



运维世界大会·深圳站

Zabbix高级玩法

吴兆松

关于我

吴兆松 @itnihao

OneOaaS技术负责人

Zabbix用户, 《Zabbix企业级分布式监控系统》作者

运维人员, Go python shell用户

从业经历有大型ISP Cache系统建设、互联网游戏、O2O运维, 大型企业私有云建设、大型企业监控系统建设

目前专注于运维工具平台的开发设计工作

mail: itnihao@qq.com

目录



自动化配置



故障与警告



趋势预测



性能优化

Zabbix可以支持的监控类型

- Agent
- SNMP Agent
- IPMI Agent
- Agentless Monitoring
- Web Monitoring
- Database Monitoring
- Internal Check
- Calculated Monitoring
- Custom Command Monitoring

Agent监控功能

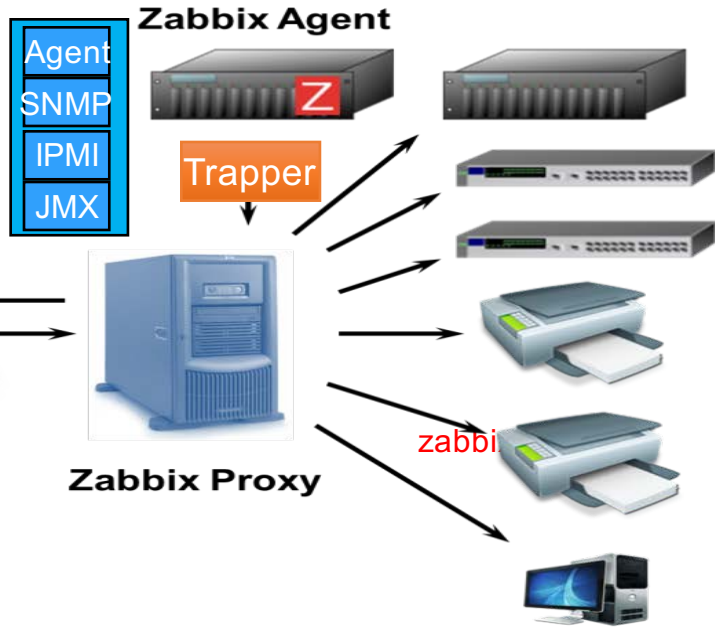
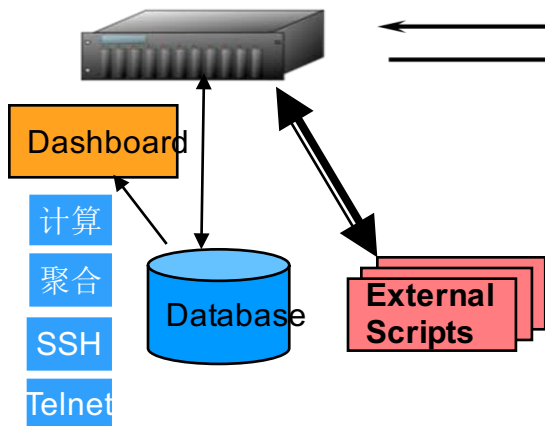
CPU	Load Average
	CPU Utilization
Memory	Memory Utilization
	Swap/Pagefile Utilization
Network	Network Transfer
	Network Error/Drop Packet
	Collisions
Disk	Filesystem Utilization
	Disk I/O
Service	Process Monitoring
	Windows Service
	TCP Port connectivity
	TCP Port response time
	DNS Monitoring
Log	NTP Monitoring
	Text Log
File	Eventlog
	File Monitoring
Other	Performance Counter (Windows only)

Zabbix的分布式架构

跨机房、跨地区分布式部署

使用场景: 一个中心管理节点, 多个分部, 多个机房

Zabbix Server



自动发现-Discovery

网络发现

条件

定期搜索IP+端口+特定条件范围的回复信息

动作

创建/删除主机和模板，远程命令/开关主机模板/添加分组

自动注册

条件

根据Porxy信息/agent信息判断是否添加

动作

创建/删除主机和模板，开关主机模板/添加分组

低级别发现

条件

定期发现主机上面丢失的监控元素

动作

创建/删除监控项

自动

智能

省心
省力

新设备上线!

新分区加入!

新网卡加入!

新元素加入!

多端口加入!

