Тонкая настройка производительности системы с помощью DTrace

Sergey Klyaus, Инженер
Sergey.Klyaus@Tune-IT.Ru

tuneit

Версия 1.1

Трассировка ядра ОС

- Сбор статистики, анализ производительности
- Отладка ядра
- Аудит системы

- Статически установленные счетчики
  - kstat (3KSTAT)
- Статическая трассировка ядра
  - TNF
- Отладочная печать

Не годится для production-систем!

tuneit

Трассировка ядра ОС

- Утилиты для анализа производительности: kstat → *stat

```
bash-3.00# iostat -xtc 5
                 extended device statistics                  tty
    cpu
device    r/s    w/s   kr/s   kw/s wait actv  svc_t  %w  %b  tin tout
us sy wt id
sd0       0.1    0.2    6.7    1.6  0.0  0.0   18.0   0   0    0   45
 0  0  0 100
sd2       0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0
ssd0      0.2    3.4    9.6  128.2  0.0  0.1   24.2   0   4
```

- iSCSI = BIO + iSCSI + TCP/IP + Ethernet

tuneit

DTrace

- Трассировка любой функции в ядре (FBT), любой инструкции в приложении (PID)
- Исполнение проб в RISC VM (DIF) — безопасность
- Только интересующие данные
  - Фильтрация данных с помощью предикатов
  - Агрегация статистических данных

tuneit

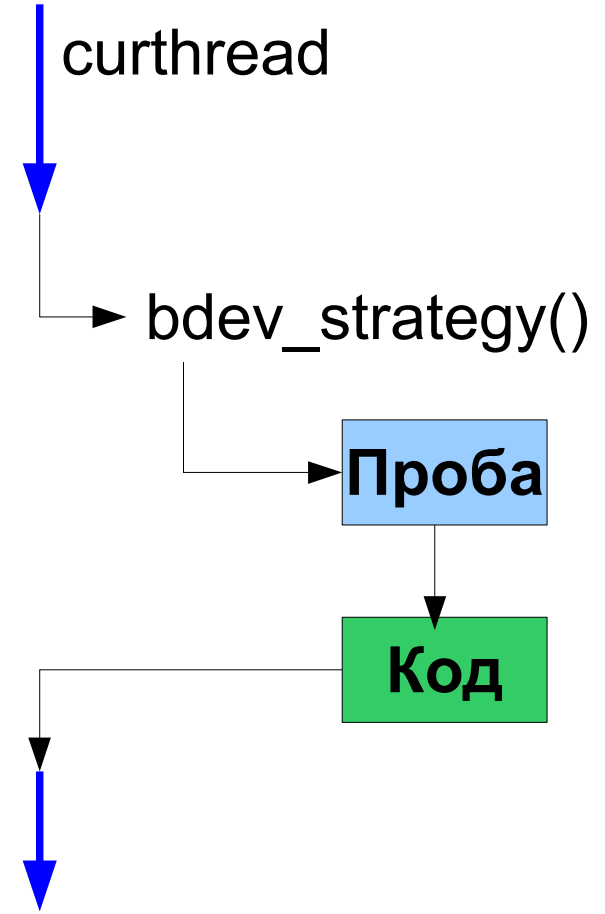
DTrace

- Подмена инструкции (FBT, PID)

```
bdev_strategy: pushq  %rbp
bdev_strategy+1:  movq   %rsp,%rbp
bdev_strategy+4:  subq   $0x10,%rsp
```

```
bdev_strategy:  int $0x3
bdev_strategy+1:  movq   %rsp,%rbp
bdev_strategy+4:  subq   $0x10,%rsp
```

- Статическая точка трассировки — **nop**



tuneit

DTrace

- Более 50000 проб ядра в Solaris 10

```
io:::start
```

← **проба**

```
/ args[0]->b_flags & B_WRITE &&
  args[1]->dev_statname == "ssd0" &&
  execname == "oracle" /
```

← **предикат**

```
{
   self->trace = 1;
}
```

← **действия**

tuneit

Провайдеры DTrace

- Function boundary tracing (fbt:::)

