Block I/O bandwidth Control
ブロックI/O帯域制御

高橋浩和 <taka@valinux.co.jp>
稲越宏弥 <hiroya@fujitsu.com>

2008年 11月 20-21日

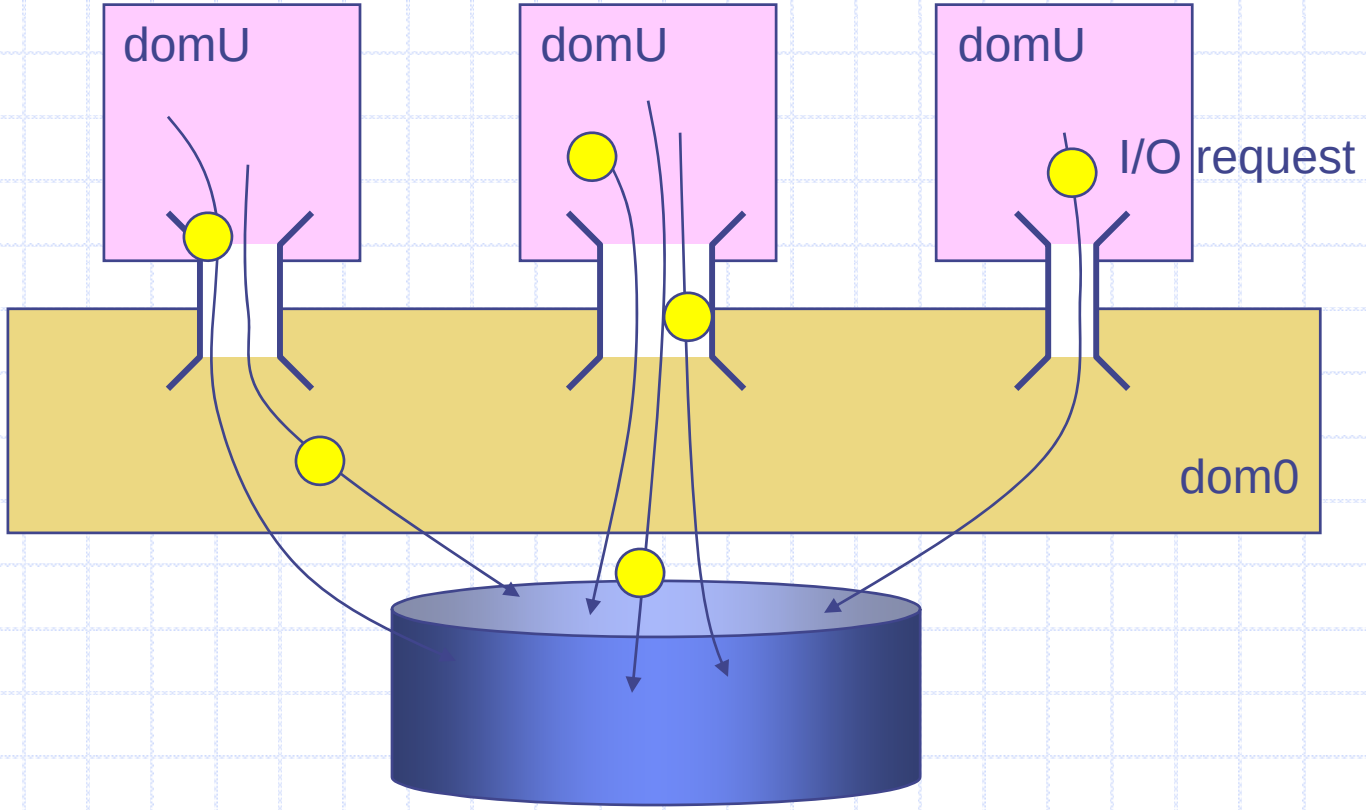
本研究・開発の一部は，経済産業省の委託を受けた
技術研究組合　超先端電子技術開発機構(ASET) の
セキュア・プラットフォームプロジェクトの成果です．

