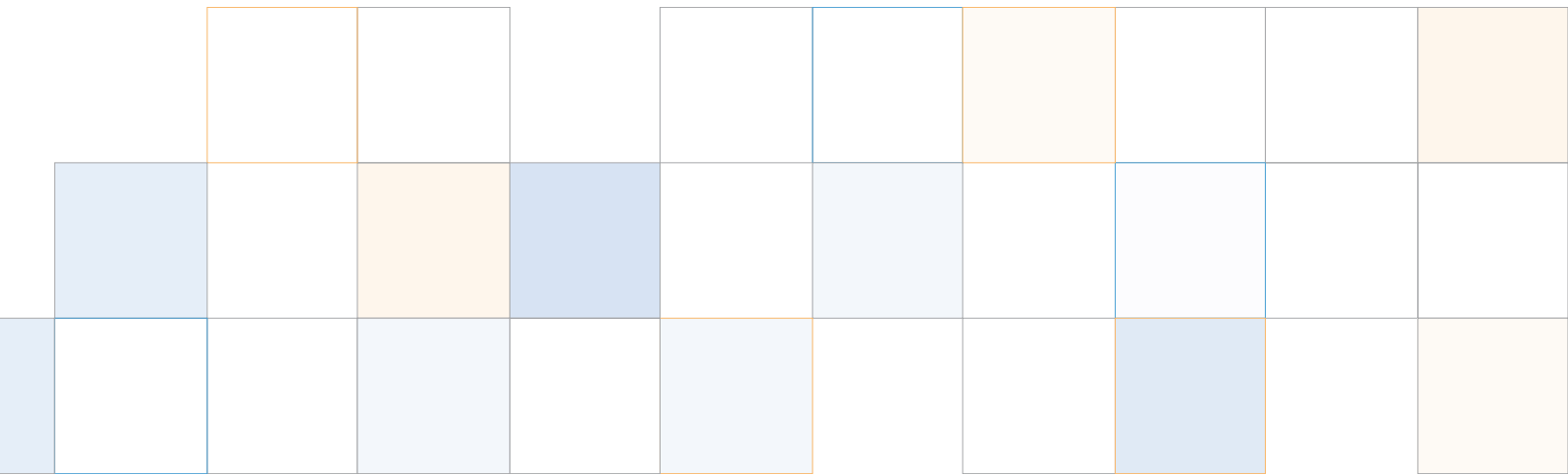


tech tips



Tuning the TSM Client for Performance

Tuning the performance of your TSM clients can free up a large amount of time, increasing the time for daily house-keeping tasks such as space reclamation. Tuning the performance of your TSM clients can be done by changing a lot of the variables in the client's 'dsm.opt' file.

This '**dsm.opt**' file can be found in different locations, dependant on your operating system.

AIX: `/usr/tivoli/tsm/client/ba/bin`

Windows: `C:\Program Files\Tivoli\TSM\baclient`

If you're unable to find it within those directories, you can simply use the search feature of Windows, or the find command on UNIX, eg:

```
find / -name "dsm.opt"
```

Let's have a look at some of the variables we can tune:

COMMRESTARTDURATION (minutes) – The maximum number of minutes you want the client to attempt to reconnect with the server after communication failure occurs. **Default is 60.**

COMMRESTARTDURATION **30**

COMMRESTARTINTERVAL (seconds) – The number of seconds you want the client to wait before attempting to re-connect with the server after a communication failure. **Default is 15.**

COMMRESTARTINTERVAL **45**

RESOURCEUTILIZATION – This determines the number of concurrent sessions that the client and server can use. There are two types of sessions, producer and consumer. The difference between producer and consumer is illustrated with this diagram:

Producer thread writes data to a buffer (determines what files to be backed up, builds transactions)

Consumer thread reads from a buffer (handles file I/O, compression, encryption)

You can increase this **RESOURCEUTILIZATION** variable from 0 to 10.

RESOURCEUTILIZATION **5**

Determining the best value for this option will require a bit of trialing. Factors such as network bandwidth, computer processing power and memory resources can all play a role in how effective a change in **RESOURCEUTILIZATION** makes.

When using this option, you may need to change the maximum number of mount points on the node from the TSM server using:

```
tsm: server1> update node [node_name] maxnummp=[number]
```

TCPBUFFSIZE – This relates to the size of the TCP/IP communication buffer. The higher the value, the more memory used on the client, but can improve performance across the link. This communication refers to that between the TSM client(s) and TSM server(s). **The default is 32 KB, and maximum of 512 KB.**

```
TCPBUFFSIZE 32
```

TCPWINDOWSIZE – The TCPWINDOWSIZE option allows you to specify the TCP window in kilobytes, overriding the operating system's default. If you set this value to '0', the operating system default will be used; however, this is not recommended. **The recommended value is '63'**. Like TCPBUFFSIZE, a greater value here uses additional memory resources, but can improve communication performance.

```
TCPWINDOWSIZE 63
```

TXNBYTELIMIT – This option allows you to specify the number of kilobytes that the client buffers prior to being sent to the server as a transaction. A transaction is effectively an object of data, so rather than sending a lot of small files across the link (increasing overhead), they're grouped into transaction groups.

The transaction byte limit (TXNByteLimit) values range from 300 to 2097152 (2GB). **The default is 25600 (25MB).**

```
TXNBYTELIMIT 25600
```

These are just some of the options you can tune on the client-side, for further information regarding client-side tuning and also server-side, please visit the following link:

<http://publibfp.boulder.ibm.com/epubs/pdf/c3201411.pdf>

By Michael Andrews