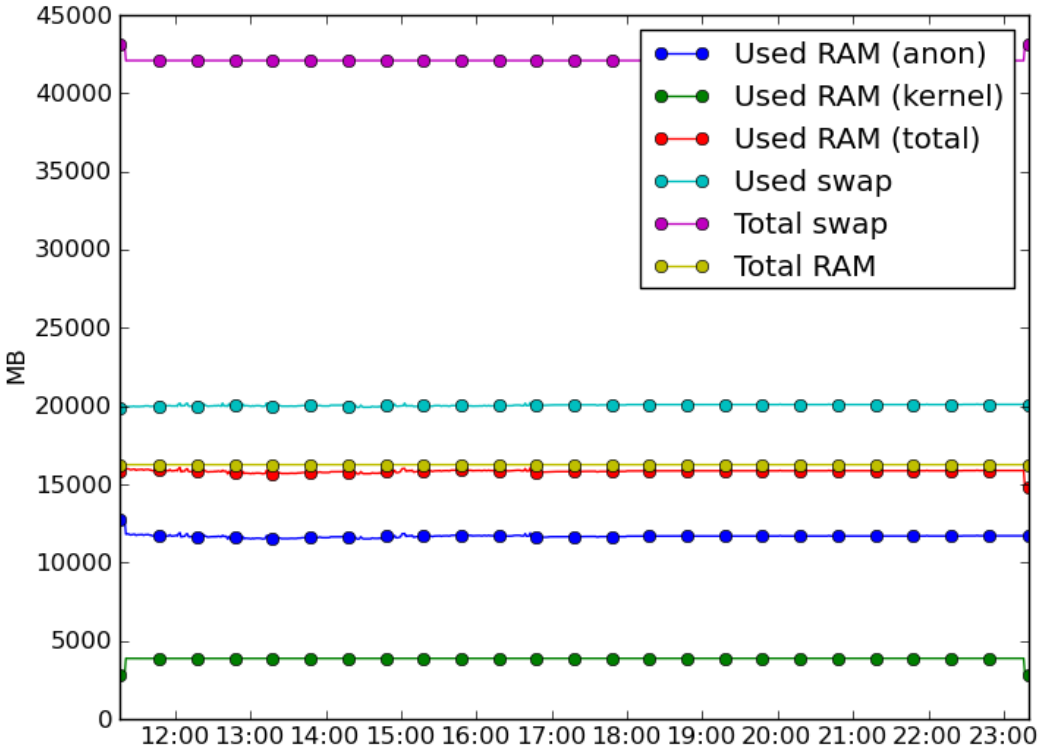
```
fbt:iscsi:iscsi_tx_scsi_data:entry
```

- Трассировка блокировок в ядре (lockstat:::)
- Сетевые провайдеры (tcp:::, ip:::, iscsi:::) - S11
- Планировщик процессов (sched:::)
- Счетчики производительности процессора (cpc:::) - S11
- Приложения: PHP, Python, Ruby, MySQL

tuneit

Чтение данных из ядра

```
tick-1s {
    trace(`freemem);
}
```

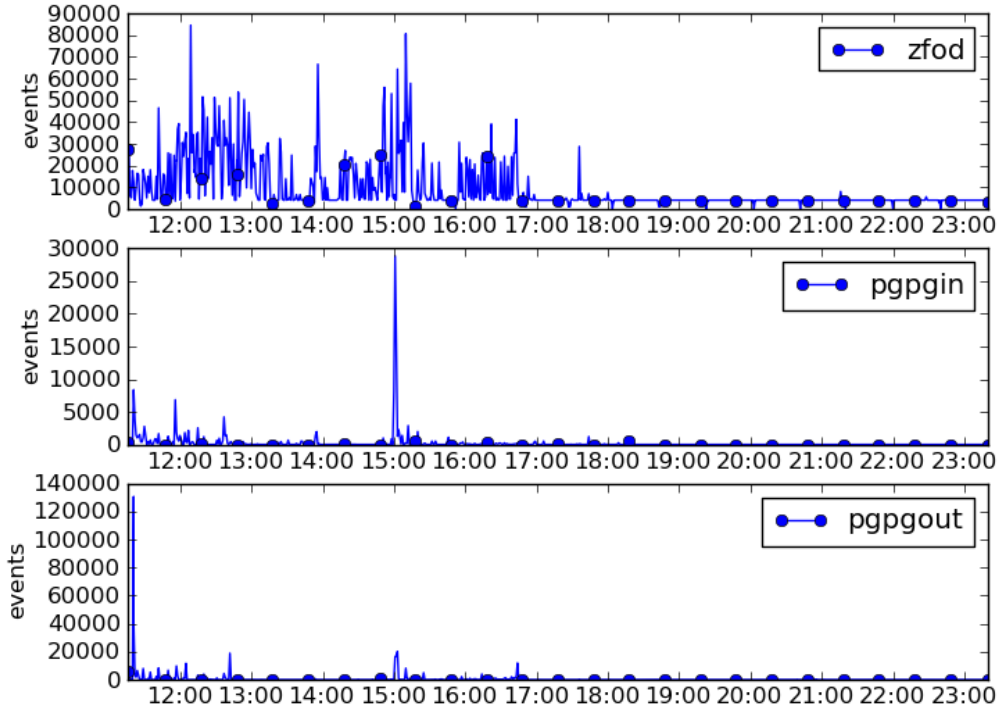


См. swapinfo.d

tuneit

Агрегация данных

```
vminfo::: {
    @[probename, execname] = sum(arg1);
}
```



tuneit

Агрегация данных

```
io:::start {
   @[execname] = quantize(args[0]->b_bcount);
}
```