目的

- 複数の仮想マシン間で、同じ物理デバイスを、共有したい。他の仮想マシンの影響を受けないように、帯域を予約したい。

やったこと

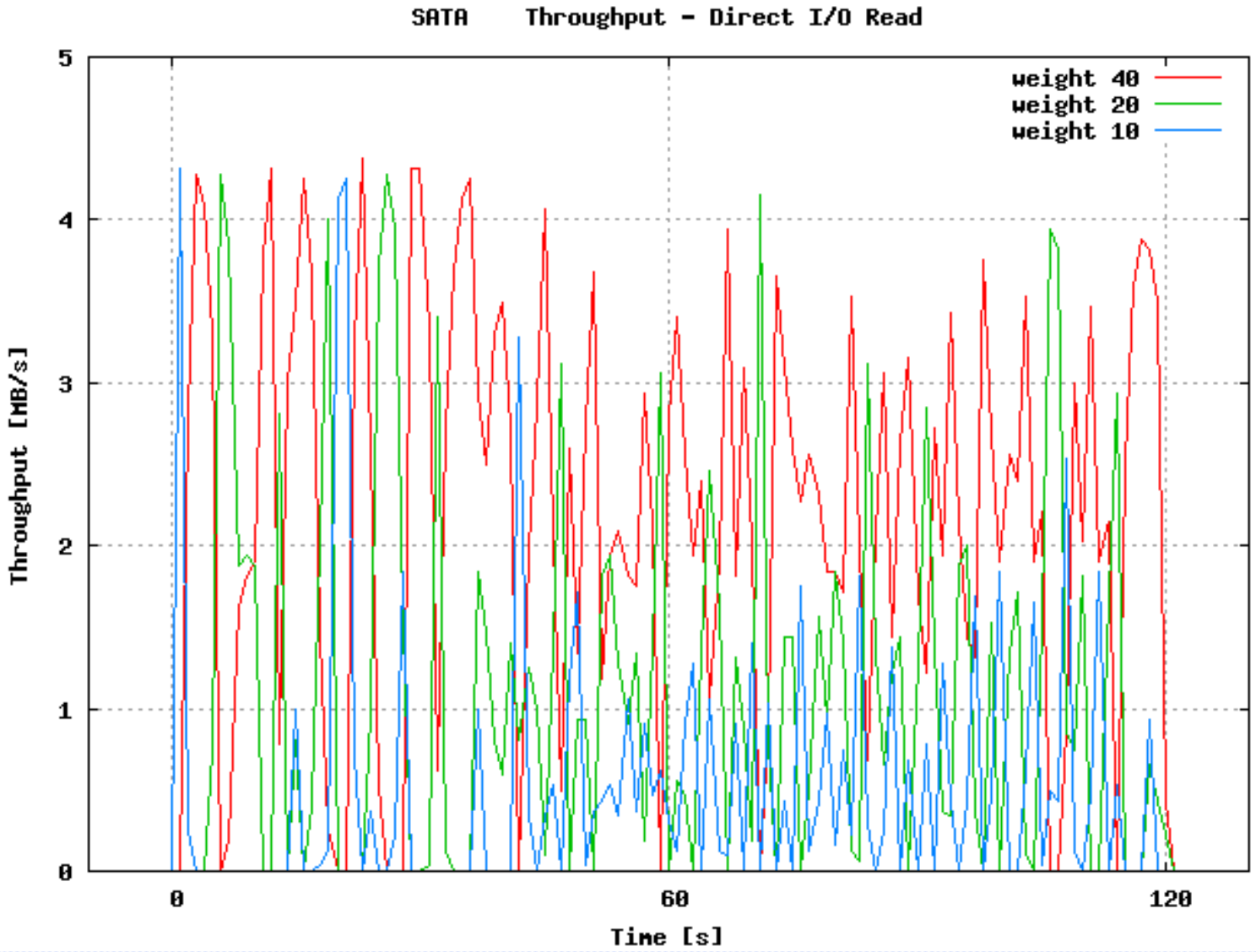
- dom0に、I/O流量を制御する機能を実装した



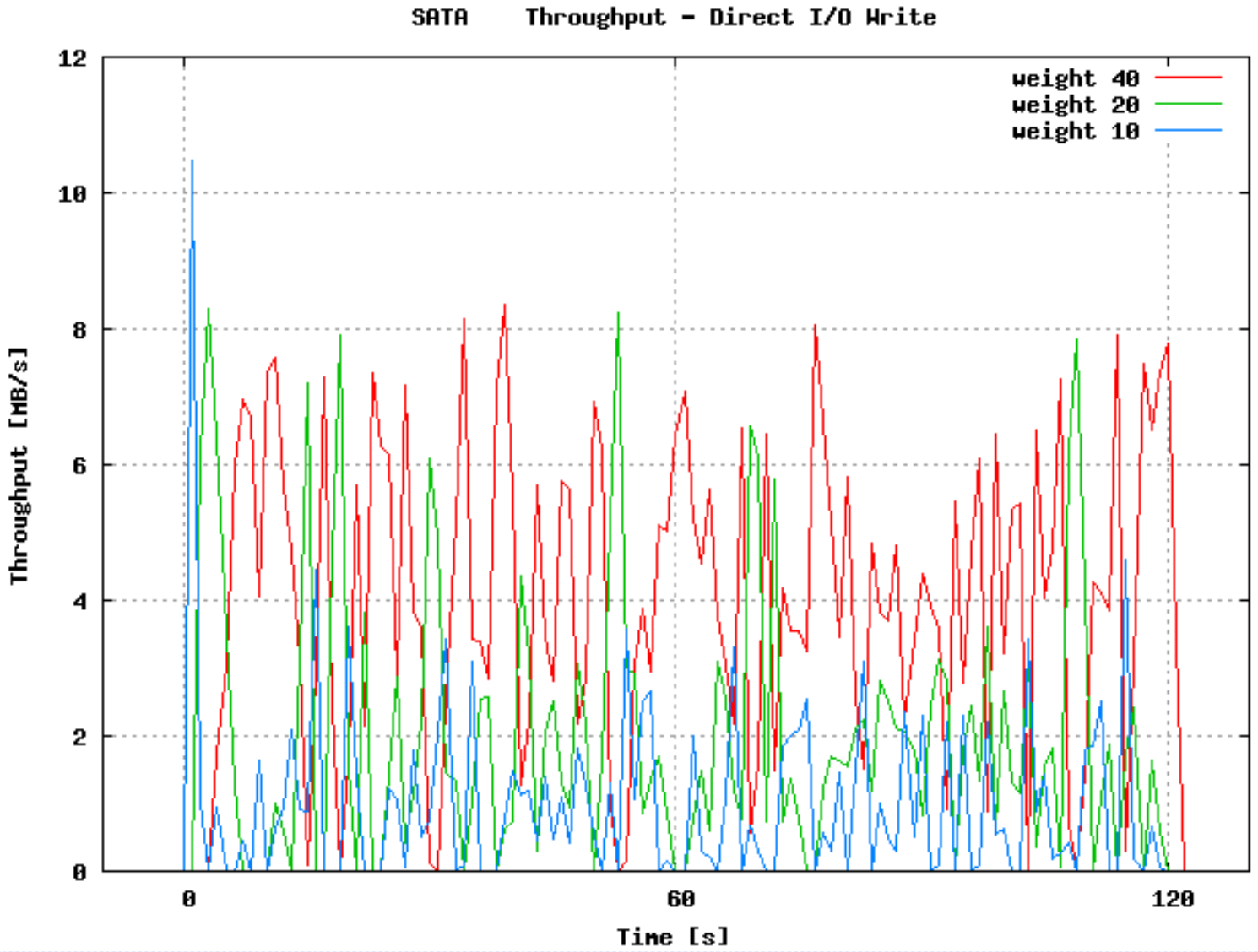
評価結果

- weight 40, 20, 10 のグループを作って、さまざまな種類のストレージ上で負荷をかけた