LLD-Low Level Discovery

```

root@localhost ~]# zabbix_get -s 127.0.0.1 -k vfs.fs.discovery
{
  "data": {
    {
      "#FSNAME": "\\/",
      "#FSTYPE": "rootfs",
    },
    {
      "#FSNAME": "\\/proc",
      "#FSTYPE": "proc",
    },
    {
      "#FSNAME": "\\/sys",
      "#FSTYPE": "sysfs",
    },
    {
      "#FSNAME": "\\/dev",
      "#FSTYPE": "devtmpfs",
    },
    {
      "#FSNAME": "\\/dev/pts",
      "#FSTYPE": "devpts",
    },
    {
      "#FSNAME": "\\/dev/shm",
      "#FSTYPE": "tmpfs",
    },
    {
      "#FSNAME": "\\/",
      "#FSTYPE": "ext4",
    },
    {
      "#FSNAME": "\\/proc/bus/usb",
      "#FSTYPE": "usbfs",
    },
    {
      "#FSNAME": "\\/boot",
      "#FSTYPE": "ext4",
    },
    {
      "#FSNAME": "\\/proc/sys/fs/binfmt_misc",
      "#FSTYPE": "binfmt_misc",
    },
    {
      "#FSNAME": "\\/misc",
      "#FSTYPE": "autofs",
    },
    {
      "#FSNAME": "\\/net",
      "#FSTYPE": "autofs",
    }
  }
}
    
```

Discovery rule **Filters**

Parent discovery rules: Template OS Linux

Name: Mounted filesystem discovery

Type: Zabbix agent

Key: vfs.fs.discovery

Host interface: 127.0.0.1 : 10050

Update interval (in sec): 3600

Flexible intervals: INTERVAL PERIOD ACTION
No flexible intervals defined.

New flexible interval: Interval (in sec) 50 Period 1-7,00:00-24:00 Add

Keep lost resources period (in days): 30

Description: Discovery of file systems of different types as defined

Zabbix server: Disk space usage / (1h)



■ Total disk space on / [avg]
■ Free disk space on / [avg]

Data from history. Generated in 0:05 on

■ Value: 291.13 GB (100%)
■ Value: 271.85 GB (93.38%)

SEVERITY **NAME** ▲

Warning **Template OS Linux: Free disk space is less than 20% on volume (#FSNAME)**

Hosts / web-app-01 Enabled ZBX SNMP JMX IPMI Discovery list / Mounted filesystem discovery Item prototypes 5

NAME ▲	KEY
Template OS Linux: Free disk space on (#FSNAME)	vfs.fs.size{#FSNAME},free
Template OS Linux: Free disk space on (#FSNAME) (percentage)	vfs.fs.size{#FSNAME},pfree
Template OS Linux: Free inodes on (#FSNAME) (percentage)	vfs.fs.inode{#FSNAME},pfree
Template OS Linux: Total disk space on (#FSNAME)	vfs.fs.size{#FSNAME},total
Template OS Linux: Used disk space on (#FSNAME)	vfs.fs.size{#FSNAME},used

Zabbix server: Disk space usage /boot (1h)



■ Total disk space on /boot [avg]
■ Free disk space on /boot [avg]

Data from history. Generated in 0:07 on

■ Value: 189.69 MB (100%)
■ Value: 148.53 MB (78.30%)

Graph **Disk space usage /boot** ▼

- not selected
- CPU jumps
- CPU load
- CPU utilization
- Disk space usage /**
- Disk space usage /boot**
- Memory usage
- Network traffic on eth0
- percent_line95
- Swap usage

In Zabbix, six types of discovery items are supported out of the box:

- discovery of file systems;
- discovery of network interfaces;
- discovery of CPUs and CPU cores;
- discovery of SNMP OIDs;
- discovery using ODBC SQL queries;
- discovery of Windows services.

Multi OID SNMP Discovery

```
discovery[#{MACRO1}, oid1, #{MACRO2}, oid2, ...]
```

Example: Discover ifDescr & ifAlias

```
{
  "data":[
    {#{SNMPINDEX}:"1",#{IFDESCR}:"Interface #1",#{IFALIAS}
    {#{SNMPINDEX}:"2",#{IFDESCR}:"Interface #2",#{IFALIAS}
    {#{SNMPINDEX}:"3",#{IFALIAS}:"eth3"},
    {#{SNMPINDEX}:"4",#{IFDESCR}:"Interface #4"},
    {#{SNMPINDEX}:"5",#{IFALIAS}:"eth5"}
  ]
}
```

