Performance Tuning — The iostat Tool

Special Notices

Information in this document is correct to the best of our knowledge at the time of this writing. Please send feedback by fax to "AIXServ Information" at (512) 823-4009.

Please use this information with care. IBM will not be responsible for damages of any kind resulting from its use. The use of this information is the sole responsibility of the customer and depends on the customer's ability to evaluate and integrate this information into the customer's operational environment.

> **Note:** The information in this document has **not** been verified for AIX 4.1.

About this Document

This document is based on "Performance Tuning: A Continuing Series — The iostat Tool", by Barry Saad, from the January/February 1994 issue of *AIXtra: IBM's Magazine For AIX Professionals*.

Introduction

This article discusses iostat and how it can identify I/O-subsystem and CPU bottlenecks. iostat works by sampling the kernel's address space and extracting data from various counters that are updated every clock tick (1 clock tick = 10 milliseconds). The results — covering tty, CPU, and I/O subsystem activity — are reported as per second rates or as absolute values for the specified interval.

Syntax

Normally, iostat is issued with both an interval and a count specified, with the report sent to standard output or redirected to a file. The command syntax is given below:

```
iostat [-t] [-d] [PV(s)] [Interval [Count]]
```

The "-d" flag causes iostat to provide only disk statistics for all physical volumes. The "-t" flag causes iostat to provide only system-wide tty and CPU statistics.

Note: The "-t" and "-d" options are mutually exclusive.

If one or more physical volumes (PVs) are specified, the output is limited to those physical volumes. Multiple physical volumes can be specified, separated with blanks.

A time in seconds can be specified for the interval between records to be included in the reports. The initial record contains statistics since system boot. Succeeding records contain data for the preceding interval. If no interval is specified, a single record is generated.

If an interval has been specified, then the count of the number of records to be included in the report can also be specified. If an interval is specified without a count, iostat will continue running until it is killed.

```
tty: tin      tout    cpu: % user      % sys   % idle  % iowait
     2.2      3.3             0.4         1.3     97.7    0.6

 Disks:     % tm_act  Kbps    tps   msps    Kb_read   Kb_wrtn
 hdisk0     0.4       1.1     0.3           117675    1087266
 hdisk1     0.3       1.0     0.2           59230     1017734
 hdisk2     0.0       0.2     0.0           180189    46832
 cd0        0.0       0.0     0.0           0         0

tty: tin      tout    cpu: % user      % sys   % idle  % iowait
     2.2      3.3             0.4         1.3     97.7    0.6

 Disks:     % tm_act  Kbps    tps     msps    Kb_read Kb_wrtn
 hdisk0        0.4    1.1     0.3             117675  1087266
 hdisk1        0.3    1.0     0.2             59230   1017734
 hdisk2        0.0    0.2     0.0             180189  46832
 cd0           0.0    0.0     0.0             0       0
```

Figure 1. Sample Output from "iostat 2 2"

TTY Statistics in the iostat Output

The two columns of tty information (tin and tout) in the iostat output show the number of characters read and written by all tty devices. This includes both real and pseudo tty devices. Real tty devices are those connected to an asynchronous port. Some "pseudo tty devices" are shells, telnet sessions, and aixterms.

- tin shows the total characters per second read by all tty devices.
- tout indicates the total characters per second written to all tty devices.

Generally there are fewer input characters than output characters. For example, if you run the following:

```
iostat -t 1 30
cd /usr/sbin
ls -l
```

you will see few input characters and many output characters. On the other hand, applications such as vi result in a smaller difference between the number of input and output characters. Analysts using modems for asynchronous file transfer may notice the number of input characters exceeding the number of output characters. Naturally, this depends on whether the files are being sent or received relative to the measured system.

Since the processing of input and output characters consumes CPU resource, look for a correlation between increased tty activity and CPU utilization. If such a relationship exists, evaluate ways to improve the performance of the tty subsystem. Steps that could be taken include changing the application program, modifying tty port parameters during file transfer, or perhaps upgrading to a faster or more efficient asynchronous communications adapter.

CPU Statistics in the iostat Output

The CPU statistics columns (% user, % sys, % idle, and % iowait) provide a breakdown of CPU usage. This information is also reported in the output of the vmstat command in the columns labeled us, sy, id, and wa.

