# File Transfer Throttles Design Document

Goal: Add support for policies that explicitly limit the bandwidth and/or I/O load of file transfers.

Scope: In this proposal, we will limit ourselves to throttling aggregate disk load, but we will structure the code to support throttles of other metrics in the future.

## What Exists Today

The existing scheduler allows up to *U* upload and *D* download file transfer processes to be active. It chooses which queue to service in starvation order, in an attempt to achieve equal sharing. By default, jobs are assigned to queues that are unique to the owner’s name, so the equal sharing is between users; other configurations are supported. Once a transfer process is activated, it is expected to run to completion unless the scheduler aborts it due to running longer than the maximum allowed time.

If one wishes to restrict file transfers to a certain maximum bandwidth or I/O load, it is not clear what fixed values of *U* and *D* would achieve the desired result, if any. For example, if one wishes to limit the number of processes writing to the disk, *D* could be set to the desired limit, but in periods when the active transfers happen to be spending more time waiting for network input than writing to the disk, this configuration may result in more conservative throttling than actually desired.

## Proposed Policy Form and Scheduling Algorithm

We propose to modify the existing scheduler, rather than to drastically revamp it. The proposed modification is for the scheduler to use a high/low water mark for deciding whether to allow new transfers to start:

* **above** the high water mark: the scheduler will not start new transfers.
* **between** the high and low water mark: the scheduler will only start new transfers to replace ones that finish.
* **below** the low water mark, the scheduler will start new transfers up to the limits *U* and *D*. The rate at which the number of active transfers is allowed to increase will be limited so the effect has time to be measured before making further increases.

The administrator provides the high/low water mark for aggregate disk load, which is the average number of processes blocked in reads plus writes to the disk. If no low water mark is specified, the default is 10% below the high level.

Example configurations:

* FILE\_TRANSFER\_DISK\_LOAD\_THROTTLE = 3
* FILE\_TRANSFER\_DISK\_LOAD\_THROTTLE = 2 to 3

The time scales of the load average metric that are used for throttling will be 1m for the high water mark and 5m and less for the low water mark. (Should this be configurable? For now, I say no.)

When throttling is configured, the number of allowed active transfers may increase by at most 1 per minute when the I/O load update interval is the default 10s. If the update interval is configured differently (not expected), then the rate will be decreased if necessary to not exceed 1/3 the update frequency. This ensures adequate time to observe the effect of increased activity. When resuming after a lull in demand, transfers may start without delay up to the pre-lull level of allowed active transfers; the decision about the next increment must then wait according to the usual rate limit.

How should uploads vs. downloads be prioritized when throttling is in play? Currently, uploads and downloads are independently scheduled. Throttling of net disk load could result in starvation of one due to activity of the other. To avoid chronic starvation, we will continue to place uploads and downloads in separate queues that each have equal weight from the scheduler’s point of view.

## Monitoring

The effectiveness of the scheduling algorithm in keeping usage below the high water mark will be monitored. The following values will be published:

* FileTransferDiskThrottleHigh: The configured high water mark.
* FileTransferDiskThrottleLow: The configured low water mark.
* FileTransferDiskThrottleExcess\_<horizon>: The exponential moving average of the aggregate disk load that was above the throttle’s high water mark. The time horizons will be the same as for other I/O stats: default 1m, 5m, 1h, 1d.
* FileTransferDiskThrottleShortfall\_<horizon>: The exponential moving average of the aggregate disk load that is below the throttle’s high water mark when there are queued transfers.

In addition to the above, we have identified addition transfer metrics that we would like to add. They are not directly related to the throttling policy, but they may be useful for understanding what is going on and eventually for throttling at the matchmaking level. These attributes report bytes queued to be sent plus bytes not yet sent in transfers that are active.

* FileTransferUploadBytesWaiting
* FileTransferDownloadBytesWaiting
* <queue>\_FileTransferUploadBytesWaiting
* <queue>\_FileTransferDownloadBytesWaiting

## Effort Estimate

[ 3 days ] Implement, test, and document disk load throttle.

[ 1 day ] Compute, publish, and document information about excess disk load.

[ 2 days ] Collect, publish, and document information about bytes waiting to be sent.