KYLINSOFT
麒麟软件

Budget Fair Queueing+

胡 海
麒麟软件内核研发部

kylinos.cn

2021.10

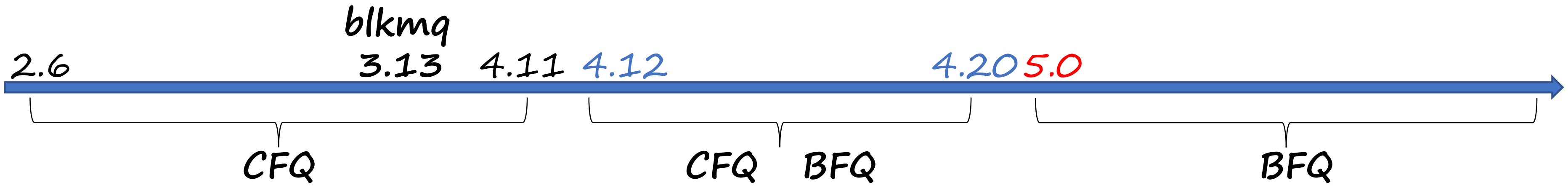
Contents

Introduce to BFQ

The balance between throughput and service guarantess

The works we done on BFQ

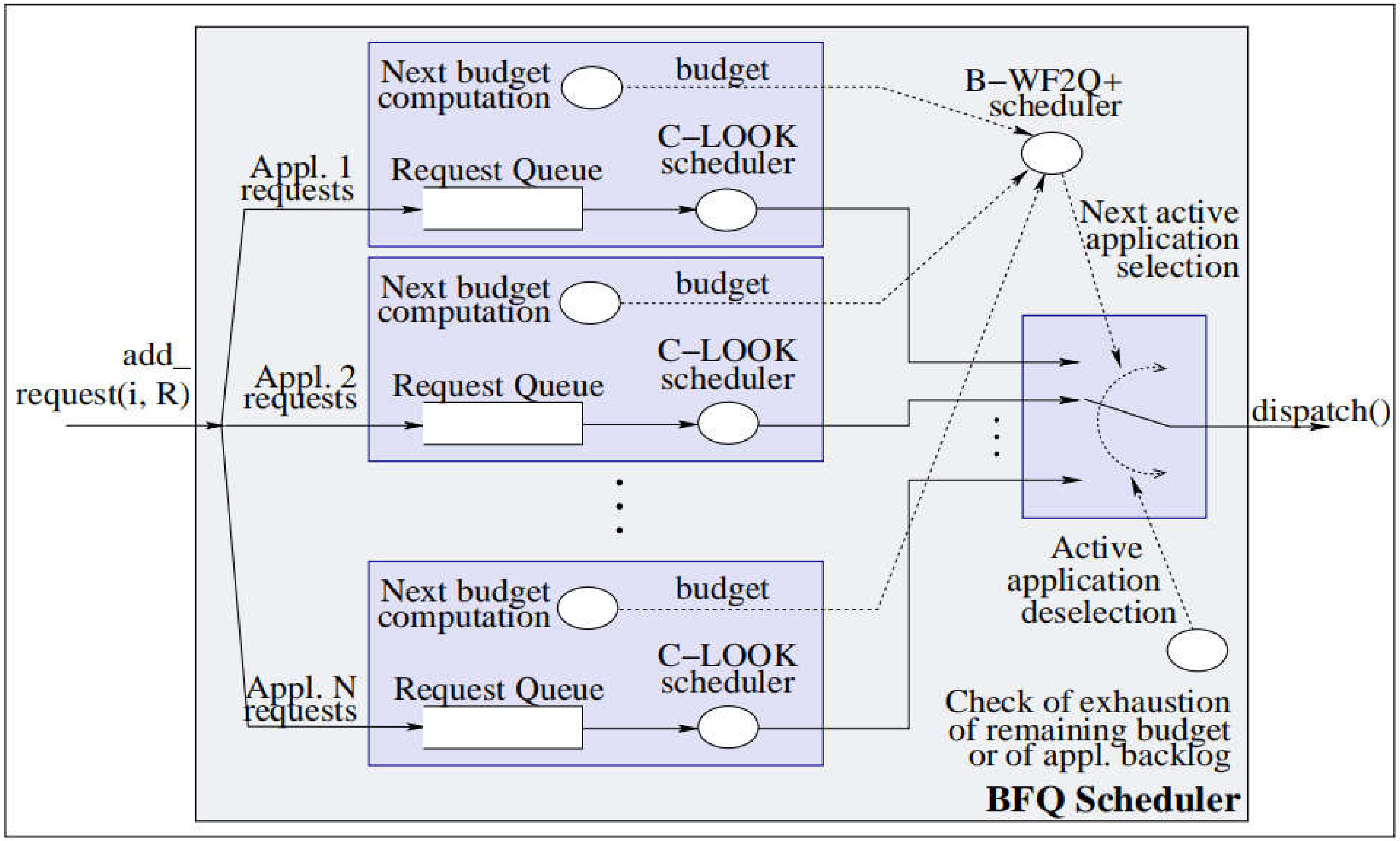
BFQ VS CFQ



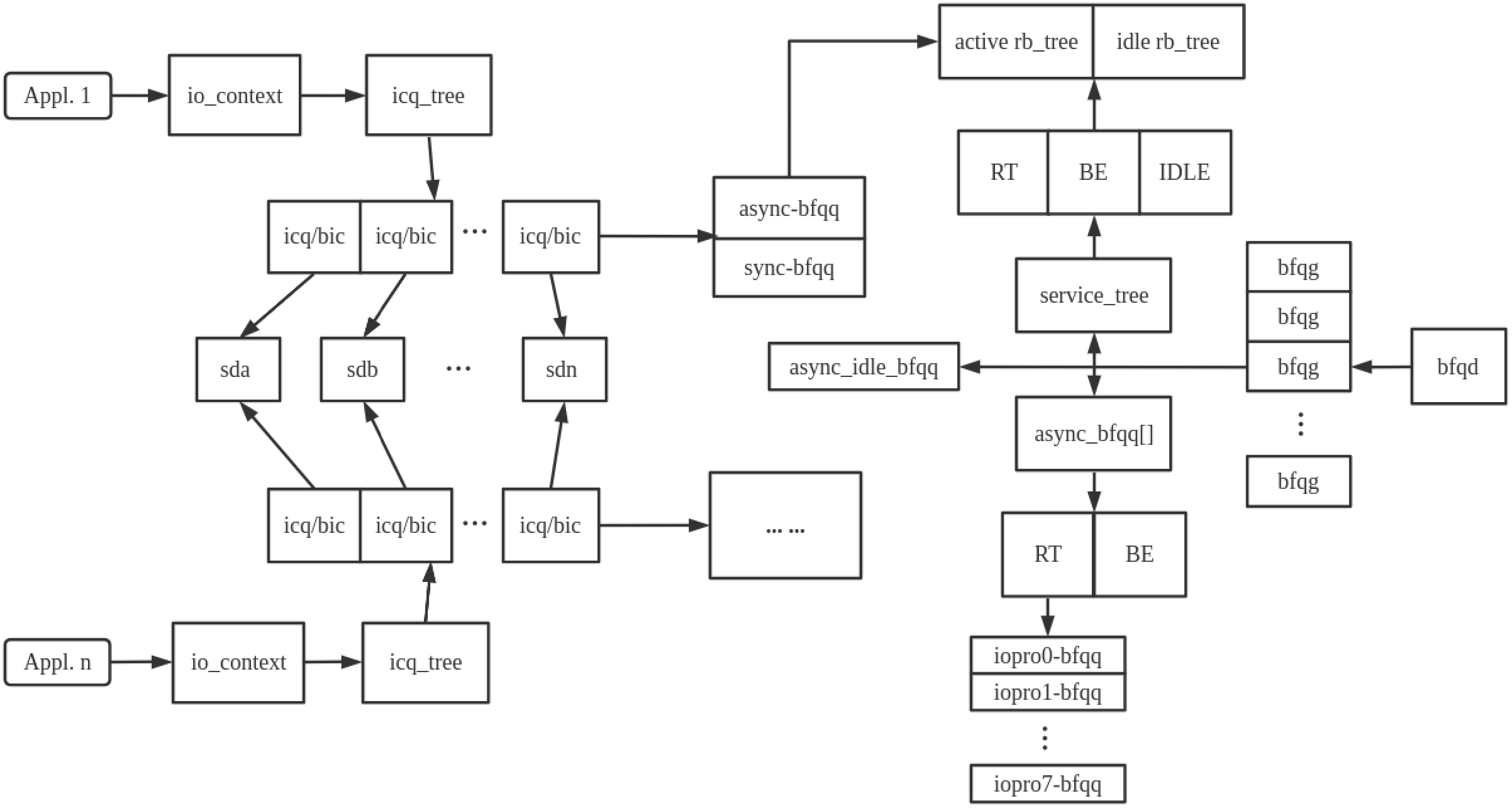
CFQ: **time-based,** In fact, even if the **same time slice** is assigned to two processes, they may get a **different