  - SATA
    - 普通のSATA。I/Oスケジューラはcfq
  - High-end Storage
    - ある開発中のハイエンドストレージ(SCSI over FC)。I/Oスケジューラは noop
  - SSD
    - ある開発中の高性能SSD。I/Oスケジューラはnoop

SATA Read (Direct I/O)

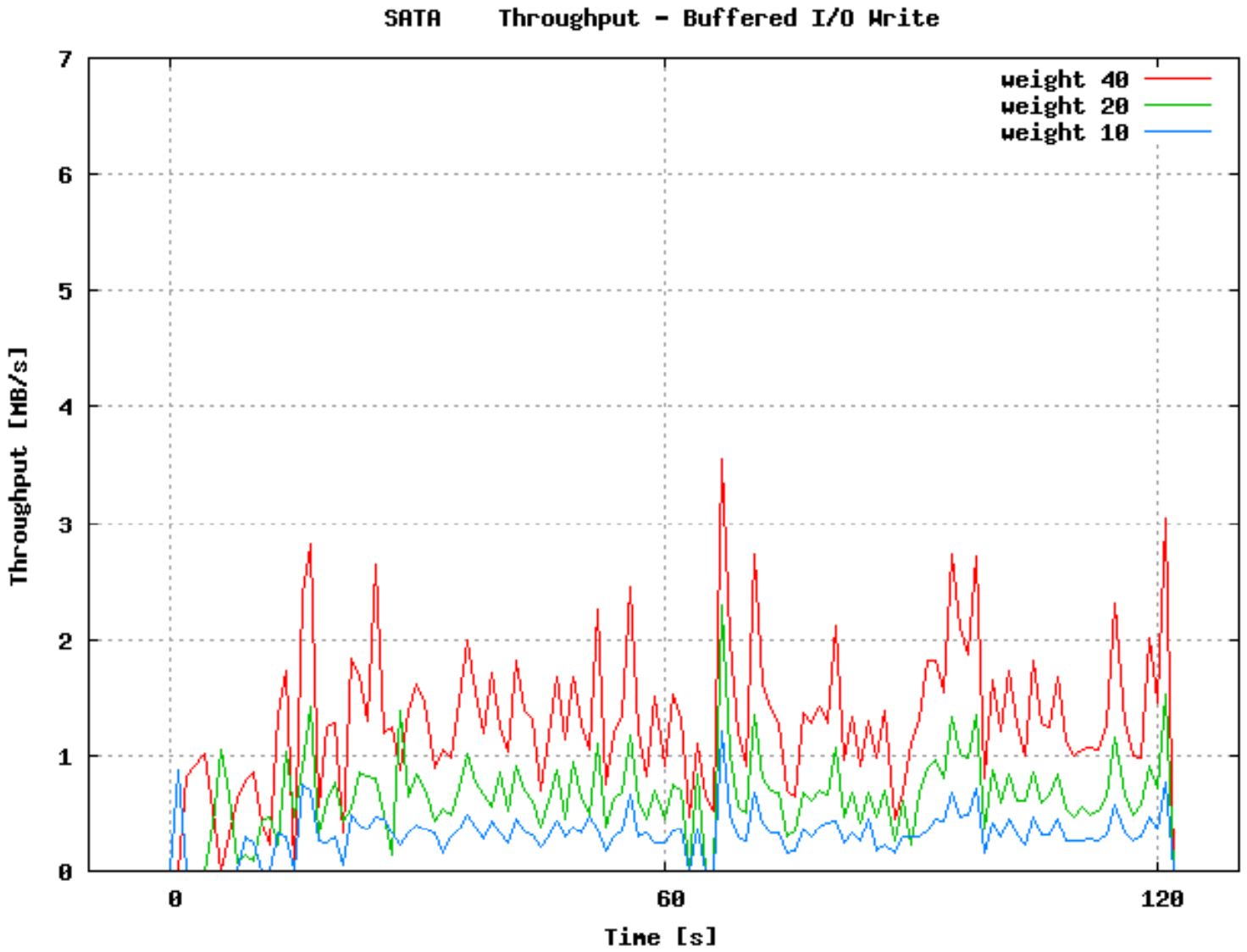


SATA Write (Direct I/O)

