

## Tsar介绍

Tsar是淘宝的一个用来收集服务器系统和应用信息的采集报告工具，如收集服务器的系统信息（cpu，mem等），以及应用数据（nginx、swift等），收集到的数据存储在服务器磁盘上，可以随时查询历史信息，也可以将数据发送到nagios报警。

Tsar能够比较方便的增加模块，只需要按照tsar的要求编写数据的采集函数和展现函数，就可以把自定义的模块加入到tsar中。

## 安装

Tsar目前托管在github上，[下载编译安装步骤](#)：

```
$git clone git://github.com/kongjian/tsar.git
$cd tsar
$make
$make install
```

安装后：

定时任务配置：/etc/cron.d/tsar，负责每分钟调用tsar执行采集任务；

日志文件轮转配置：/etc/logrotate.d/tsar，每个月会把tsar的本地存储进行轮转；

Tsar配置文件路径：/etc/tsar/tsar.conf，tsar的采集模块和输出的具体配置；

模块路径：/usr/local/tsar/modules，各个模块的动态库so文件；

## Tsar配置

Tsar刚安装完，还没有历史数据，[想要check是否正常，执行tsar -l，查看是否有实时信息输出](#)：

```
[kongjian@v132172.sqa.cm4 tsar]$ tsar -l -i 1
Time          ---cpu---  ---mem---  ---tcp---  ---traffic---  --xv  ---load-
Time          util    util    retran    pktin  pktout    ut
11/04/13-14:09:10  0.20    11.57    0.00      9.00    2.00    0.
11/04/13-14:09:11  0.20    11.57    0.00      4.00    2.00    0.
```

Tsar的配置主要都在 /etc/tsar/tsar.conf 中，常用的有：

- 增加一个模块,添加 mod\_<yourmodname> on 到配置文件中
- 打开或者关闭一个模块,修改 mod\_<yourmodname> on/off
- output\_stdio\_mod 能够配置执行tsar时的输出模块
- output\_file\_path 采集到的数据默认保存到的文件（如果修改的话需要对应修改轮转的配置 /etc/logrotate.d/tsar）
- output\_interface 指定tsar的数据输出目的，默认file保存本地，nagios/db输出到监控中心/数据库中，这两个功能还需要结合其它配置，具体见后面

## Tsar使用

- 查看历史数据，tsar
- -l/--list 查看可用的模块列表
- -l/--live 查看实时数据,tsar -l --cpu
- -i/--interval 指定间隔，历史,tsar -i 1 --cpu
- --modname 指定模块,tsar --cpu
- -s/--spec 指定字段,tsar --cpu -s sys,util
- -d/--date 指定日期,YYYYMMDD或者n代表n天前
- -C/--check 查看最后一次的采集数据
- -d/--detail 能够指定查看主要字段还是模块的所有字段
- -h/--help 帮助功能

## 高级功能

- 输出到nagios

配置：首先配置 output\_interface file,nagios，增加nagios输出

然后配置nagios服务器和端口，以及发送的间隔时间

```
####The IP address or the host running the NSCA daemon
server_addr nagios.server.com
####The port on which the daemon is running - default is 5667
server_port 8086
####The cycle of send alert to nagios
cycle_time 300
```

由于是nagios的被动监控模式，需要制定nsca的位置和配置文件位置

```
####nsca client program
send_nsca_cmd /usr/bin/send_nsca
send_nsca_conf /home/a/conf/amon/send_nsca.conf
```

接下来制定哪些模块和字段需要进行监控，一共四个阈值对应nagios中的不同报警级别

```
####tsar mod alert config file
####threshold servicename.key;w-min;w-max;c-min;cmax;
threshold cpu.util;50;60;70;80;
```

- 输出到mysql

配置： 首先配置 `output_interface` file,db , 增加db输出

然后配置哪些模块数据需要输出

```
output_db_mod mod_cpu,mod_mem,mod_traffic,mod_load,mod_tcp,mod_udp
```

然后配置sql语句发送的目的地址和端口

```
output_db_addr console2:56677
```

目的地址在该端口监听tcp数据，并且把数据入库即可，可以参照tsar2db: <https://github.com/kongjian/tsar2db>

## 模块开发

Tsar的一个比较好的功能是能够增加自己的采集，这时候需要编写模块代码，编译成so文件即可。

首先安装tsar-devel，刚才安装时，如果执行 `make tsar-devel`，就会把模块开发的基本文件安装到系统 然后执行tsar-devel，就能在当前模块生成一个模块目录：

```
[kongjian@v132172.sqa.cm4 tsar]$ tsar-devel test
build:make
install:make install
uninstall:make uninstall
[kongjian@v132172.sqa.cm4 tsar]$ ls test
Makefile  mod_test.c  mod_test.conf
```

按照要求修改mod\_test.c中的read\_test\_stats，set\_test\_record 完成后  
make;make install就完成新模块的配置文件和so的设置，执行tsar --test就能查看效果

另外也可以通过配置文件对自定义模块传递参数，方法是 修改配置文件中的 mod\_test on myparameter 然后在mod\_test.c中的read\_test\_stats函数中，通过parameter参数就可以获得刚才配置文件中的内容

## 其它

Taocode地址：<http://code.taobao.org/p/tsar/> 有其它问题请联系：[kongjian@taobao.com](mailto:kongjian@taobao.com)