thread Count	ObjectCopyTest	---	Process	\\BD-RD-075
☑		0.0001	# Bytes in all Heaps	ObjectCopyTest	---	.NET CLR Memory	\\BD-RD-075

性能调优的挑战与应对

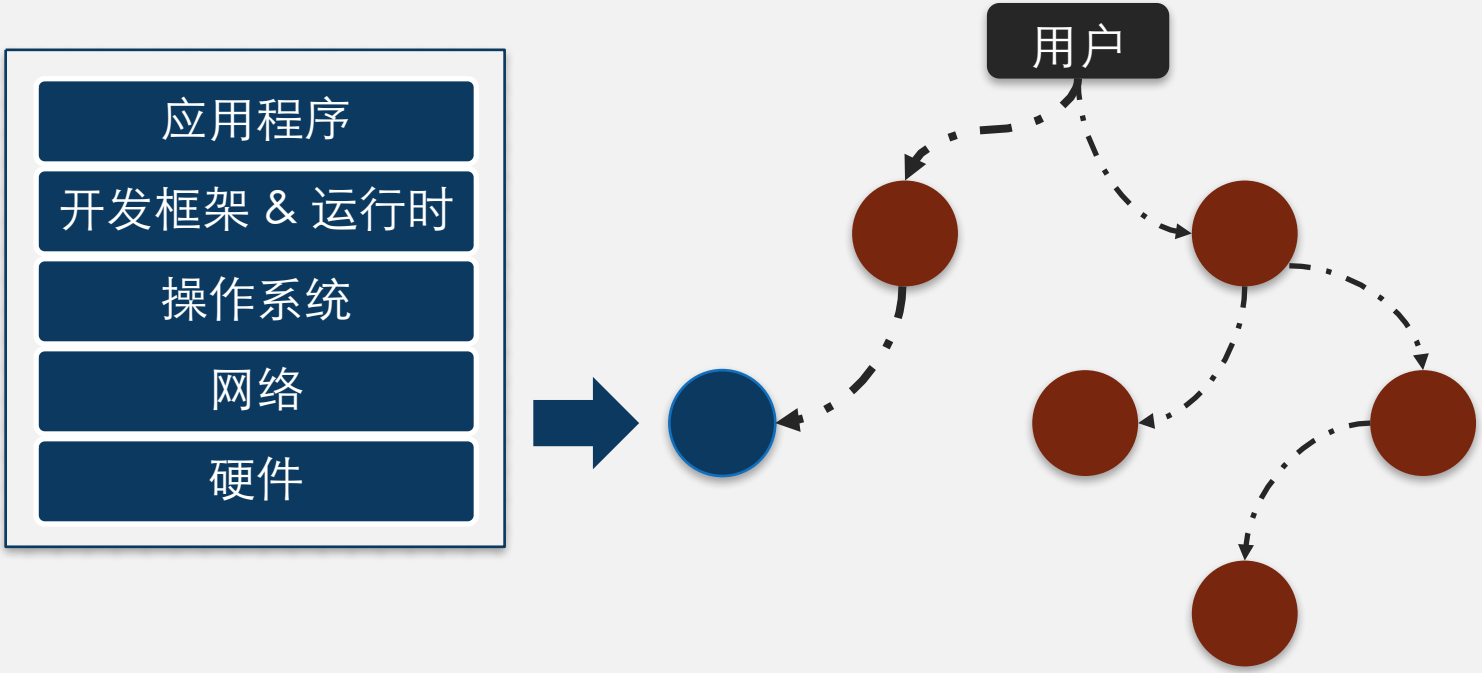
挑战

- 重现概率低，依赖环境.
- 日志分析，耗时耗力.
- 问题复杂，牵涉知识点众多.
- 解决方案需要进行验证.

应对

- 监控覆盖足够的广度和深度.
- 日志具备**多个维度**的关联性.
- 合适的**工具**，如Perf，Perfview等.
- **人**的经验与技术积累.
- 找出重现步骤，对比测试，有针对性的进行监控.