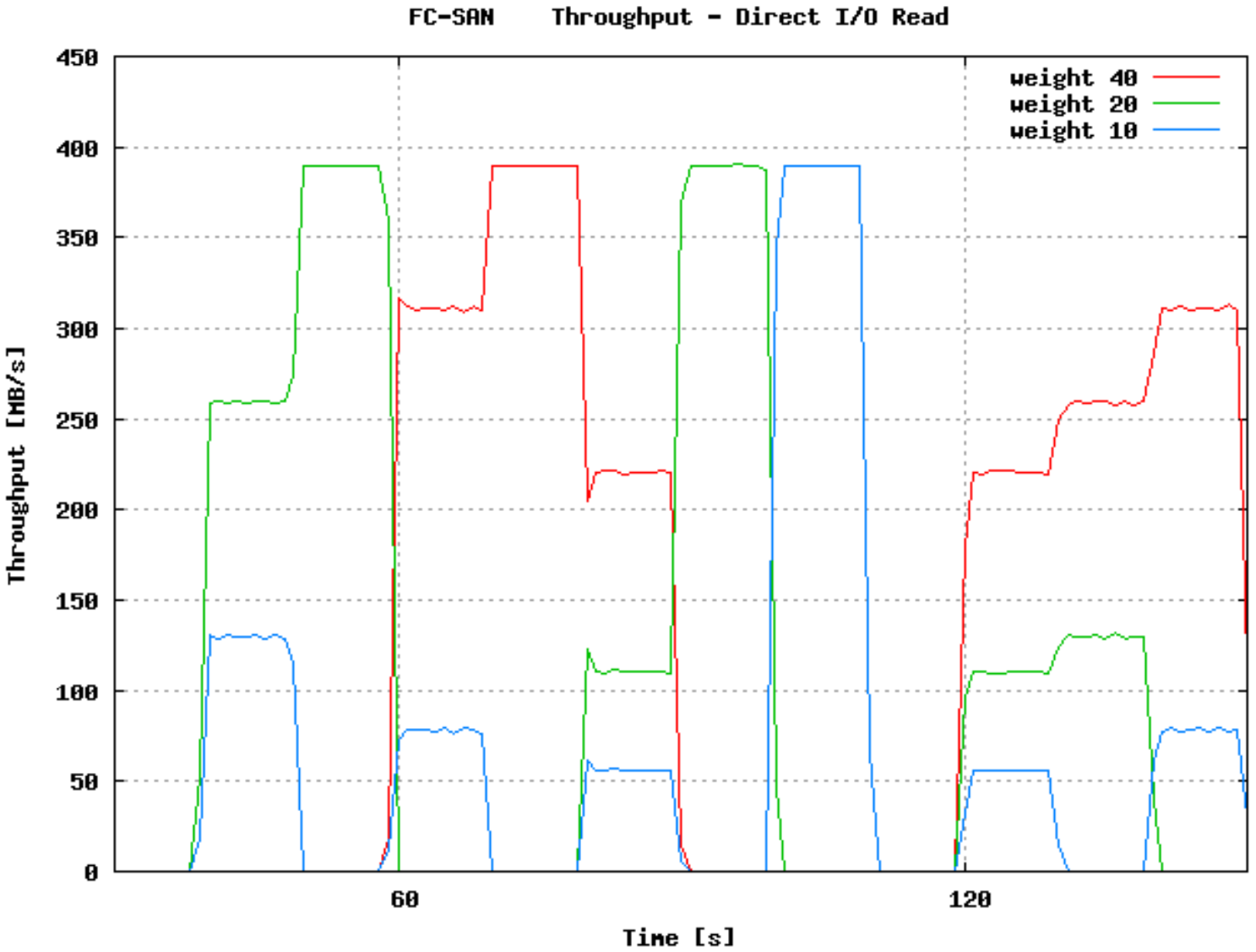


SATA Read (Buffered I/O)