Discover Windows services

```
service.discovery
```

- Return multiple macros
- Filter results

```
{#SERVICE.NAME}
{#SERVICE.DISPLAYNAME}
{#SERVICE.DESCRPTION}
{#SERVICE.STATE}
{#SERVICE.STATENAME}
{#SERVICE.PATH}
{#SERVICE.USER}
{#SERVICE.STARTUP}
{#SERVICE.STARTUPNAME}
```

ros Host inventory

Host macros

Inherited and host macros

MACRO	VALUE
{LOW_SPACE_LIMIT}	⇒ 10
{LOW_SPACE_LIMIT:/}	⇒ 20
{LOW_SPACE_LIMIT:/BOOT}	⇒ 5

自动发现的trigger阈值定制:

Set the free disk space trigger prototype for a host to:

```
{host:vfs.fs.size[#{FSNAME},pfree].last()}<{$LOW_SPACE_LIMIT:#{FSNAME}}
```

And add macros:

```
{LOW_SPACE_LIMIT} 10
```

```
{LOW_SPACE_LIMIT:/home} 20
```

```
{LOW_SPACE_LIMIT:/tmp} 50
```


JSON-RPC API gives access to nearly any functionality

```
POST http://company.com/zabbix/api_jsonrpc.php HTTP/1.1
Content-Type: application/json-rpc

{"jsonrpc": "2.0", "method": "apiinfo.version", "id": 1, "auth": null, "params": {}}
```

Example "host.get"

```
{
  "jsonrpc": "2.0",
  "method": "host.get",
  "params": {
    "output": [
      "hostid",
      "host"
    ],
    "selectInterfaces": [
      "interfaceid",
      "ip"
    ]
  },
  "id": 2,
  "auth": "0424bd59b807674191e7d77572075f33"
}
```

Request

=>

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10084",
      "host": "Zabbix server",
      "interfaces": [
        {
          "interfaceid": "1",
          "ip": "127.0.0.1"
        }
      ]
    }
  ],
  "id": 2
}
```

Response

Zabbix API

推荐项目pyzabbix

<https://github.com/lukecyca/pyzabbix>

```
from pyzabbix import ZabbixAPI

group='WEB'
template='APP_WEB_NGINX'

zapi = ZabbixAPI('http://127.0.0.1/zabbix')

# Login to the Zabbix API
zapi.login('Admin', 'zabbix')

group_id = zapi.hostgroup.getobjects(name=group)[0]['groupid']
template_id = zapi.template.getobjects(name=template)[0]['templateid']
zapi.host.create (
    {
        "host": 'web-01',
        "interfaces": [{
            "type": 1,
            "dns": "",
            "main": 1,
            "ip": '10.0.1.2',
            "port": 10050,
            "useip": 1,
        }],
        "groups": [{ "groupid": group_id }],
        "templates": [{ "templateid": template_id }],
    })
print('Host Added')
```

```
from pyzabbix import ZabbixAPI, ZabbixAPIException
import sys

zapi = ZabbixAPI('http://192.168.142.99')

# Login to the Zabbix API
zapi.login('api', 'api123456')

host_name = 'Zabbix server'

hosts = zapi.host.get(filter={"host": host_name},selectInterfaces=["interfaceid"])
if hosts:
    host_id = hosts[0]["hostid"]
    print("Found host id {0}".format(host_id))