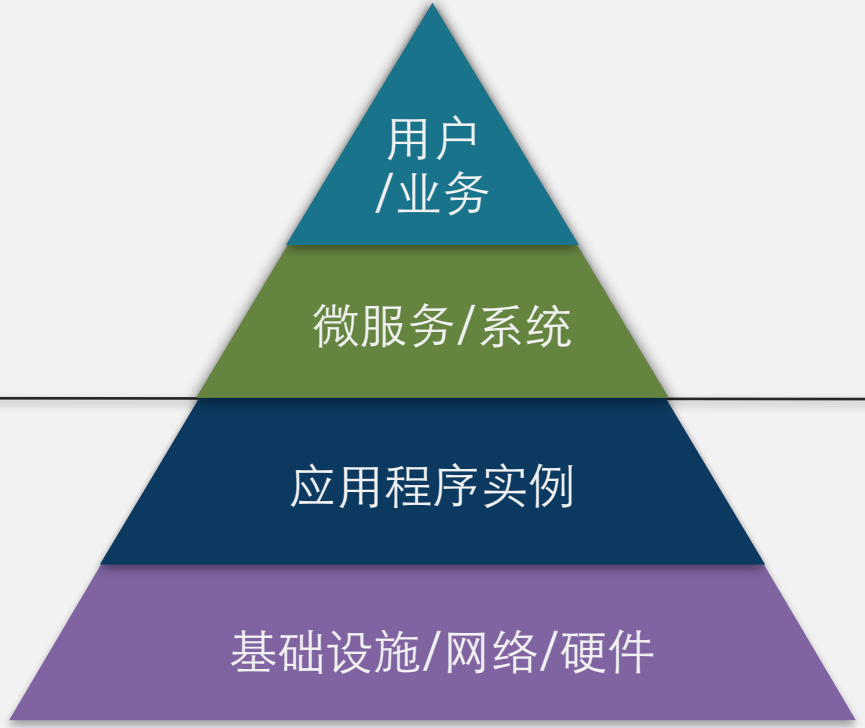
新的问题



我们需要掌控全局...

宏观

微观



追踪

日志

监控

在tutorabc, 我们在实践自己的APM

业务挑战

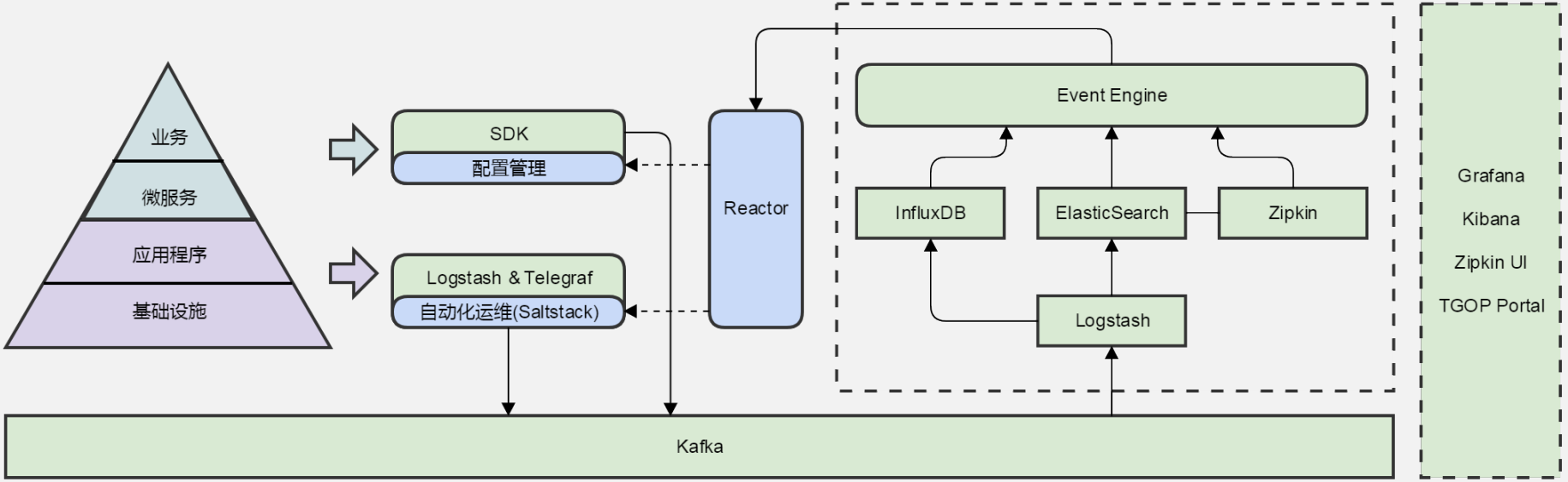
- 遍及全球135个国家和地区的客户群.
- 遍布全球80多个国家, 100多座城市的超过15,000名外籍顾问.
- 每年提供超过1000万堂在线课程.
- 为高质量客户群体提供高质量服务.