throughput** each, as a function of the positions on the disk of their requests.

BFQ: can provide strong guarantees on bandwidth distribution because the assigned **budgets** are measured in number of sectors。

BFQ Logical Scheme



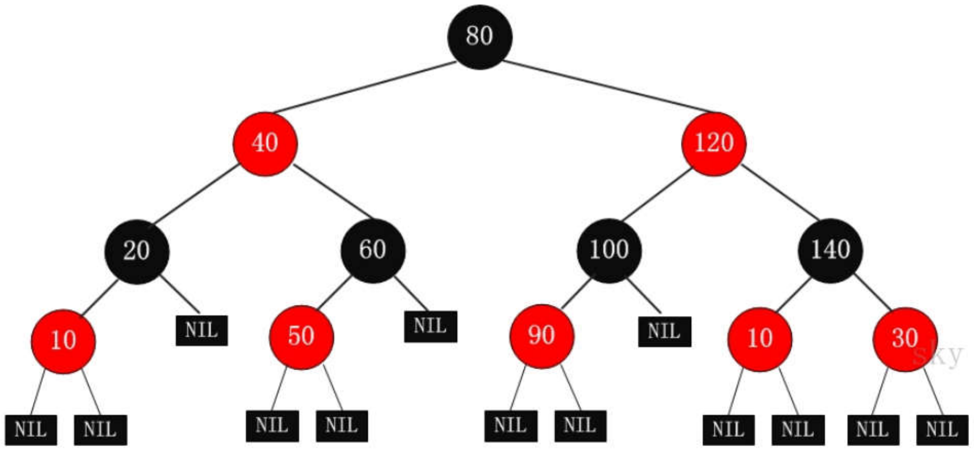
Data structure



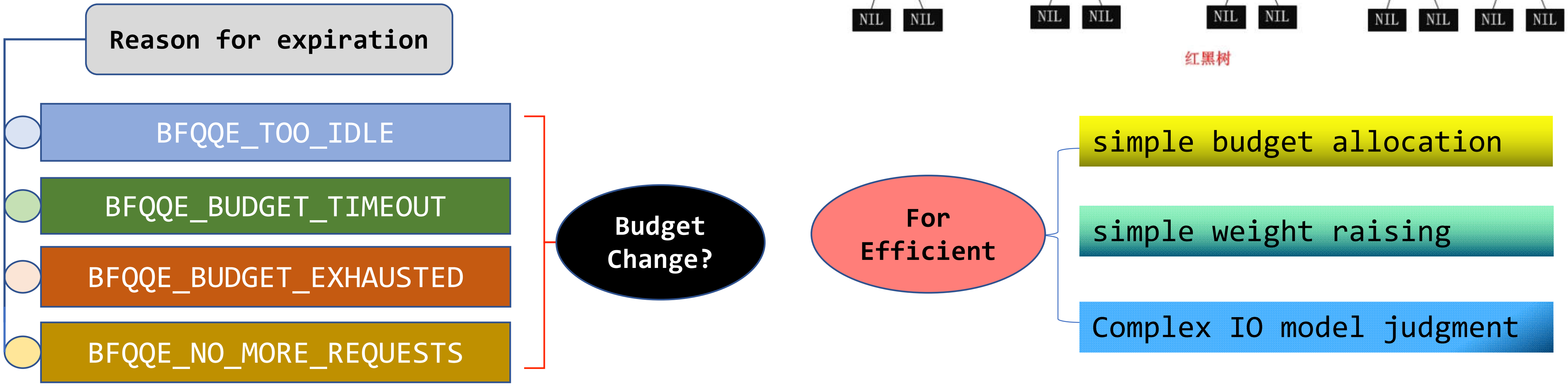
Entity / budget

$$entity.\boldsymbol{finish} = entity.start + \frac{entity.\boldsymbol{budget}}{entity.\boldsymbol{weight}}$$