Вывод:

```
 tar
           value  ------------- Distribution ---------count
            1024 |                                         0
            2048 |@                                        13
            4096 |@@                                       18
            8192 |@@@                                      30
           16384 |@@@@@@@@                                 87
           32768 |@@@                                      31
           65536 |@@@@@@@@@@@@@@@@@@@@@@@@@@              277
          131072 |@                                        8
          262144 |                                         0
```

tuneit

Время обработки запроса

```
syscall::read:entry
/execname == "bash"/
{
   self->start = timestamp;
}

syscall::read:return
/self->start/
{
   printf("%d ms", (timestamp - self->start) / 1000000);
   self->start = 0;
}
```

Вывод:

```
CPU     ID                    FUNCTION:NAME
  0  74315                      read:return 49 ms
```

tuneit

Складываем все вместе

iscsistat.d

```
BIO    OP     OPS/S      DATA     SVC_T ISCSI/BIO OPS
       RD        38    155648      4631 2
       WR        64    262144     14702 2
```

BIO

```
iSCSI OP       OPS/S           DATA           PDU
      TX RD        38         155648          1824
      RX RD        38         155648        157472
      TX WR        64         262144          3072
      RX WR        64         262144        265216

iSCSI OP       SVC_T
      WR           43
      RD           45
```

iSCSI

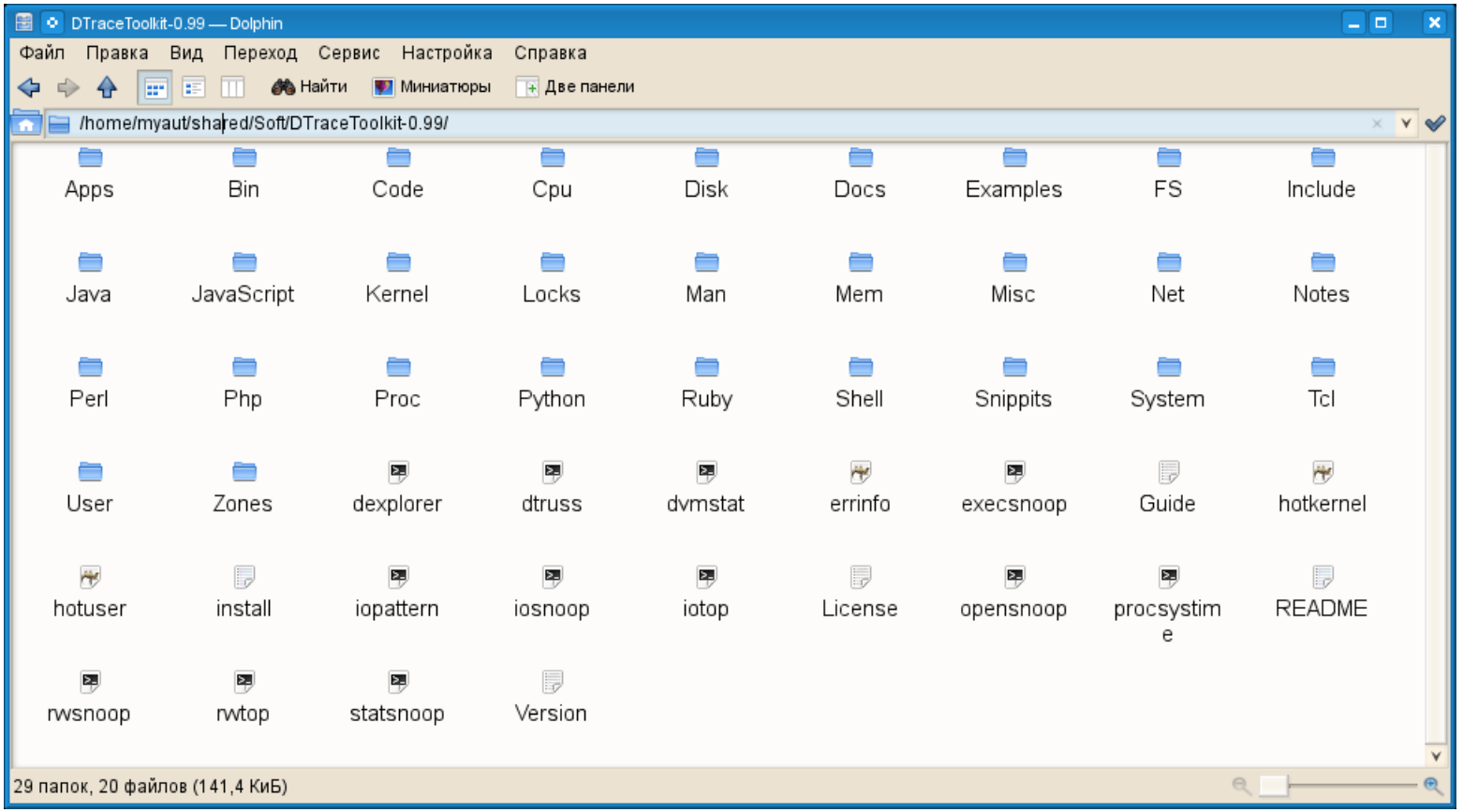
```
IDM    OP       OPS/S           DATA         SVC_T        vSVC_T
       send         102         267040            68            58
       recv         140         160544          6912            26
```