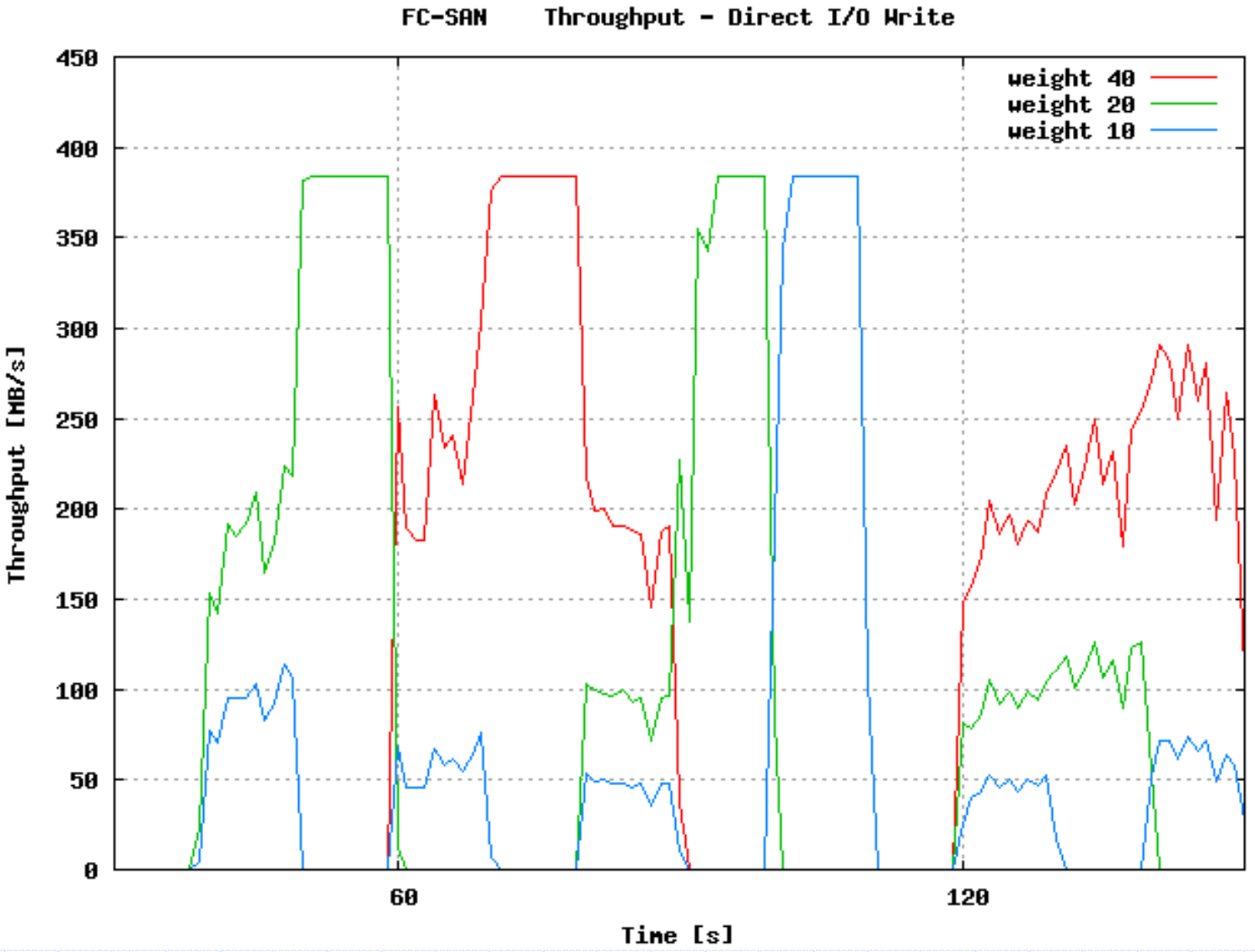
SATA Write (Buffered I/O)



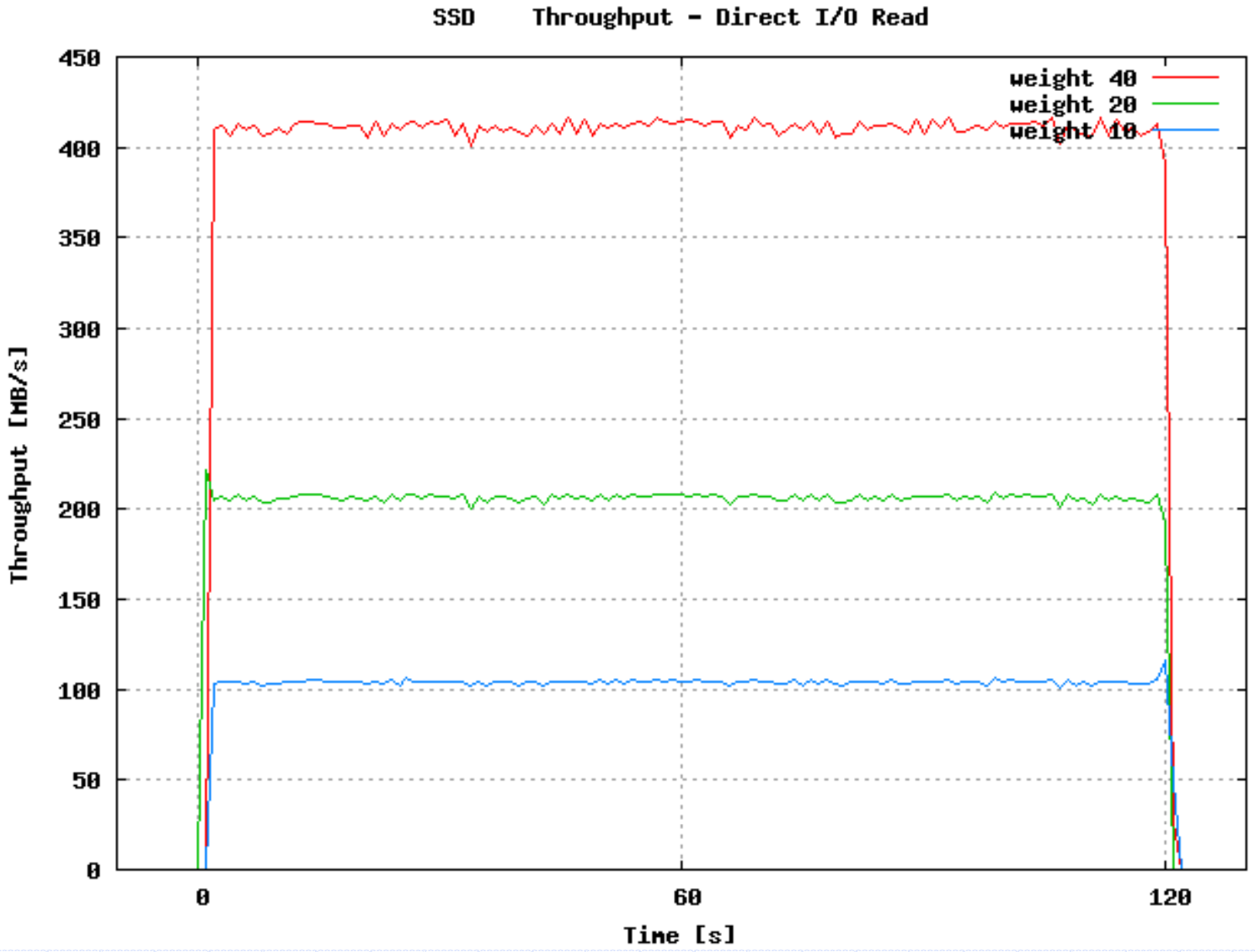
FC-SAN Read (Direct I/O)