$$budget\_timeout = jiffiest + \frac{HZ}{8} \times \boldsymbol{timeout\_coff}$$



红黑树



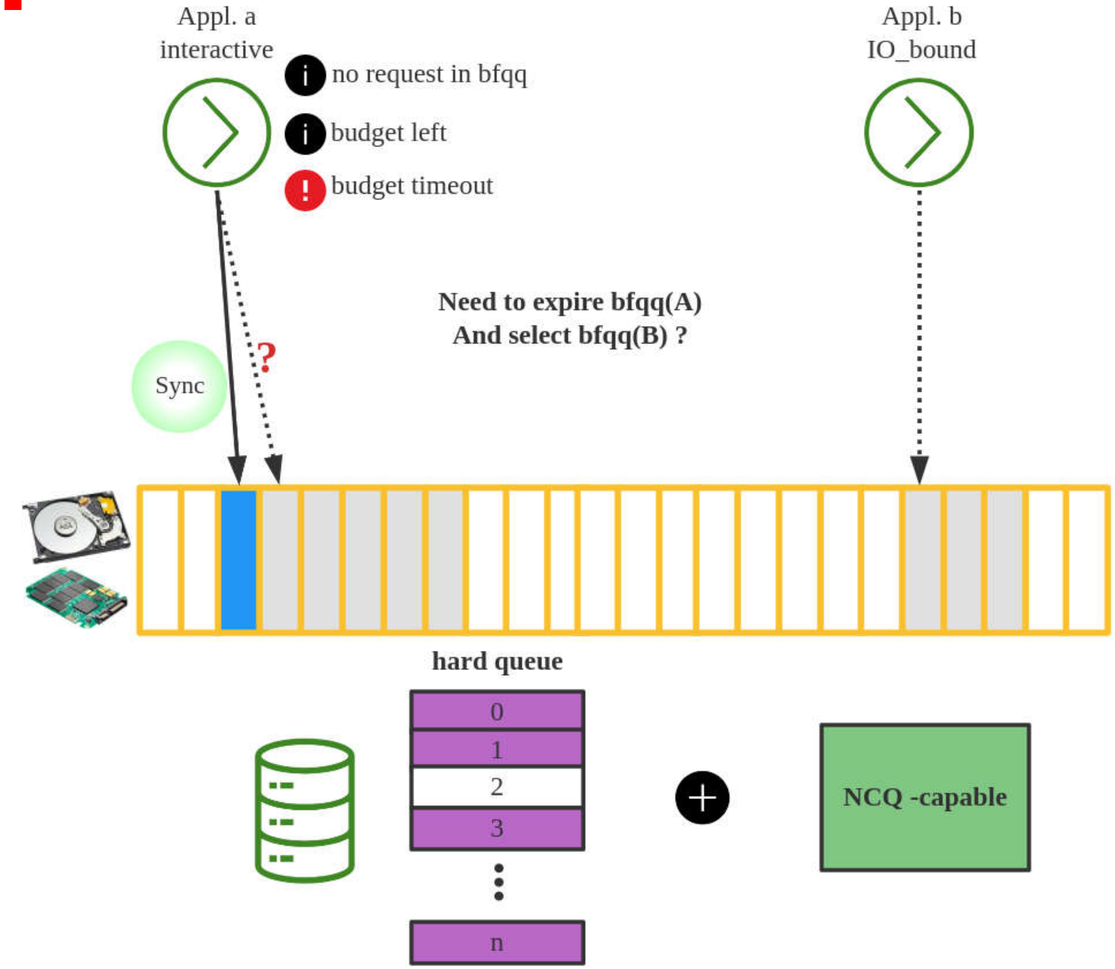
Contents

Introduce to BFQ

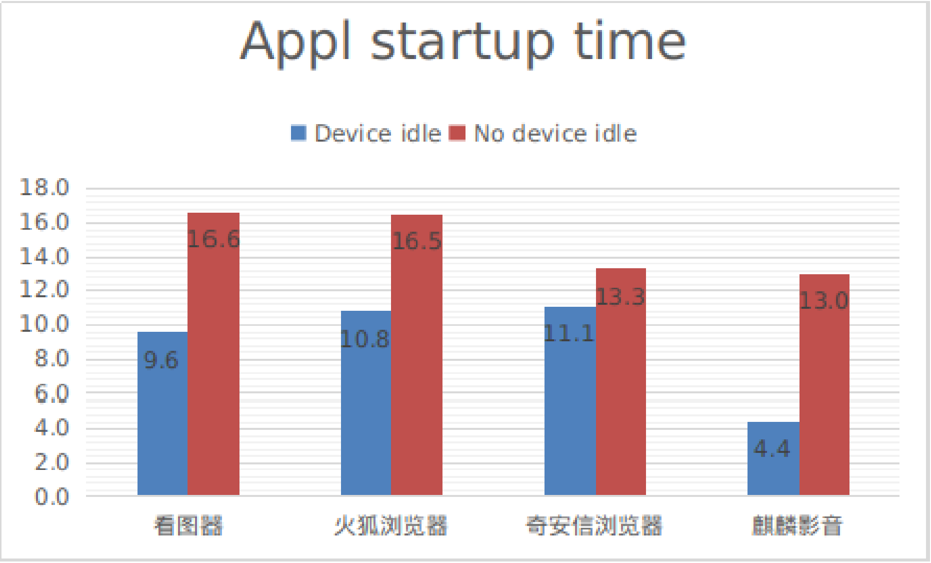
The balance between throughput and service guarantess