技术挑战

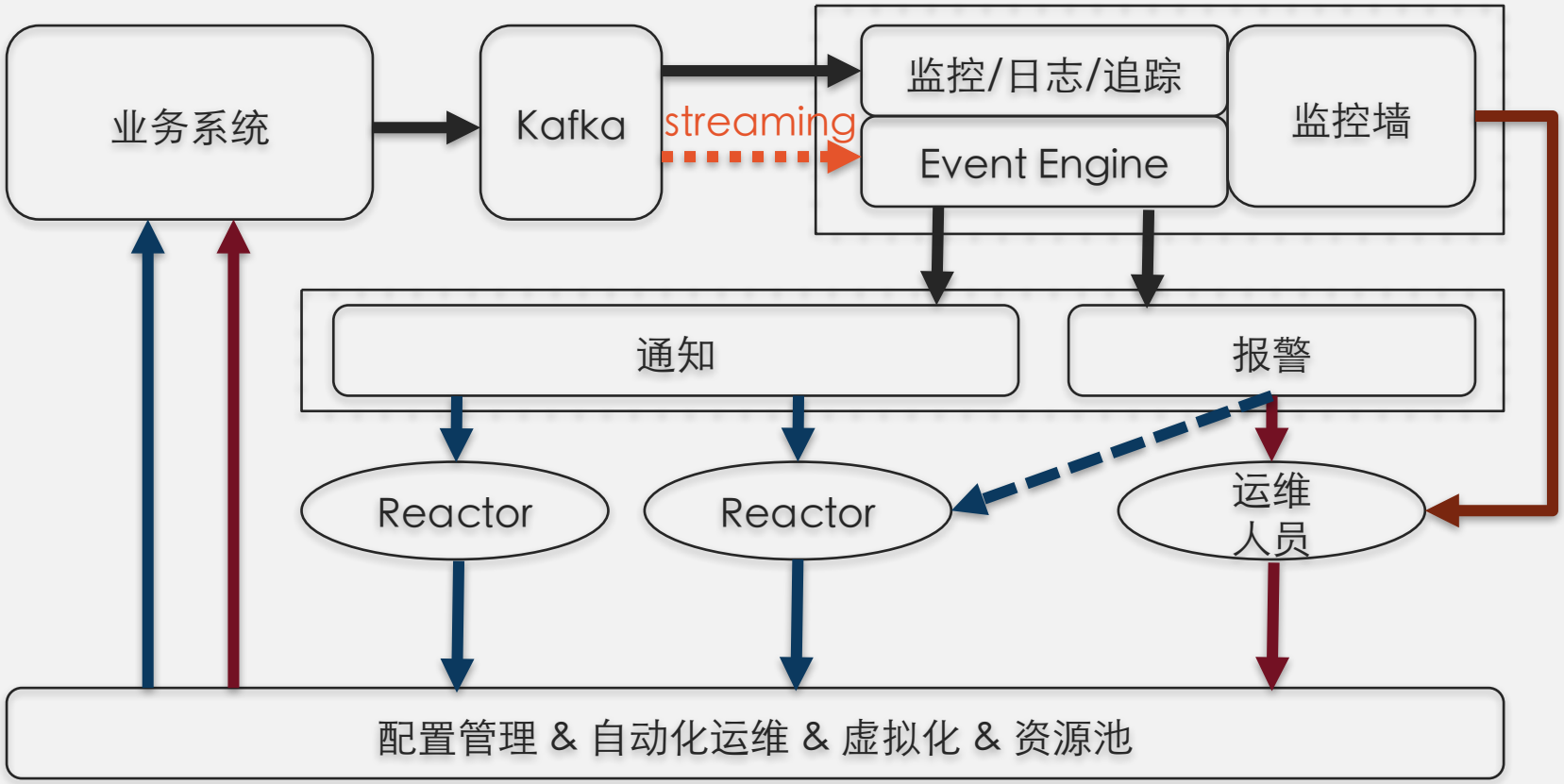
- 技术转型, 业务解耦.
- 多语言开发, 多平台部署.
- 遗留系统众多, 依赖关系复杂.