    try:
        item=zapi.item.create(
            hostid=host_id,
            name='Used disk space on $1 in %',
            key_='vfs.fs.size[/,used]',
            type=0,
            value_type=3,
            interfaceid=hosts[0]["interfaces"][0]["interfaceid"],
            delay=30
        )
    except ZabbixAPIException as e:
        print(e)
        sys.exit()
    print("Added item with itemid {0} to host: {1}".format(item["itemids"][0],host_name))
else:
    print("No hosts found")
```

通过API构建DevOps生态



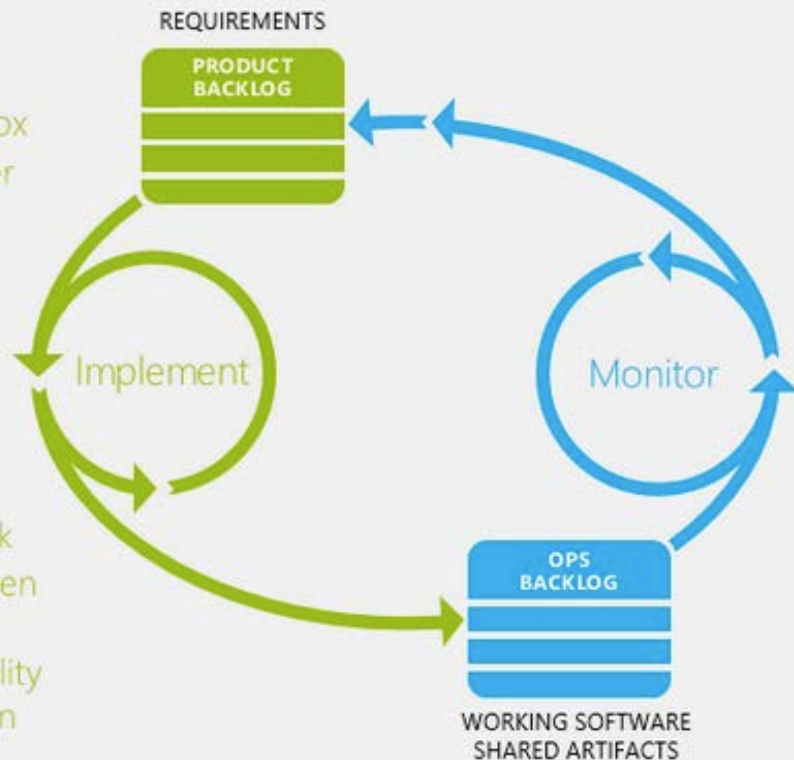
Define

- Requirements toolbox
- User and stakeholder engagement



Develop

- Continuous testing
- Continuous feedback
- Acceptance test driven development
- End-to-end traceability with PMO integration



Operate

- Continuously validate acceptance criteria
- Automated test lab management
- Integrating incident systems
- Actionable diagnostics

目录



自动化配置



故障与警告



趋势预测



性能优化

故障？

什么是故障？

如何定义故障？

故障事例

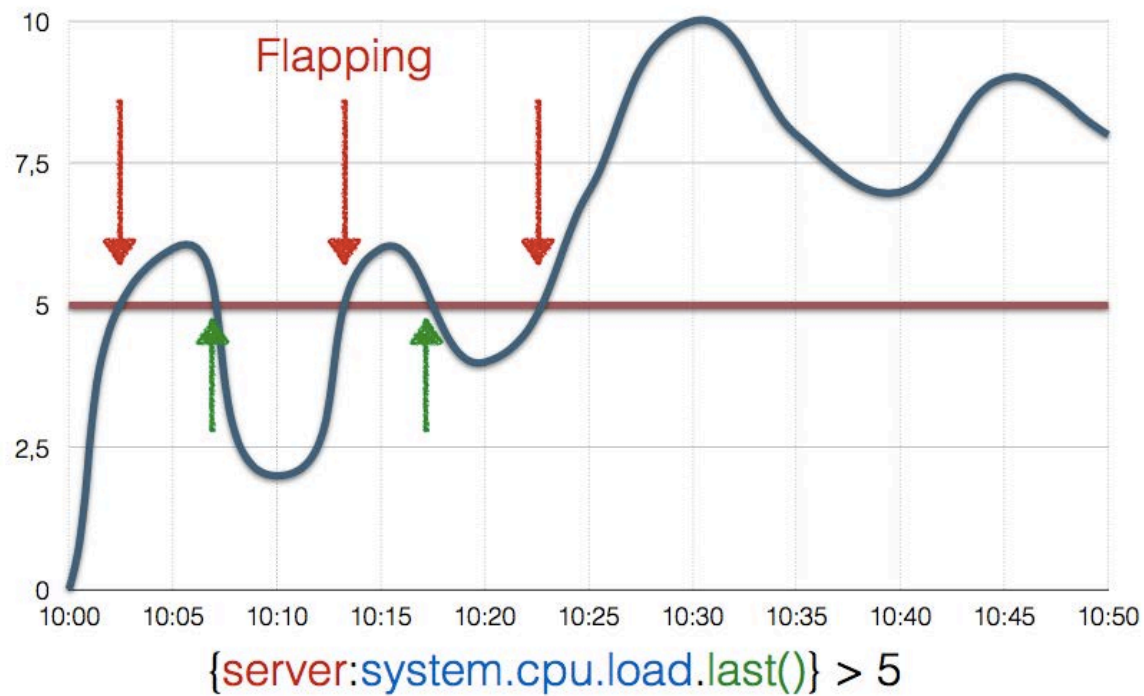
CPU利用率 > 90%

内存利用率 > 90%

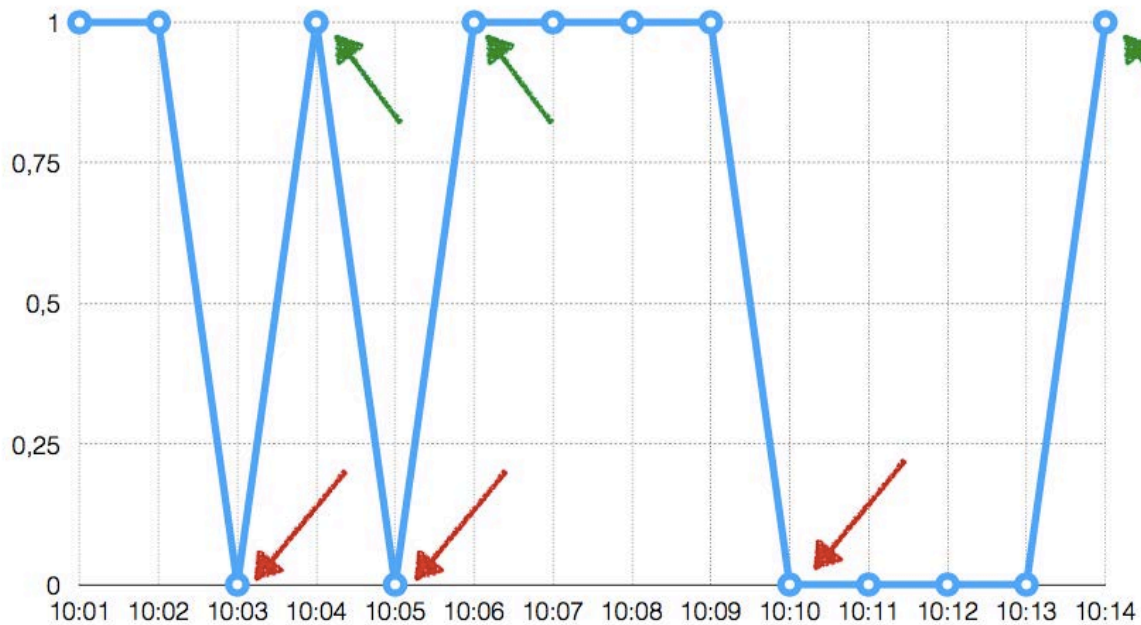