The works we done on BFQ

Device idle

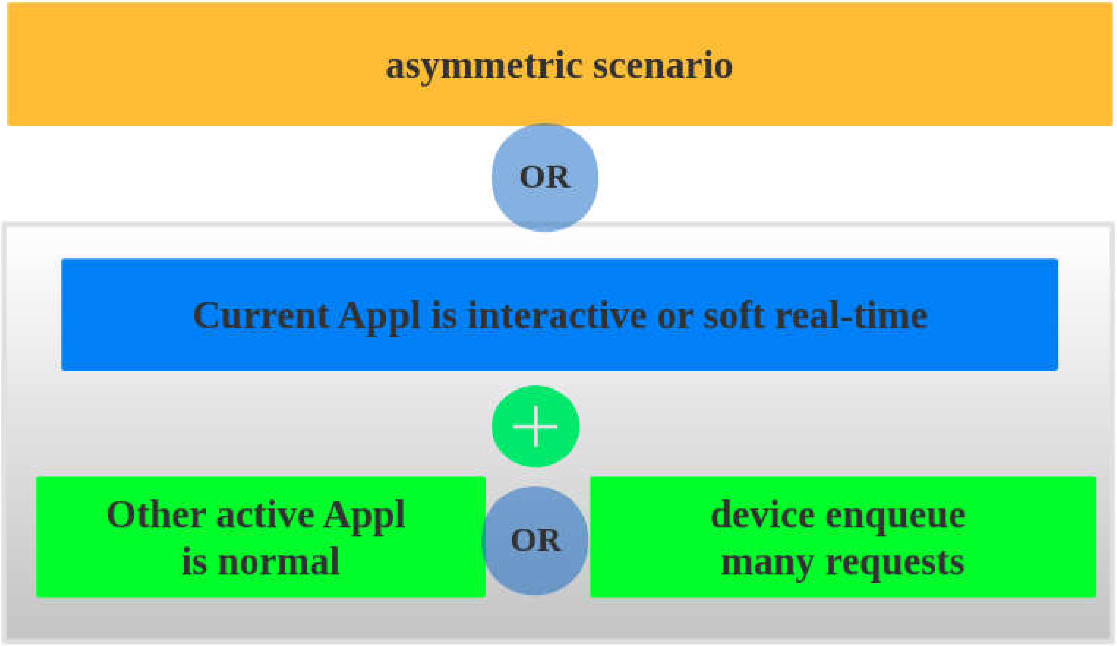


device idle either boosts the throughput (without issues), or is necessary to ensure service guarantees.



Idle for service guarantess or throughput

For service guarantess



asymmetric scenario

OR

Current Appl is interactive or soft real-time

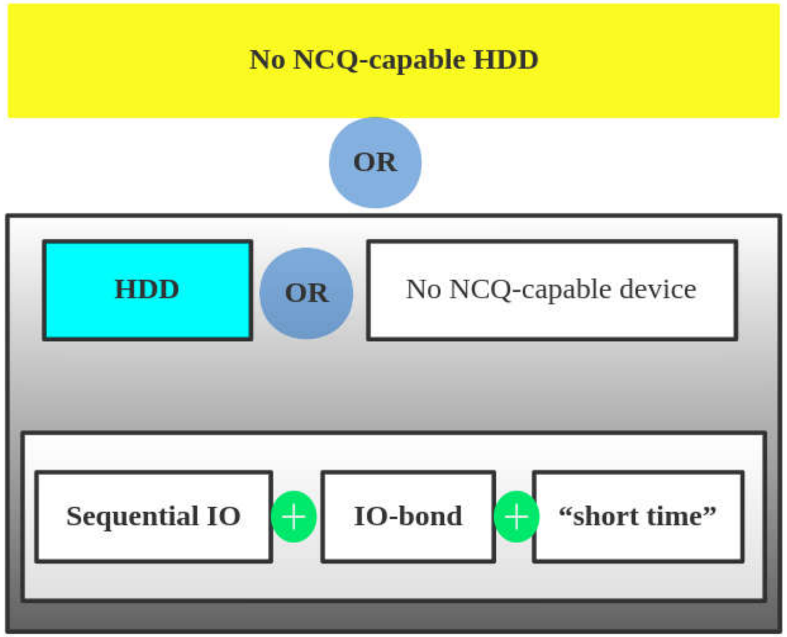
+

Other active Appl is normal

OR

device enqueue many requests

For throughput



No NCQ-capable HDD

OR

HDD

OR

No NCQ-capable device

Sequential IO

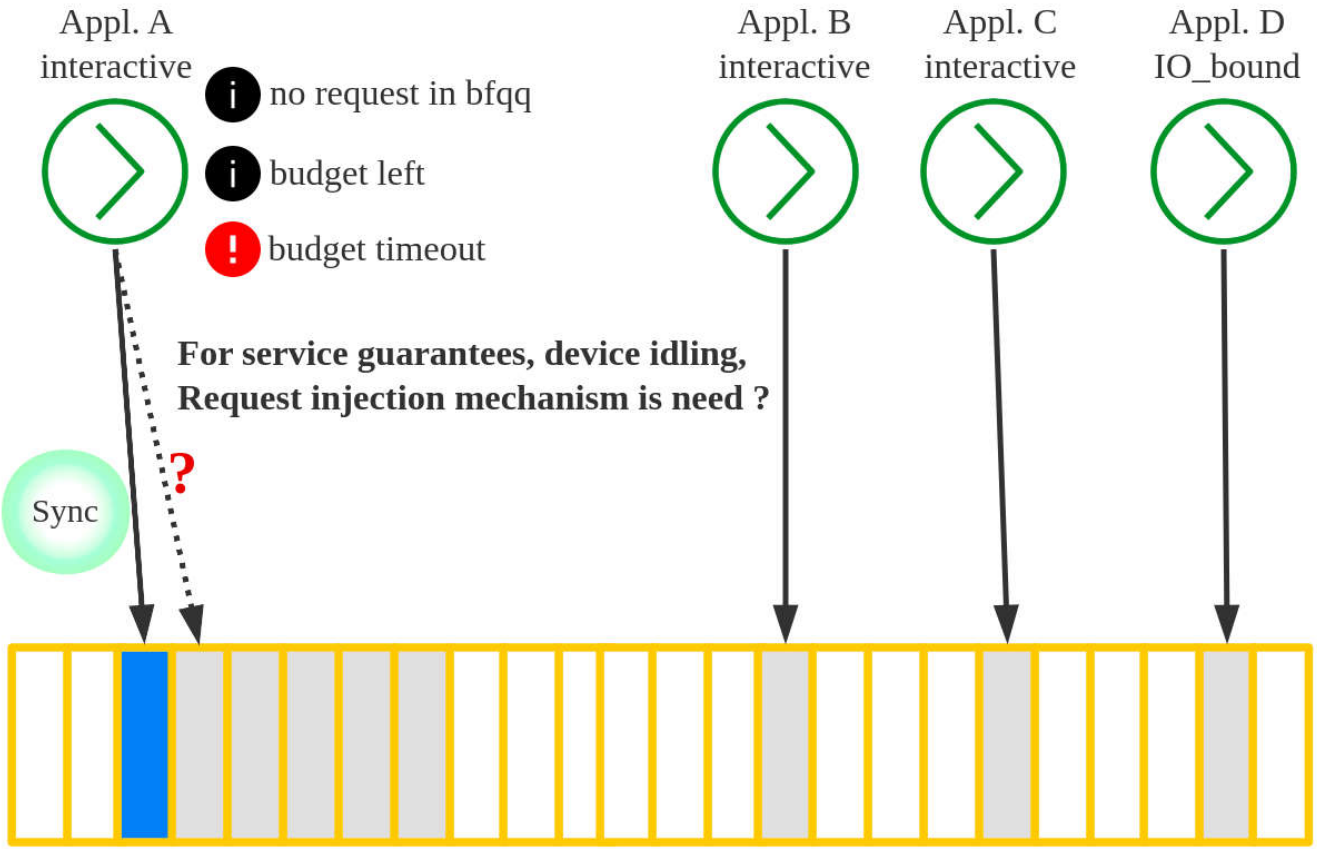
+

IO-bond

+

“short time”

