

# Table of Contents

<b>Throttles for Xrootd</b> .....	<b>1</b>
Configuration.....	1
Throttle Example.....	2

# Throttles for Xrootd

We have contributed throttles for Xrootd, a capability commonly requested by sysadmins. This page covers installation and configuration of the throttles, complementing the upstream document at [http://xrootd.org/doc/dev48/xrd\\_config.htm](http://xrootd.org/doc/dev48/xrd_config.htm).

Currently, the throttle code comes as part of the Xrootd install; no additional RPMs are needed.

## Configuration

To enable the Xrootd plugin, you must request Xrootd to explicitly load it as a filesystem plugin:

```
xrootd.fslib throttle default
```

The following directives control the behavior of the throttle plugin.

---

```
throttle.throttle [data dlimit] [concurrency climit] [iops ilimit]
```

Each option in this directive controls a different aspect of the throttling:

- **dlimit**: The maximum number of bytes per second the server may read and write to clients. You may use the suffixes "k", "m", or "g" to indicate values in kilobytes, megabytes, or gigabytes, respectively.
- **ilimit**: The maximum number of read or write operations per second. A read() call is counted as one read; a readv() call with 50 chunks is counted as 50 reads.
- **climit**: The maximum number of IO operations in-progress at any given time.
- A read or write counts equally toward the limit; for example, if **dlimit** is set to 10M clients may perform 5MB/s of reads and 5MB/s of writes, but not 10MB/s of reads and 10MB/s of writes.
- For the **dlimit** and **ilimit** limits, a fairshare is applied. Clients mapped to the same username utilize the same fairshare. If Bob runs 50 clients and Alice runs 1 client, Bob and Alice should still get the same amount of aggregate bandwidth.
- Typically, there is no good way to determine the correct value of **ilimit**; most sites will really care about the **climit**, as that is a more direct measure of filesystem load. When the filesystem starts slowing down due to overload, this will eventually increase up to the number of connected clients.
  - ◆ This is roughly analogous to the Linux "load" metric as reported by "uptime".

Example usage:

```
throttle.throttle data 25M concurrency 10
```

---

```
throttle.trace [loadshed] [ioload] [bandwidth] [debug]
```

This directive controls the logging behavior of the throttle. In general, each option prints out a status report per second. When combined with **debug**, pertinent debugging information is printed out once per IO request.

Notes:

- **loadshed**: Note when a client is loadshed.
- **ioload**: Print out load summaries once a second.
- **bandwidth**: Print out bandwidth usage summaries once a second.
- **iops**: Print out information about the IO operations per second.
- **debug**: Print out sufficient debugging information to determine how throttle calculations are done for each request.

Some sites find this useful while debugging / validating the throttle code but overly verbose for day-to-day operations.

## Throttle Example

This is the configuration was used at Nebraska for testing the throttles

```
xrootd.fslib throttle default
throttle.throttle concurrency 10
throttle.trace loadshed ioload
```

Typically, a site will only want to implement a concurrency limit. Bandwidth limits are usually only useful if you want to limit Xrootd under 1Gbps.

---

This topic: [Main > XrootdThrottlingExperimental](#)  
Topic revision: r4 - 2019-02-06 - [BrianBockelman](#)



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? [Send feedback](#)