磁盘空间 > 90%

.....

故障事例

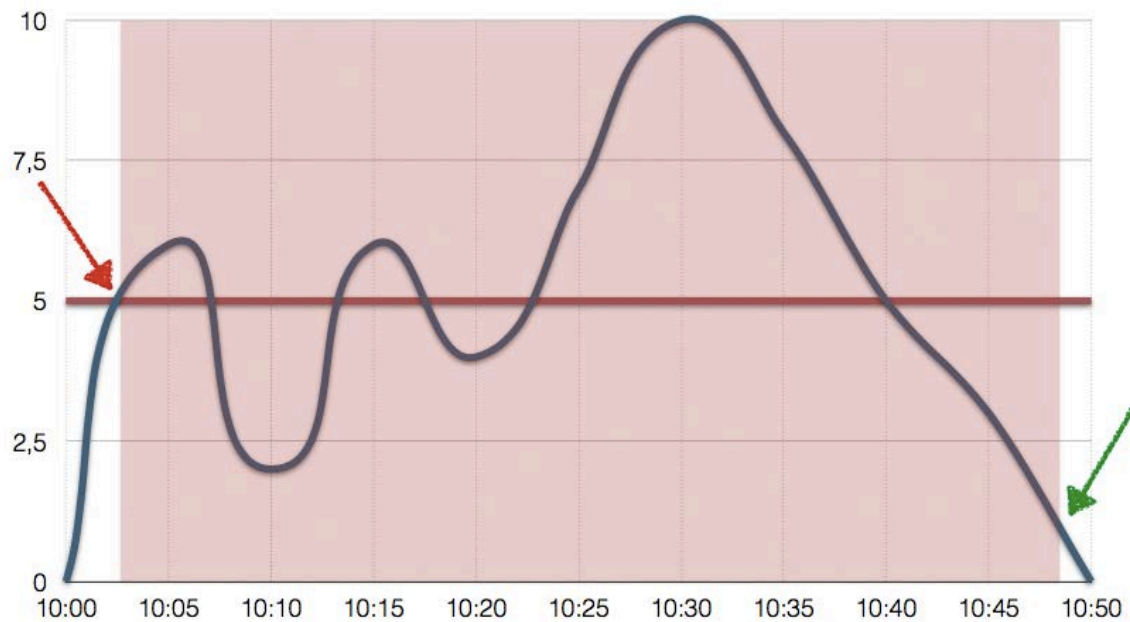


故障事例



`{server:net.tcp.service[http].last()} = 0`

故障事例



`{server:system.cpu.load.last()} > 5 ... {server:system.cpu.load.last()} > 1`

例子: {host:system.cpu.load.last()}>5

操作符号: - + / * < > = <> <= >= or and not

功能函数: min max avg last count date time diff regex等

故障分析和依赖:

{oneoaas1:system.cpu.load.last()} > 5 and

{oneoaas2:system.cpu.load.last()} > 5 and

{oneoaas:tps.last()} > 5000

目录



自动化配置



故障与警告



趋势预测



性能优化

凡事预则立,不预则废
基于现在,预测未来

磁盘还能用多久？
带宽还能撑多久？
用户数会到多少？

1. Trigger函数forecast() 在一段时间内，多久会达到设置的阈值

{Zabbix server:vfs.fs.size[/,free].forecast(7d,,7d)}<100M

2. Trigger函数timeleft() 达到阈值，需要多久的时间

{Zabbix server:vfs.fs.size[/,free].timeleft(7d,,104857600)}<1h

The screenshot displays the Zabbix web interface for configuring triggers. The top navigation bar shows 'All hosts / Zabbix server' with various monitoring tools (ZBX, SNMP, JMX, IPMI) and counts for Applications (14), Items (81), and Triggers (60). Below this, there are two panels for trigger configuration. The left panel is