Request injection mechanism

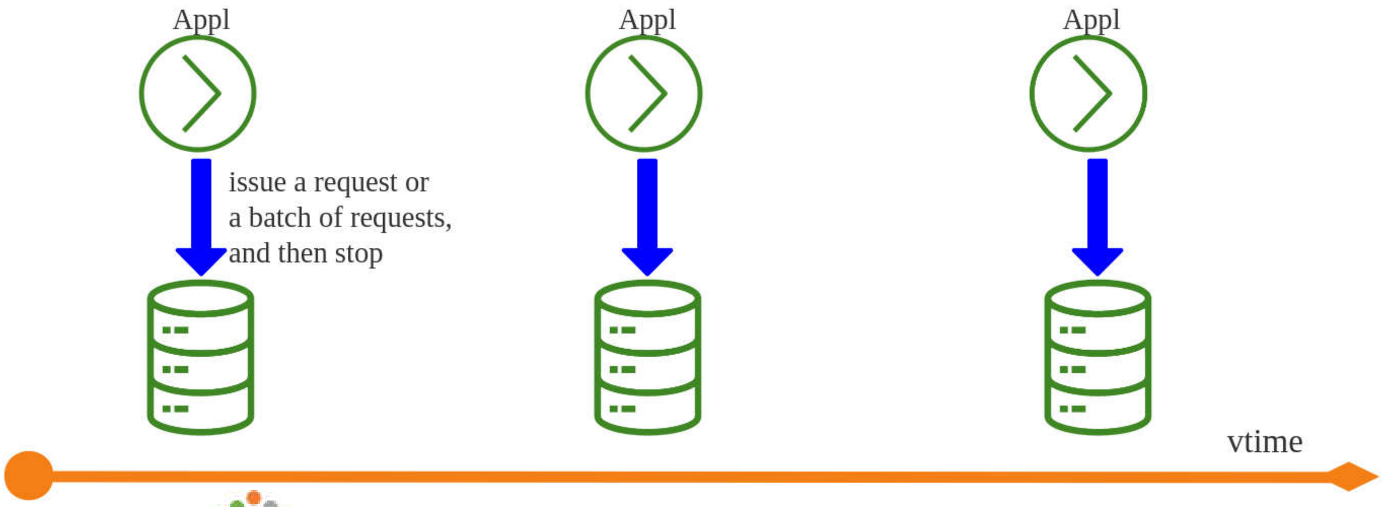


- Baseline for the total service time of the requests of bfqq

- Request inject, and comparing the total service time with the above baseline, Increase or decrease inject limit

- Periodically, reset the inject limit

Soft real-time appliction



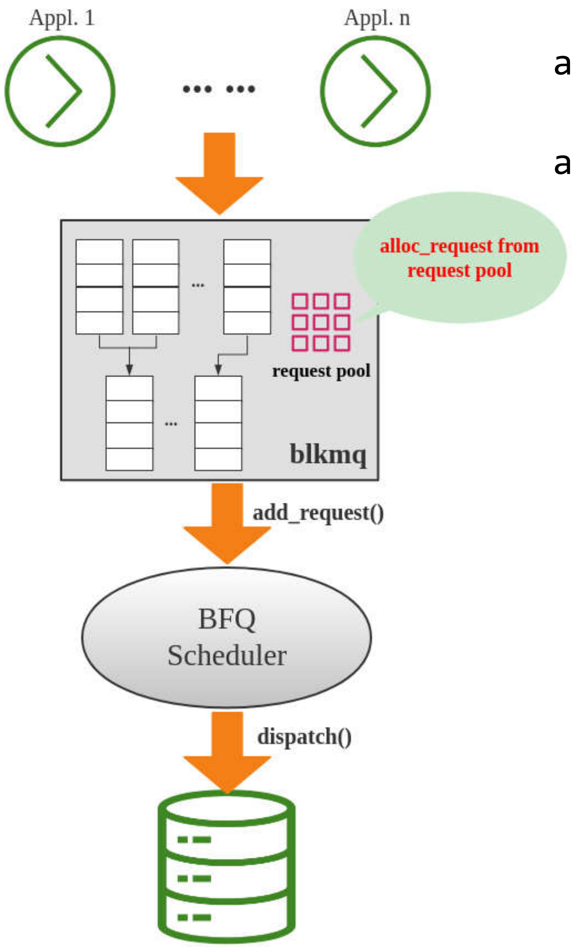
isochronous

the request pattern of the application is isochronous

Average bandwidth

the application must not require an average bandwidth higher than the approximate bandwidth required to playback or record a compressed high-definition video