% user

The "% user" column shows the percentage of CPU resource spent in user mode. A UNIX process can execute in user or system mode. When in user mode, a process executes within its own code and does not require kernel resources.

% sys

The "% sys" column shows the percentage of CPU resource spent in system mode. This includes CPU resource consumed by kernel processes (kprocs) and others that need access to kernel resources. For example, reading and/or writing of a file requires kernel resources to open the file, seek a specific location, and read and/or write data. A UNIX process accesses kernel resources by issuing system calls.

Typically, the CPU is pacing (the system is CPU bound) if the sum of user and system time exceeds 90 percent of CPU resource on a single-user system or 80 percent on a multiuser system. This means that the CPU is the limiting factor in system performance.

The ratio of user to system mode is determined by workload and is more important when tuning an application than when evaluating performance.

A key factor when evaluating CPU performance is the size of the run queue (provided by the vmstat command). In general, as the run queue increases, users will notice degradation (an increase) in response time.

% idle

The "% idle" column shows the percentage of CPU time spent idle, or waiting, without pending local disk I/O. If there are no processes on the run queue, the system dispatches a special kernel process called wait. On most AIX systems, the wait process ID (PID) is 514.

% iowait

The "% iowait" column shows the percentage of time the CPU was idle with pending local disk I/O.

The iowait state is different from the idle state in that at least one process is waiting for local disk I/O requests to complete. Unless asynchronous I/O is being used by the process, an I/O request to disk causes the calling process to block (or sleep) until the request is completed. Once a process's I/O request completes, it is placed on the run queue.

In general, a high iowait percentage indicates the system has a memory shortage or an inefficient I/O subsystem configuration. Understanding the I/O bottleneck and improving the efficiency of the I/O subsystem require more data than iostat can provide; however, typical solutions might include:

- Limiting number of active logical volumes and file systems placed on a particular physical disk. (The idea is to balance file I/O evenly across all physical disk drives.)
- Spreading a logical volume across multiple physical disks. (This is useful when a number of different files are being accessed.)
- Creating multiple JFS logs for a volume group and assigning them to specific file systems. (This is beneficial for applications that create, delete, or modify a large number of files, particularly temporary files.)

  Backing up and restoring file systems to reduce fragmentation. (Fragmentation causes the drive to seek excessively and can be a large portion of overall response time.)
- Adding additional drives and rebalancing the existing I/O subsystem.

On systems running a primary application, high I/O wait percentage may be related to workload. In this case, there may be no way to overcome the problem. On systems with many processes, some will be running while others wait for I/O. In this case, the iowait can be small or zero because running processes "hide" wait time. Although iowait is low, a bottleneck may still limit application performance. To thoroughly understand the I/O subsystem, it is necessary to examine the statistics in the next section.

Disk Statistics in the iostat Output

The disk statistics portion of the iostat output provides a breakdown of I/O usage. This information is useful in determining whether a physical disk is limiting performance.

Disk I/O History

The system maintains a history of disk activity by default. Note that history is disabled if you see the message:

```
Disk history since boot not available.
```

This message displays only in the first output record from iostat.

Disk I/O history should be enabled since the CPU resource used in maintaining it is insignificant. History keeping can be disabled or enabled in SMIT under the following path:

```
-> System Environments
    -> Change/Show Characteristics of Operating System
        -> Continuously maintain DISK I/O history
            -> true | false
```

Choose "true" to enable history keeping or "false" to disable it.

Disks

The "Disks:" column shows the name(s) of the physical volume(s). They are either "hdisk" or "cd" following by a number. (hdisk0 and cd0 refer to the first physical disk drive and the first CD disk drive, respectively.)

% tm_act

The "% tm_act" column shows percentage of time the volume was active. This is the primary indicator of a bottleneck.

A drive is active during data transfer and command processing, such as seeking to a new location. The disk-use percentage is directly proportional to resource contention and inversely proportional to performance. As disk use increases, performance decreases and response time increases. In general, when a disk's use exceeds 70 percent, processes are waiting longer than necessary for I/O to complete because most UNIX processes block (or sleep) while waiting for their I/O requests to complete.

Kbps

Kbps shows the amount of data read from and written to the drive in KBs per second. This is the sum of Kb_read plus Kb_wrtn, divided by the seconds in the reporting interval.

tps

tps reports the number of transfers per second. A transfer is an I/O request at the device driver level.

