

lostat Mining



Data mining iostat logs.

Mark Kirkwood
Feb 09, 2010

Why?



- Performance checking database servers
 - IO is the common bottleneck
- Need to determine several metrics
 - Utilization
 - Wait
 - Read / write balance
- Usual monitor(s) don't provide these

Collecting



- Collecting data straightforward:

```
#!/bin/bash
#
DEFAULTOUTFILE=~/.trace/iostat-`date +%Y%m%d`.out
OUTFILE=${1:-$DEFAULTOUTFILE}
HOURS=${2:-23}
DELAY=${3:-1}
DEVICEFILTER="-d sda sdb sdc sdd sde sdf sdg sdh dhi sdj sdk"

REPEATS=$((($HOURS*3600/$DELAY))
iostat -xmt $DEVICEFILTER $DELAY $REPEATS >> $OUTFILE
```

The data



Linux 2.6.26-2-amd64 (unknown) 09/02/10 _x86_64_

Time: 00:31:01

Device:	rrqm/s	wrqm/s	r/s	w/s	rMB/s	wMB/s	avgrq-sz	avgqu-sz	await	svctm	%util
sda	2.74	40.47	7.21	10.25	0.35	0.20	65.01	0.10	5.69	4.35	7.60
sdb	2.73	40.55	7.14	10.45	0.34	0.22	65.30	0.02	32.04	4.02	7.06

Time: 00:31:02

Device:	rrqm/s	wrqm/s	r/s	w/s	rMB/s	wMB/s	avgrq-sz	avgqu-sz	await	svctm	%util
sda	0.00	83.84	0.00	26.26	0.00	0.44	34.46	0.22	8.31	6.00	15.76
sdb	0.00	83.84	0.00	28.28	0.00	0.45	32.86	0.23	11.86	5.43	15.35

Time: 00:31:03

Device:	rrqm/s	wrqm/s	r/s	w/s	rMB/s	wMB/s	avgrq-sz	avgqu-sz	await	svctm	%util
sda	0.00	197.00	0.00	11.00	0.00	0.82	152.00	0.19	17.09	5.82	6.40
sdb	0.00	197.00	0.00	11.00	0.00	0.82	152.00	0.18	16.73	5.82	6.40

Time: 00:31:04

Device:	rrqm/s	wrqm/s	r/s	w/s	rMB/s	wMB/s	avgrq-sz	avgqu-sz	await	svctm	%util
sda	0.00	34.00	0.00	37.00	0.00	0.28	15.35	2.46	56.32	3.57	13.20
sdb	0.00	34.00	0.00	39.00	0.00	0.29	15.18	2.30	55.69	3.38	13.20

What to mine?



- Want to know if IO is being “maxed out”
- What does this mean?
 - Utilization > some threshold (70 – 90% say)
 - Wait > some threshold (100 ms say)
- Also want to know:
 - When the above happens
 - How often it does

Mining



- Mining the data:
 - Find device usage lines that satisfy
 - $\%util > \text{threshold1}$
 - $\%await > \text{threshold2}$
 - Prepend timestamp to each such line
- Should be graphable

Mining



```
while ($line=<FH>) {  
    $lineno++;  
  
    # timestamp line  
    if ($line =~ m/^Time:/ || $line =~ m/^((\d){2}V){2}/) {  
        $timestamp = $line;  
        $timestamp =~ s/Time: //;  
        chomp($timestamp);  
    }  
    # cpu avg line(s)  
    elsif ($line =~ m/^avg-cpu:/) {  
        next;  
    }  
    # leading blank lines  
    elsif ($line =~ m/^\s+/) {  
        next;  
    }  
    # actual device line  
    elsif ($line =~ m/^[s|h|a|x](\w+)/) {  
        @fields = split(' ', $line);  
        $util = $fields[$#fields];  
        $wait = $fields[$#fields - 2];  
        if ($util > $utilthreshold &&  
            $wait > $waitthreshold) {  
            print "$timestamp $line";  
        }  
    }  
}
```

The output



07:03:13 sda	0.00	5705.00	3.00	220.00	0.01	29.92	274.87	141.93	578.48	4.48	100.00
07:03:13 sdb	0.00	5702.00	3.00	301.00	0.04	40.31	271.82	83.57	378.29	3.29	100.00
10:01:57 sdb	70.00	357.00	394.00	281.00	19.61	2.35	66.63	143.87	178.36	1.48	100.00
10:02:03 sdb	24.00	220.00	46.00	334.00	3.50	2.80	34.00	70.82	275.86	2.38	90.40
10:02:13 sda	0.00	554.00	13.00	357.00	0.10	3.02	17.28	136.88	238.32	2.69	99.60
10:02:13 sdb	0.00	555.00	16.00	370.00	0.13	3.16	17.47	139.80	290.33	2.55	98.40
10:02:23 sda	0.00	3241.00	1.00	285.00	0.00	8.17	58.55	145.26	274.60	3.41	97.60
10:02:23 sdb	0.00	3242.00	0.00	329.00	0.00	8.93	55.56	131.44	189.70	2.97	97.60
10:02:28 sda	0.00	11387.00	0.00	310.00	0.00	44.70	295.33	94.42	277.38	3.08	95.60
10:02:28 sdb	0.00	11387.00	0.00	315.00	0.00	44.29	287.92	90.14	267.95	3.02	95.20

Analysis



- Clearly shows spikes:
 - Write overload (red)
 - Read overload (blue)