目标: 全系统APM驱动

APM系统构成


业务
微服务
应用程序
基础设施
SDK
配置管理
Logstash & Telegraf
自动化运维(Saltstack)
Reactor
Event Engine
InfluxDB
ElasticSearch
Zipkin
Logstash
Kafka
Grafana
Kibana
Zipkin UI
TGOP Portal

反馈机制


业务系统
Kafka
streaming
监控/日志/追踪
Event Engine
监控墙
通知
报警
Reactor
Reactor
运维
人员
配置管理 & 自动化运维 & 虚拟化 & 资源池

请求追踪

数据整合

展示与追踪链关联的日志，监控和报警数据，以便于问题排查.

系统全景

实时展示各个服务节点的状态，以及相互依赖关系.

动态采样

根据宿主机(以及调用链上游服务器)的负载调整采样率.

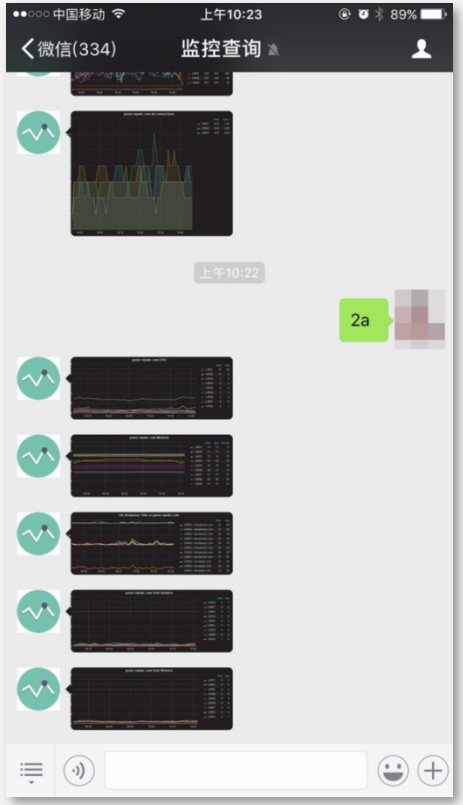
监控墙

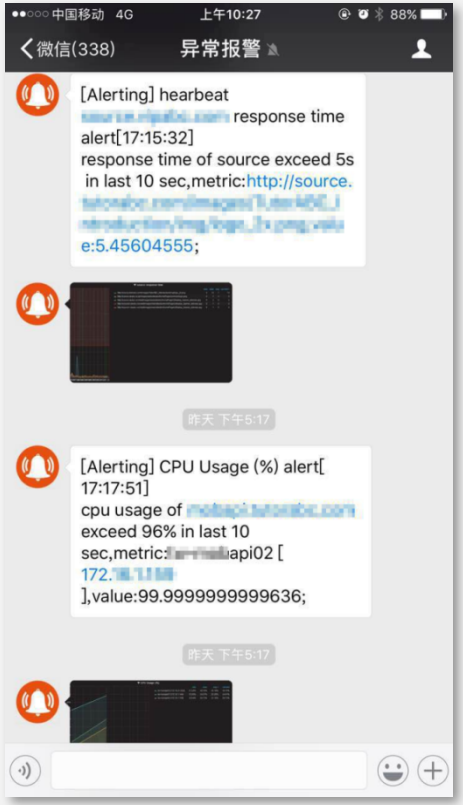

Infra
GroupByTimeInterval
auto
DataCenter
TGOP_Infra
IP
All
Role
All
Environment
CPU
max
avg
Network (KB/Sec)
IISLOG_
1264
1201
628
629
546
api
api1
158
api1
65
api3
735
api4
1226
api5
617
12
14

