NAME

rrd-fs – description of the NFS monitoring graphs

DESCRIPTION

The *nfs.xml* file in the **rrdmon** package describes a couple of data channels retrieved from the operating system that help to assess the use of the network file system.

The network file system protocol specifies a large number of commands the client can use to access information about files and directories and their contents. The four graphs of the NFS monitoring RRD group these operations logically and in such a way that they appear with roughly equal frequncy so that they can in fact be reasonably displayed in the same graph.

NFS attributes

Most NFS operations are attribute requests, which retrieve information about a file or directory from a server.

getattr Retrieve file or directory attributes like the type, the access mode, the number of links, the user and group id, the size and the access time stamps. This is essentiall the information casual users display using the **ls**(1) command.

setattr Set the attributes

lookup The lookup function returns a file handle, an opaque identifier for a file or directory. All other functions on files or directories reference a file handle, so any NFS transaction needs to first retrieve the file handle. This makes the *lookup* operation by far the most frequent operation.

access The access operation queries the server the type of operations the user has access to. The client can specify the a bit mask of operation types it wishes the server to check, the server will return the access types the user has. The operation types are *read*, *lookup* (finding a name in a directory) *modify*, *extend* (add data or add directory entries), *delete*, *execute*.

readlink
Read the data in a symbolic link

NFS file operations

By file operation we mean operations that access or change the contents of files. Directory operations are displayed in the *NFS directory operations* graph explained below.

read Read a block of data from a file.

write Write a block of data to a file.

create Create a file.

remove Remove a file identified by name.

rename Rename a file.

symlink
: Create a symbolic link

mknod Create a device node. Device nodes are interpreted on the client file system, not the server, the data read from them does not come from the server, just as data read from a device node on a disk does not come from the disk but is read by the kernel from the device associated with the device node.

NFS directory operations

Directory operations are used to create and destroy directories and to read the contents of directories. The rather infrequently used *link* operation is in this group because it is similarly rare and has better chances to be seen in these graphs.

mkdir Create a new directory.

rmdir Remove a directory

link Create a symbolic link.

readdir Read the contents of a directory.

readdirplus
: This is an extended file operation introduced by NFSv3 which retrieves additional information about a directory entry. The most important is the file handle, which saves the subsequent *lookup* operation that would be required if anything needed to be done to a file.

NFS file system operations

Operations to retrieve file system information. This information can be characterized as roughly what the unix **df**(1) command displays. By their natures, these operations are very rare and mostly used for system administration and rather seldom in normal application workloads.

fsstat Various file system statistics like the total, allocated and free number of files (familiar as the number of *i*-nodes to long time Unix users, even if modern file systems no longer use this concept) and bytes.

fsinfo Static file system information like block sizes and maximum file sizes. or the time granularity on the server. If a server has better time granularity as the client, files that are different age on the server may appear as the same age on the client.

pathconf
: Information about path names, like the maximum number of components or whether file names are case sensitive on the server. On Windows or Mac OS X servers, they aren't. If the server cannot distinguished case in file names, the client cannot create a file named *Makefile* if there already is file named *makefile* present.

commit Commit the data written to a file to disk. This is strictly speaking a file operation, but it is used very rarely so it would be impossible to see it.

SEE ALSO

rrdsetup(8), **rrdupdate**(8)

AUTHOR

Andreas Mueller, andreas.mueller@othello.ch