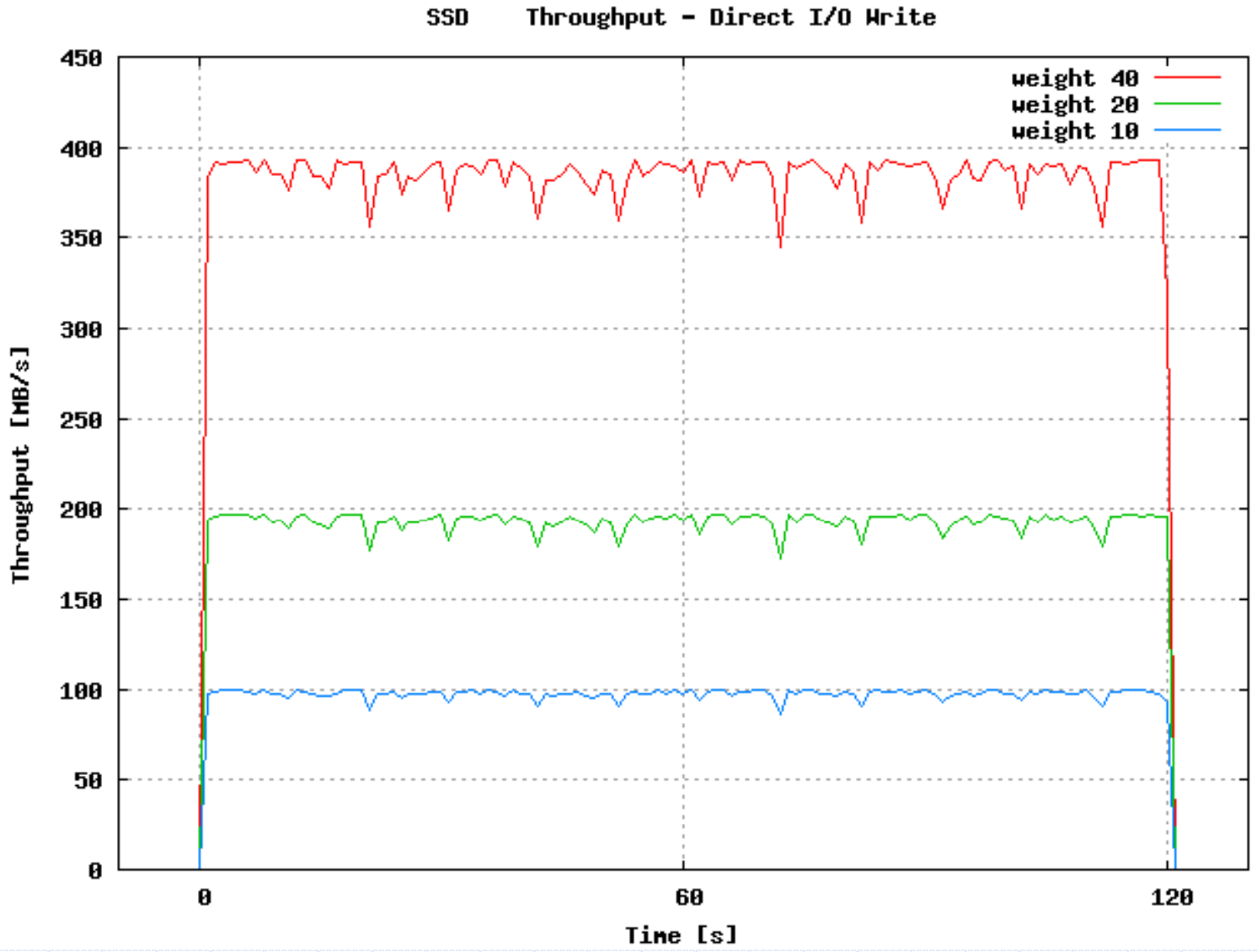
FC-SAN Write (Direct I/O)



SSD Read (Direct I/O)