ChatOps


中国移动 上午10:21 89%
微信(333) 监控查询
站点列表：
1、
2、
3、
4、
5、
6、
7、
8、
9、
10、
11、
12、
13、
14、
15、
16、
17、
18、
19、
20、
21、
22、
23、
24、
25、
您可以输入[站台编号（数字）]+[指标代码（字母）]来查看相关站台的监控信息，


中国移动 上午10:22 89%
微信(333) 监控查询
24、
25、
您可以输入[站台编号（数字）]+[指标代码（字母）]来查看相关站台的监控信息，
指标编码：
A:基础指标
B:应用指标
C:应用日志
举例：1A 查看VWeb的基础指标
昨天 下午7:46
Web
Web
HAPI
WAPI
More...
基础指标
应用指标
应用日志


中国移动 上午10:23 89%
微信(334) 监控查询
上午10:22
2a


中国移动 4G 上午10:27 88%
微信(338) 异常报警
[Alerting] hearbeat
response time
alert[17:15:32]
response time of source exceed 5s in last 10 sec,metric:http://source.
e:5.45604555;
昨天 下午5:17
[Alerting] CPU Usage (%) alert[17:17:51]
cpu usage of
exceed 96% in last 10 sec,metric:
api02 [
172.
],value:99.9999999999636;
昨天 下午5:17

一些建议

- 监控，日志和追踪**数据要具备关联性**，便于后续分析.
- **尽早建立CMDB**，方便对被监控的服务器和服务做分类，打标签.
- 重SDK，去中心化.
- InfluxDB**适合存储以数字**为主的记录，**文本较多的记录存入ES**.
- InfluxDB**需要设置Series上限**，防止服务器内存耗尽.

一些建议(续)

- Grafana报警功能使用单独的实例来运行.
- Grafana数据库Annotation表需要定期清理，当数据量累积到30万以上时，会严重影响页面性能.
- 追踪系统无法跨越消息队列，可以在发送消息前将TraceID/SpanID信息植入消息内.
- 追踪系统SDK一般支持跨线程追踪，但是对Timer线程需要清除追踪上下文.
- 追踪系统基于采样率模式，所以无法指定追踪特定事件(比如登陆失败)，需要调研基于事件的追踪模式.