TCP

tuneit

DTraceToolkit

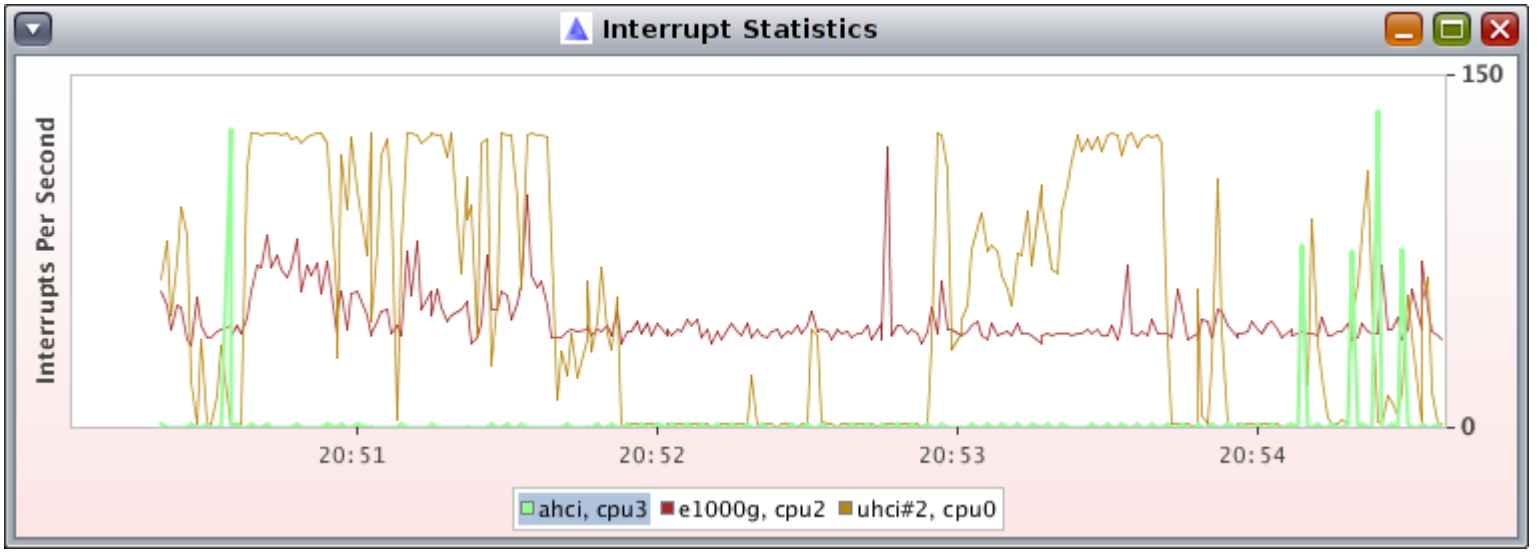
- 480 готовых скриптов



tuneit

DTrace GUI

- DTrace Chime



- NetBeans DTrace GUI Plugin

tuneit

DTrace

- Brian Gregg, Jim Mauro. DTrace: Dynamic Tracing in Oracle® Solaris, Mac OS X, and FreeBSD
- http://wikis.sun.com/display/DTrace/Documentation
- DTraceToolkit: http://hub.opensolaris.org/bin/view/Community+Group+dtrace/dtracetoolkit
- SiWiki: http://www.solarisinternals.com/
- Solaris 10 Operating System Internals (SI-365-S10)
- Dynamic Performance Tuning and Troubleshooting With DTrace (SA-327-S10)

tuneit

Спасибо за внимание!

http://www.tune-it.ru/web/myaut/home

tuneit