titled 'Trigger' and shows a trigger named '| free space < 100M' with the expression '{Zabbix server:vfs.fs.size[/,free].forecast(7d,,7d)}<100M'. An 'Add' button is visible next to the expression field. The right panel is titled 'Dependencies' and shows a trigger named '| free space is 100M < 6h base on one hour one day a' with the expression '{Zabbix server:vfs.fs.size[/,free].timeleft(1h,1d,104857600)}<6h'. A link for 'Expression constructor' is located at the bottom left of the left panel.

目录



自动化配置



故障与警告



趋势预测



性能优化

性能调优原则

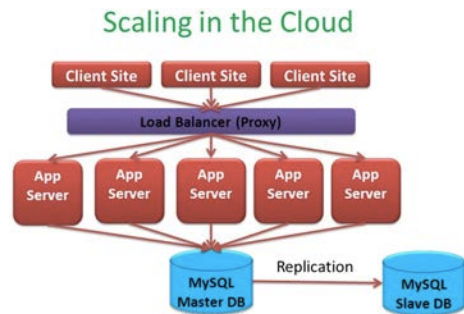
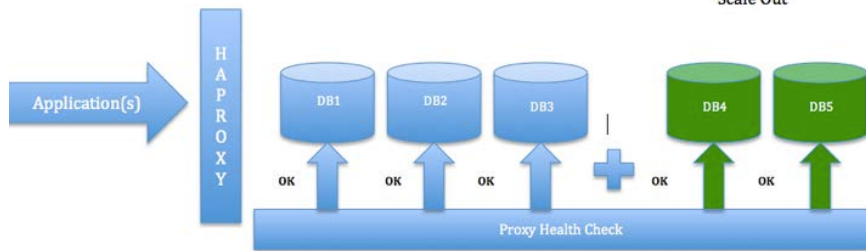
规划期间-考虑磁盘IOPS

架构方面-性能足够高，开销足够小

安装部署-保证各软件最优编译配置



监控方式调整
主动模式最优



可视化解决方案-Grafana



可视化解决方案-OneOaaS Monitor



关于OneOaaS

OneOaaS为用户提供**运维工具，解决方案**。包括**CMDB，监控系统，代码部署**等云时代的运维解决方案。

公司由一批资深运维专家组成,他们对云计算和自动化运维有着独特的见解,对技术有着狂热的追求,对各行各业业务有着透彻的理解,能够为用户提供切实有效的解决方案,并善于为客户解决运维难题。

其提倡用理论指导运维的方式,帮助客户建立运维意识,制定运维规范,使用成熟高效的运维工具去迎接大规模运维问题。公司自研的运维工具,能够有效解决运维中的**资产管理问题,配置管理问题,开发测试难题,代码管理问题,监控告警问题**。**真正的做到运筹帷幄之中,决胜千里之外。**

公司官方网站: <http://www.oneoaaS.com>



THANKS