Limit depth



appl.1(async) may have consumed all requests in the pool

appl.n has no request to use, cannot ensure service guaranteess

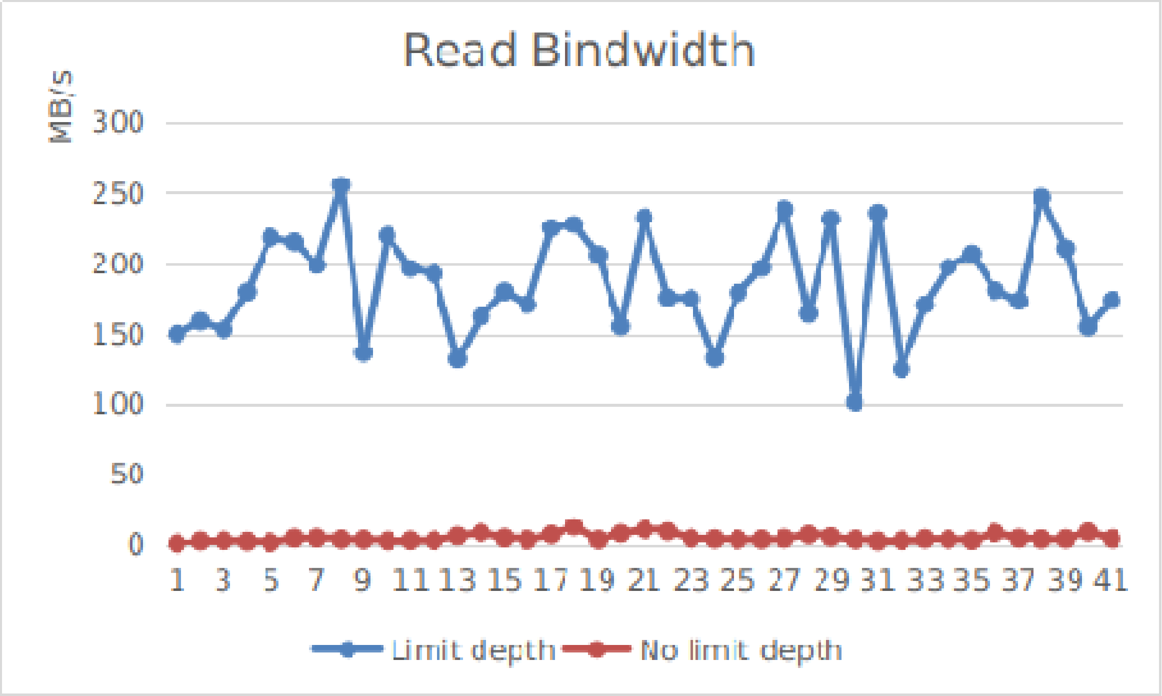
Appl — No weight-rasied

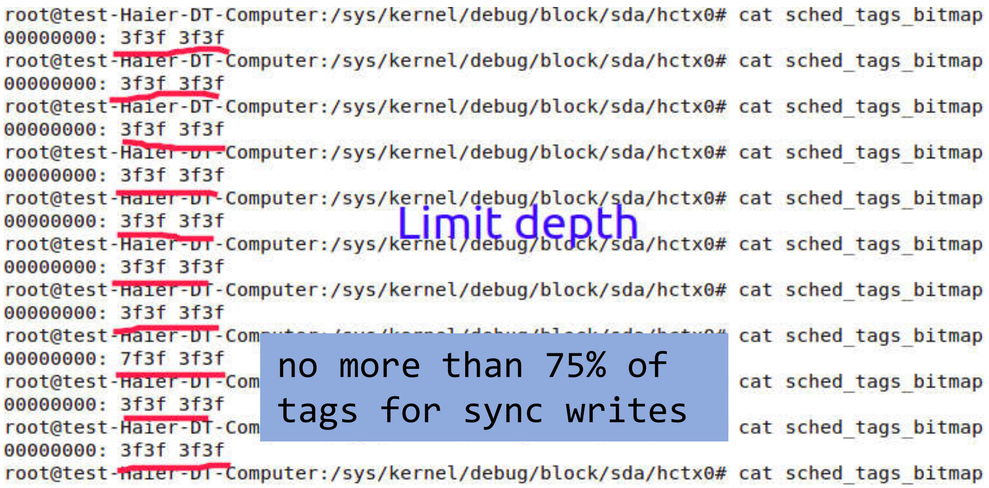
- no more than 50% of tags for async I/O
- no more than 75% of tags for sync writes

Appl — Weight-raised

- no more than ~18% of tags for async I/O
- no more than ~37% of tags for sync writes

Limit depth

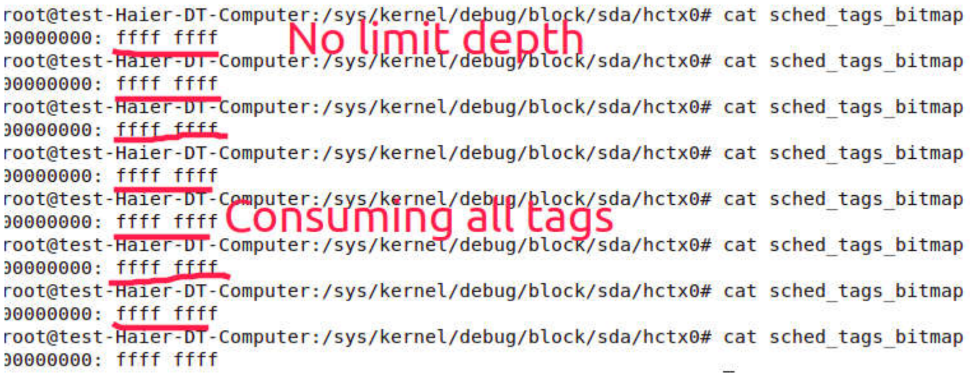




```
root@test-Haier-DT-Computer:/sys/kernel/debug/block/sda/hctx0# cat sched_tags_bitmap
00000000: 3f3f 3f3f
root@test-Haier-DT-Computer:/sys/kernel/debug/block/sda/hctx0# cat sched_tags_bitmap
00000000: 3f3f 3f3f
root@test-Haier-DT-Computer:/sys/kernel/debug/block/sda/hctx0# cat sched_tags_bitmap
00000000: 3f3f 3f3f
root@test-Haier-DT-Computer:/sys/kernel/debug/block/sda/hctx0# cat sched_tags_bitmap
00000000: 3f3f 3f3f
root@test-Haier-DT-Computer:/sys/kernel/debug/block/sda/hctx0# cat sched_tags_bitmap
00000000: 3f3f 3f3f
root@test-Haier-DT-Computer:/sys/kernel/debug/block/sda/hctx0# cat sched_tags_bitmap
00000000: 3f3f 3f3f
root@test-Haier-DT-Computer:/sys/kernel/debug/block/sda/hctx0# cat sched_tags_bitmap
00000000: 3f3f 3f3f
root@test-Haier-DT-Computer:/sys/kernel/debug/block/sda/hctx0# cat sched_tags_bitmap
00000000: 7f3f 3f3f
root@test-Haier-DT-Com                                        cat sched_tags_bitmap
00000000: 3f3f 3f3f
root@test-Haier-DT-Com                                        cat sched_tags_bitmap
00000000: 3f3f 3f3f
root@test-Haier-DT-Computer:/sys/kernel/debug/block/sda/hctx0# cat sched_tags_bitmap
```