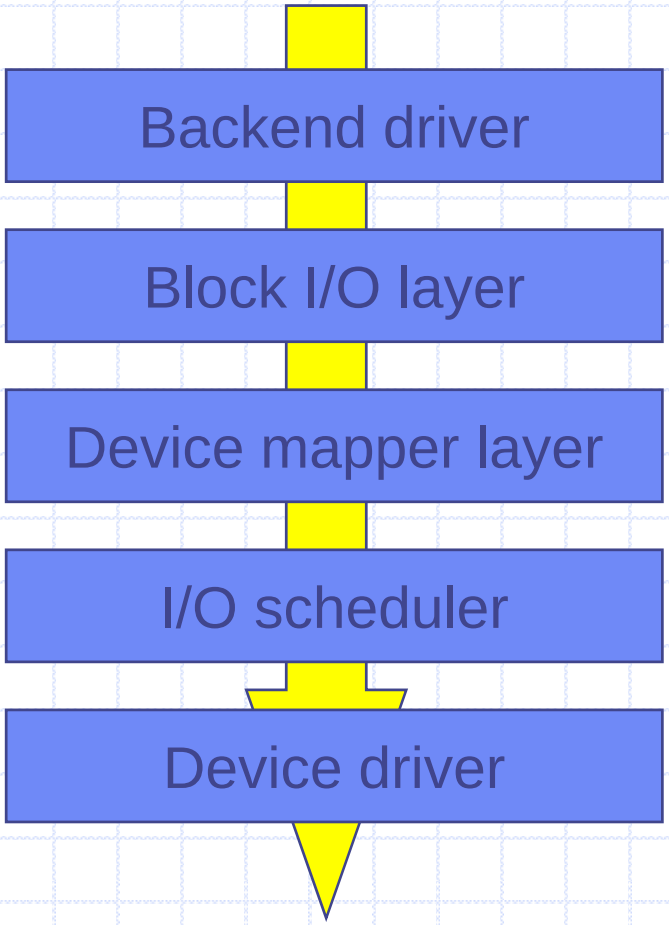


SSD Write (Direct I/O)



設計 dm-ioband

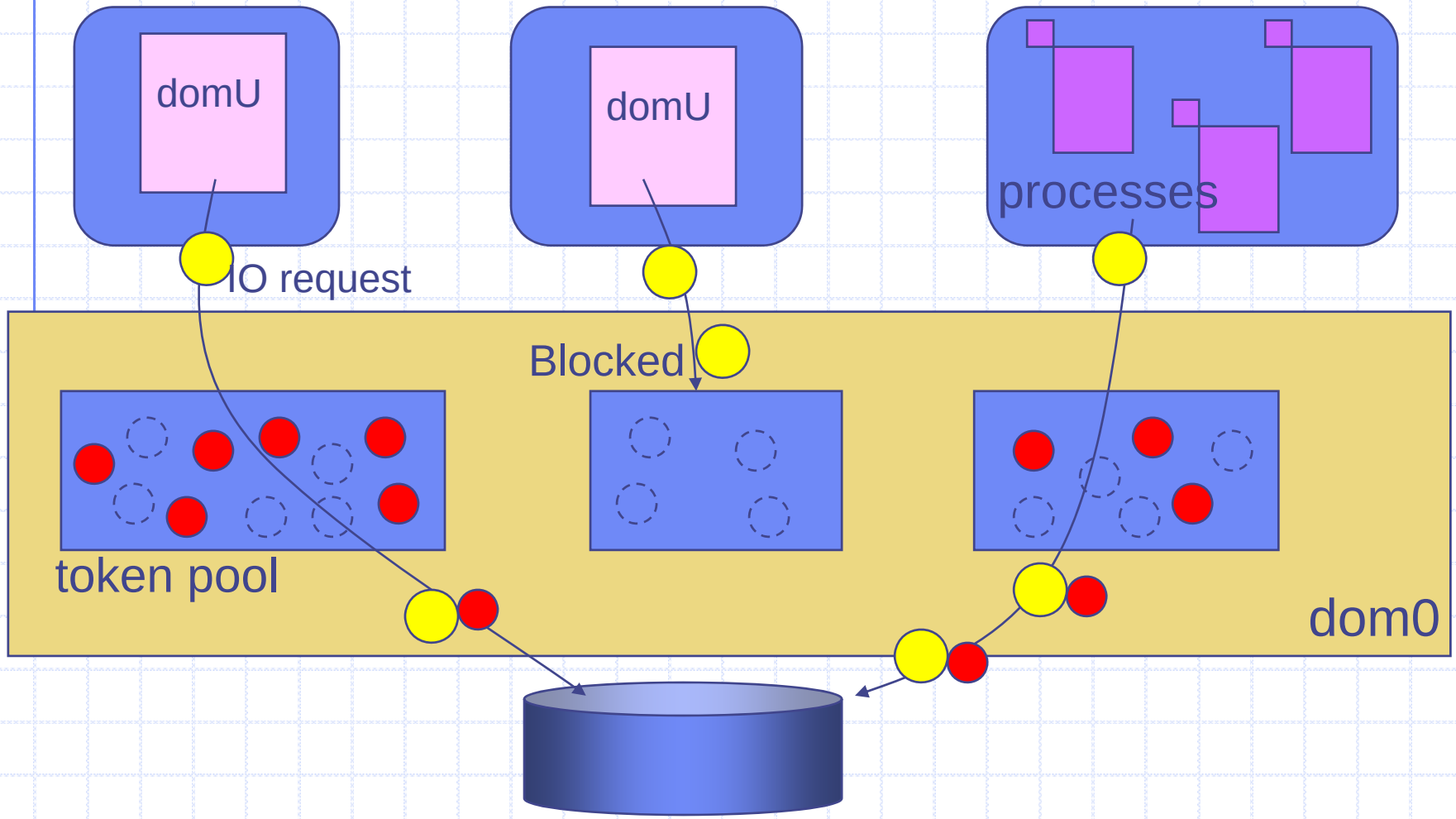
- どこに実装すべきか？
  - 汎用性を重視すると、Block I/O layer か、Device mapper layer
  - Linux単体でも利用できるべき
  - I/Oスケジューラの役目は、I/O効率向上に絞るべき
  - Device mapperは、モジュールの独立性が高い。が…
- まず、device mapperで実装