Kb_read

Kb_read reports the total data (in kilobytes) read from the physical volume during the measured interval.

Kb_wrtn

Kb_wrtn shows the amount of data (in kilobytes) written to the physical volume during the measured interval.

Analysis of the Data

Taken alone, there is no unacceptable value for any of the above fields because statistics are too closely related to application characteristics, system configuration, and type of physical disk drives and adapters. Therefore, when evaluating data, one must look for patterns and relationships. The most common relationship is between disk utilization and data transfer rate.

To draw any valid conclusions from this data, one must understand the application's disk data access patterns — sequential, random, or a combination — and the type of physical disk drives and adapters on the system.

For example, if an application reads/writes sequentially, you should expect a high disk transfer rate when you have a high disk-busy rate. (Note: Kb_read and Kb_wrtn can confirm an understanding of an application's read/write behavior; however, they provide no information on the data access patterns).

Generally you do not need to be concerned about a high disk-busy rate as long as the disk-transfer rate is also high. However, if you get a high disk-busy rate and a low data-transfer rate, you may have a fragmented logical volume, file system, or individual file.

What is a high data-transfer rate? That depends on the disk drive and the effective data transfer rate for that drive. You should expect numbers between effective sequential and effective random disk-transfer rates. Below is a chart of effective transfer rates for several common SCSI-1 and SCSI-2 disk drives.

Table 1. Effective Transfer Rate (KB/sec), Part 1 of 2

Type of Access	**400 MB drive**	**670 MB drive**	**857 MB drive**
Read-Sequential	1589	1525	2142
Read-Random	241	172	262
Write-Sequential	1185	1108	1588
Write-Random	327	275	367

Table 2. Effective Transfer Rate (KB/sec), Part 2 of 2

Type of Access	1.2 GB drive	1.37 GB drive	1.2 GB S-2 drive	1.37 GB S-2 drive
Read-Sequential	2169	2667	2180	3123
Read-Random	292	299	385	288
Write-Sequential	1464	2189	2156	2357
Write-Random	362	491	405	549

The transfer rates were determined during performance testing and give more accurate expectations of disk performance than the media transfer rate which reflects the hardware capability and does not account for operating system and application overhead.

Another use of the data is to answer the question: "Do I need another SCSI adapter?" If you've ever been asked this question, you probably provided a generic answer or just plain guessed.

You can use data captured by iostat to accurately answer the above by tracking transfer rates, finding the maximum data transfer rate for each disk. Assume the maximum rate occurs simultaneously for all drives (worst case). For maximum aggregate performance, the measured transfer rates for drives attached to a given adapter must be below the effective SCSI adapter throughput rating.

For planning purposes, you should use 70 percent of the adapter's rated throughput (e.g., 2.8 MB per second for a SCSI-1 adapter). This should provide a sufficient buffer for occasional peak rates that may occur. When adding a drive, you must assume the data transfer rate. At least you will have the collected data and the effective transfer rates to use as a basis.

Keep in mind the SCSI adapter may be saturated if the data transfer rates over multiple intervals approach the effective SCSI adapter throughput rating. In that case, the above analysis is invalid.

Conclusions

The primary purpose of the iostat tool is to detect I/O bottlenecks by monitoring the disk utilization (%tm_act field). iostat can also be used to identify CPU problems, assist in capacity planning, and provide insight into solving I/O problems. Armed with both vmstat and iostat, you can capture the data required to identify performance problems related to CPU, memory, and I/O subsystem(s).

Reader's Comments

Please fax this form to (512) 823-4009, attention "AIXServ Information". You may also e-mail comments to: elizabet@austin.ibm.com. These comments should include the same customer information requested below.

Use this form to tell us what you think about this document. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

If you need technical assistance, contact your local branch office, point of sale, or 1-800-CALL-AIX (for information about support offerings). These services may be billable. Faxes on a variety of subjects may be ordered free of charge from 1-800-IBM-4FAX. Outside the U.S. call 415-855-4329 using a fax machine phone.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Note: If you have a problem report or item number, supplying that number may help us determine why a procedure did or did not work in your specific situation.

Problem Report or Item #: Branch Office or Customer #:

Be sure to print your name and fax number below if you would like a reply:

Name: Fax Number:

01/24/95

END OF DOCUMENT (iostat.perf.tune.32.lxp)