Limit depth

no more than 75% of tags for sync writes

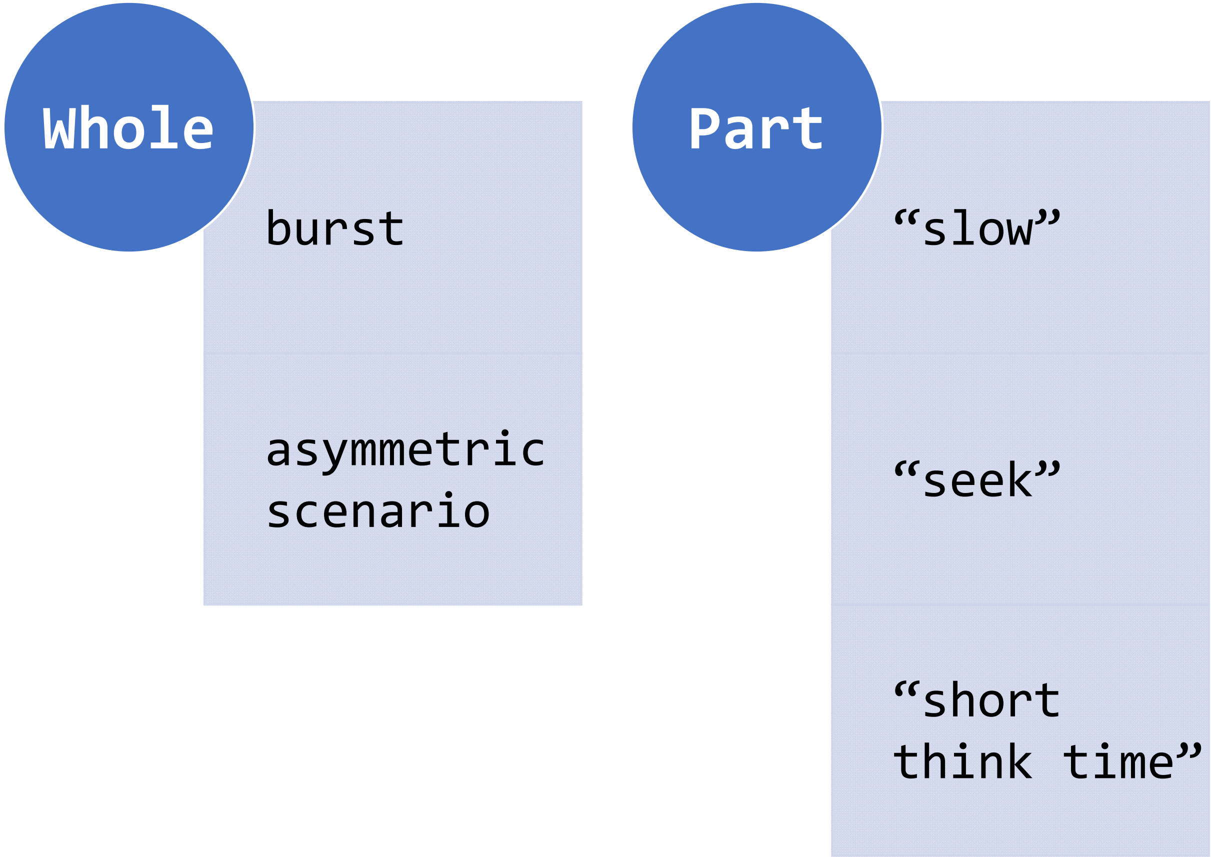


```
root@test-Haier-DT-Computer:/sys/kernel/debug/block/sda/hctx0# cat sched_tags_bitmap
00000000: ffff ffff
root@test-Haier-DT-Computer:/sys/kernel/debug/block/sda/hctx0# cat sched_tags_bitmap
00000000: ffff ffff
root@test-Haier-DT-Computer:/sys/kernel/debug/block/sda/hctx0# cat sched_tags_bitmap
00000000: ffff ffff
root@test-Haier-DT-Computer:/sys/kernel/debug/block/sda/hctx0# cat sched_tags_bitmap
00000000: ffff ffff
root@test-Haier-DT-Computer:/sys/kernel/debug/block/sda/hctx0# cat sched_tags_bitmap
00000000: ffff ffff
root@test-Haier-DT-Computer:/sys/kernel/debug/block/sda/hctx0# cat sched_tags_bitmap
00000000: ffff ffff
root@test-Haier-DT-Computer:/sys/kernel/debug/block/sda/hctx0# cat sched_tags_bitmap
00000000: ffff ffff
root@test-Haier-DT-Computer:/sys/kernel/debug/block/sda/hctx0# cat sched_tags_bitmap
00000000: ffff ffff
```

No limit depth

Consuming all tags

Other measures



Whole

burst

asymmetric scenario

Part

“slow”

“seek”

“short think time”