設計 dm-ioband

- I/Oグループを用意し、その中に仮想マシン、またはプロセス群を入れる。
- I/Oグループ毎に weightを設定する。
- weightに応じて、各I/Oグループにtokenを与える
- I/Oを発行する毎にtokenを消費する。
- tokenを使い切ったら、そのI/OグループはI/Oを発行できない。
- ActiveなI/Oグループがすべてtokenを使い切ったら、tokenを再補充する。

設計 dm-ioband



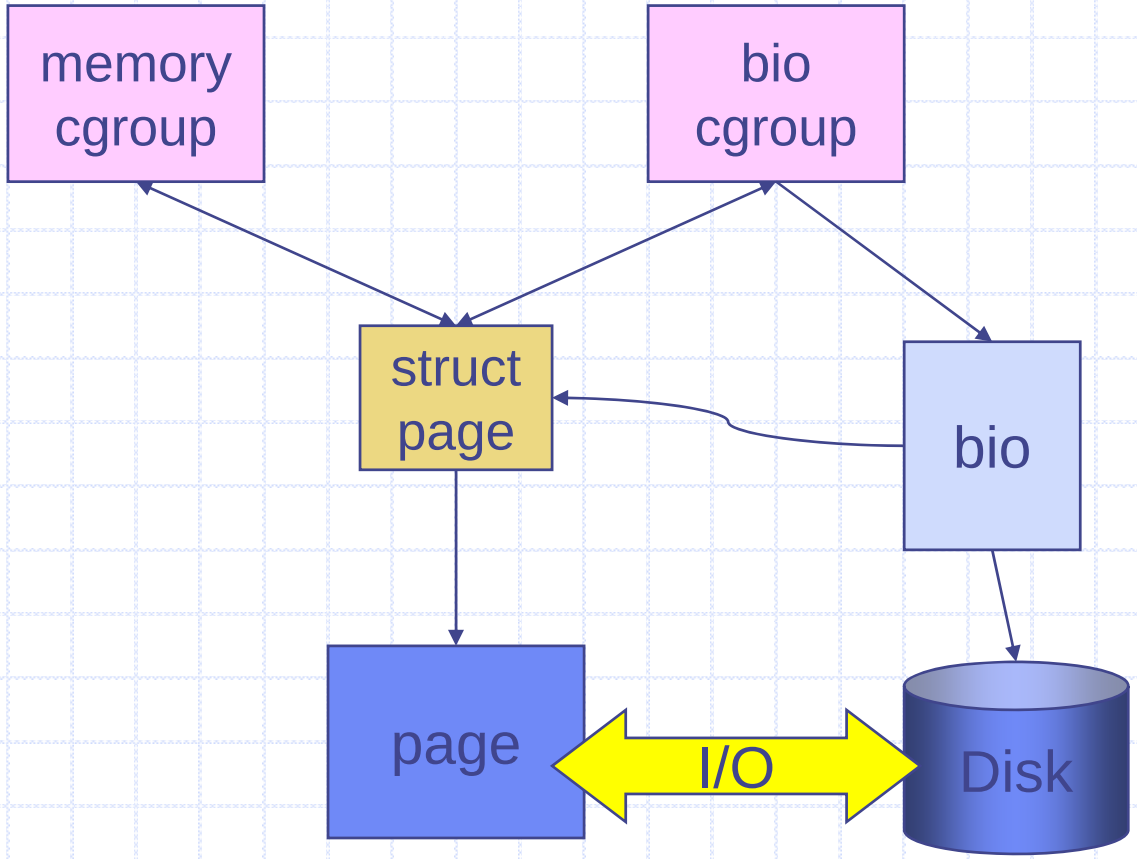
dm-ioband 実装の工夫

- inactiveなI/Oグループの余ったI/O帯域は、activeなI/Oグループで分け合う
- 複数の制御ポリシー：I/O回数をベースにした制御とI/O転送量をベースにした制御
- スループットを稼ぐため、完全に使い果たす前にtokenを再充填する
- 緊急性の高いI/O(page out要求など)は、tokenが残ってなくともブロックしない。

I/O追跡機能 bio-cgroup

- I/O要求から、発行元のI/Oグループを特定する機能
- 既にLinuxにあるcgroupのmemory controllerのフレームワークを拡張して実現
  - cgroup：資源を結びつけられたプロセスのグループ
  - cgroup memory controller：cgroupへのメモリ資源割り当てを実現する機能
- あるI/O要求の発行元は、そのI/O要求を行うメモリを所有しているcgroup

bio-cgroup の構造



現在のステータス

- dm-ioband　チューニング済み。ネイティブ環境での実行と変わらないI/Oスループット
- bio-cgroup再実装中。
  - cgroup memory controllerの変更による。オーバヘッドのより小さな実装となる。

今後の課題

- Linuxカーネルへのdm-iobandマージ
  - Device mapperのメンテナの反応が非常に鈍い。
  - Block I/O layerへのアルゴリズム移植も平行して行う。
- I/O追跡機能の改善
  - カーネル内部で生成するI/O要求の追跡。それの元となったプロセスのI/O要求のオーナを引き継ぐ。
  - Device mapperモジュールの修正が必要な点は、やはりネック
- 新しい帯域制御ポリシーの追加
  - SSD前提のポリシー。応答性保証型、スループット保証型

Any Questions?