关于APM系统的一些数字

服务实例
1000+

数据中心
3

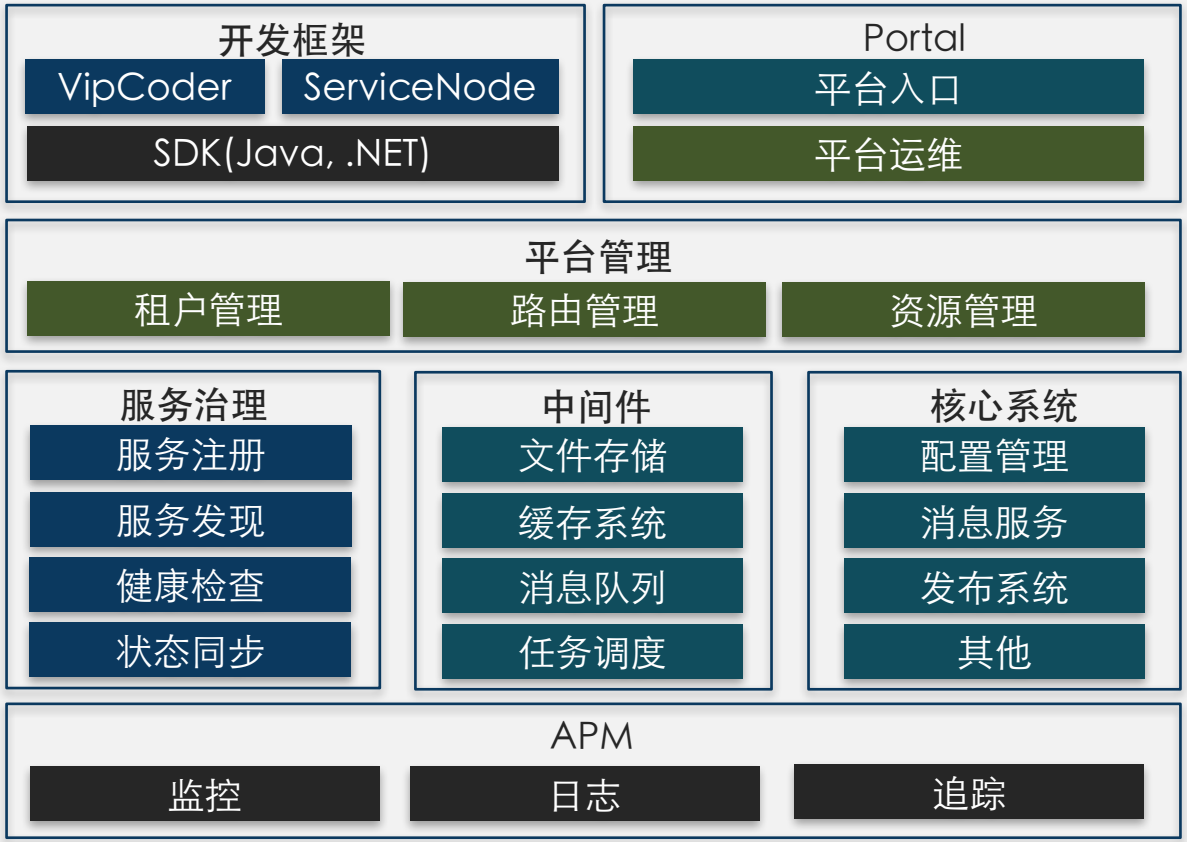
监控指标
100+

日志
500+GB/天

每秒查询
50+

每秒写入(单节点)
10,000+

服务器
600+


开发框架
VipCoder
ServiceNode
SDK(Java, .NET)
Portal
平台入口
平台运维
平台管理
租户管理
路由管理
资源管理
服务治理
服务注册
服务发现
健康检查
状态同步
中间件
文件存储
缓存系统
消息队列
任务调度
核心系统
配置管理
消息服务
发布系统
其他
APM
监控
日志
追踪

TGOP Portal


TGOPPortal
Search for...
Dashboard
Applications
DASHBOARD
主页 / Das
HEALTH TREND
Application Health Status
18:00 19:00 20:00 21:00 22:00 23:00 00:00 01:00 02:00 03:00 04:00 05:00 06:00 07:00 08:00 09:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00
ALERTS
[Alerting]
[Alerting]
[Alerting]
[OK] My
[Alerting]
66%
文件
GB
49%
缓存
GB
73%
日志
GB
RESOURCE USAGE
名称
规格
上限
用量
数据中心
昨日用量
前一小时用量
日志
A1
100GB
缓存
A1
2GB

展望

- **改进追踪系统UI**，增强与日志和监控(报警)系统的集成体验.
- 基于追踪系统，实现**分业务场景监控**.
- 基于大数据进行未来24小时流量预测，做到**主动预警**而非被动告警.
- **应用画像**，以利精准部署.
- **资源池化**，提高利用率.

总结

未有监控，不谈优化.

合适即可，演化迭代.

工具重要，人更重要.

少即是多，见微知著.

个人介绍

杨大鹏

原微软工程师, 2015年加入tutorabc,

带队负责TGOP平台设计, 实现, 运维与推广.

熟悉CLR/JVM虚拟机,

擅长Windows/Linux平台下应用性能调优.