Contents

Introduce to BFQ

The balance between throughput and service guarantess

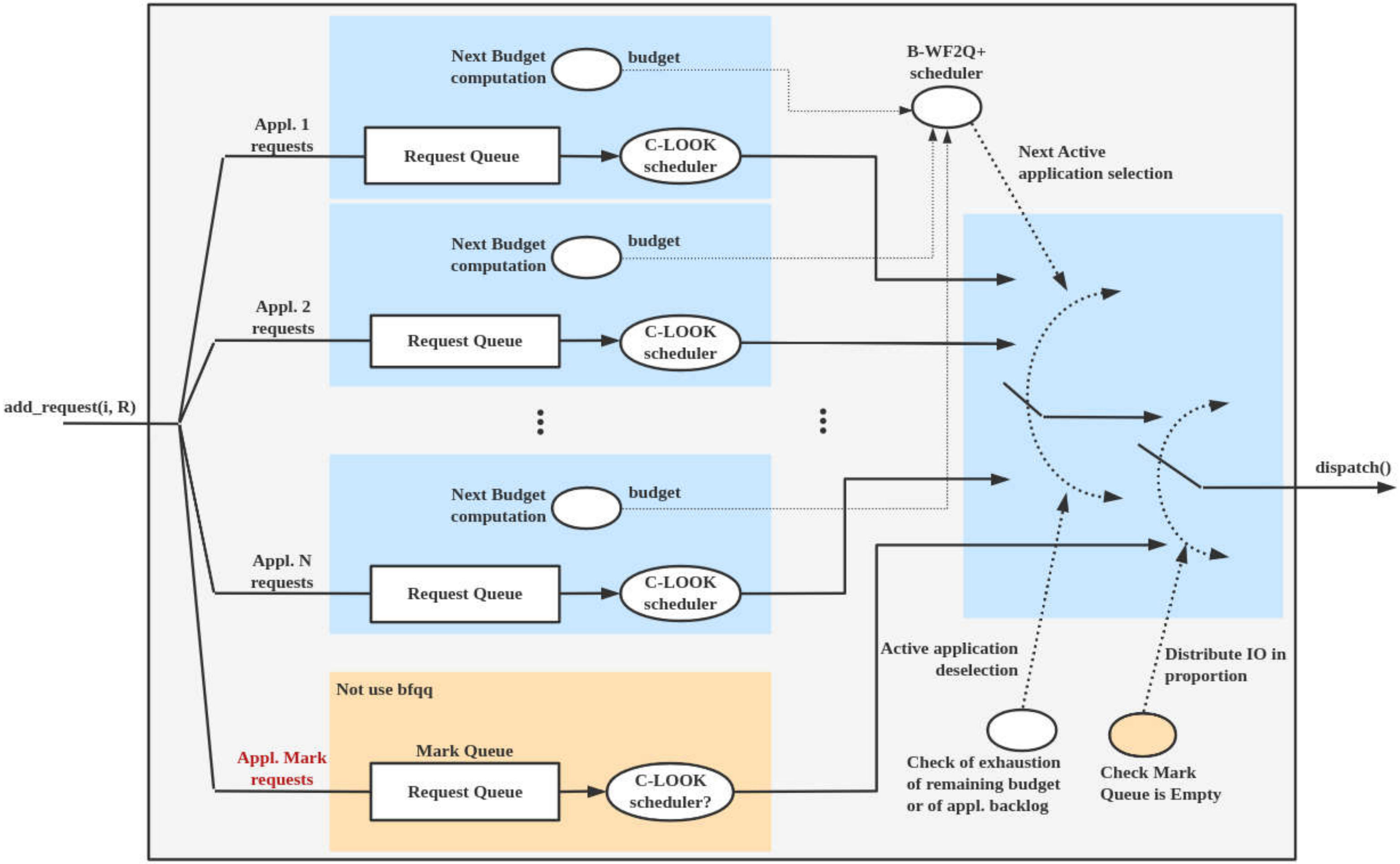
The works we done on BFQ

Background

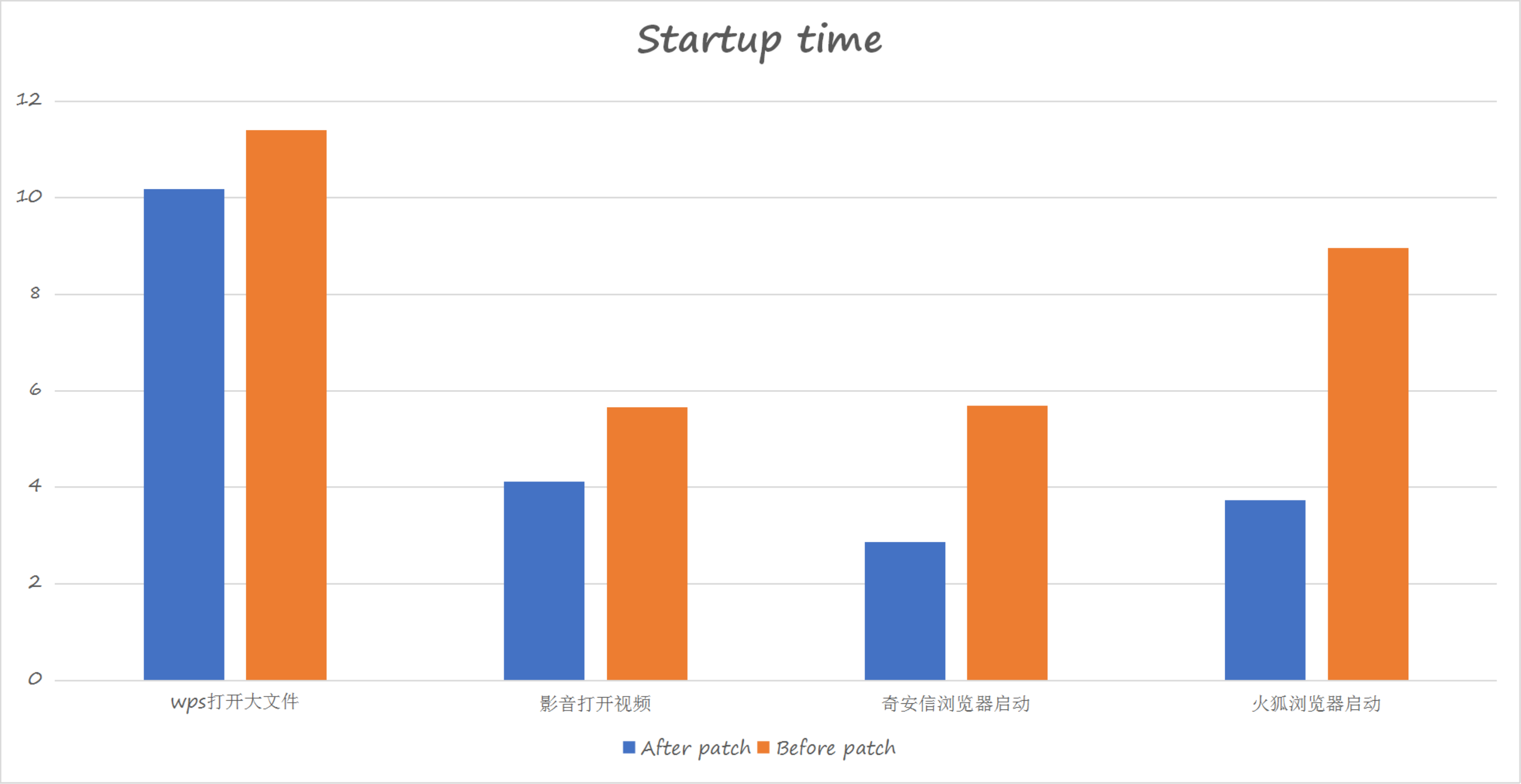
1 Optimize startup time for large applications

2 Needed for customized service?

Implementation



Optimization



KYLINSOFT
麒麟软件

谢 谢

APPLEDIX

Reference:

[1] http://algogroup.unimore.it/people/paolo/disk_sched/mst-2015.pdf

[2] http://algogroup.unimore.it/people/paolo/disk_sched/bfq-techreport.pdf

[3] http://algo.ing.unimo.it/people/paolo/disk_sched

[4] https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/tree/block?h=v5.8

[5] https://www.kernel.org/doc/Documentation/block/bfq-iosched.txt

[6] https://blog.csdn.net/feelabclihu/article/details/105502167