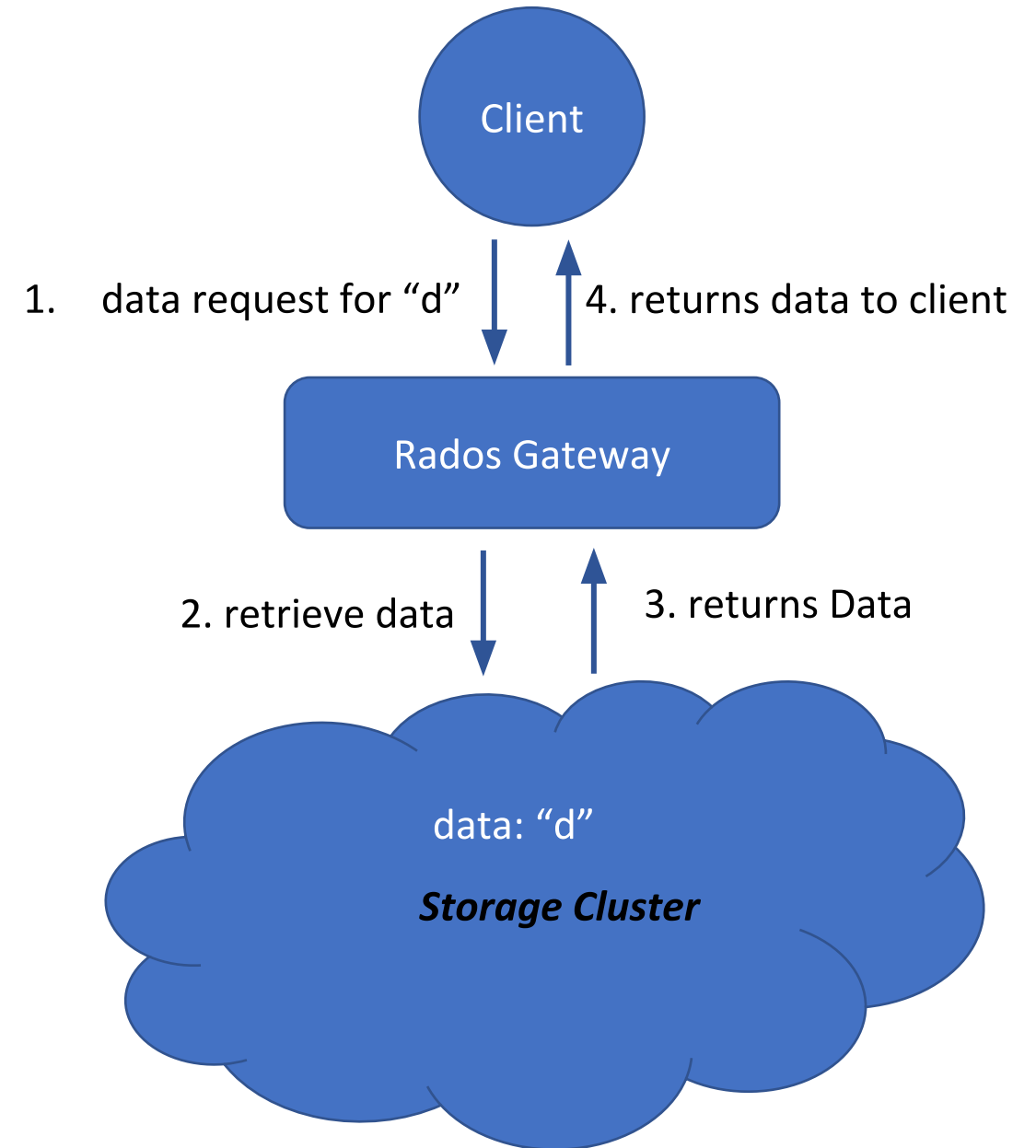


Ceph RGW-Prefetching

Ali Raza, Amin M., Bissenbay D.

What is CEPH?

- Distributed Storage System
- **Key Components** [Related to Project]
 - Client (Applications)
 - Rados Gateway
 - Storage Cluster

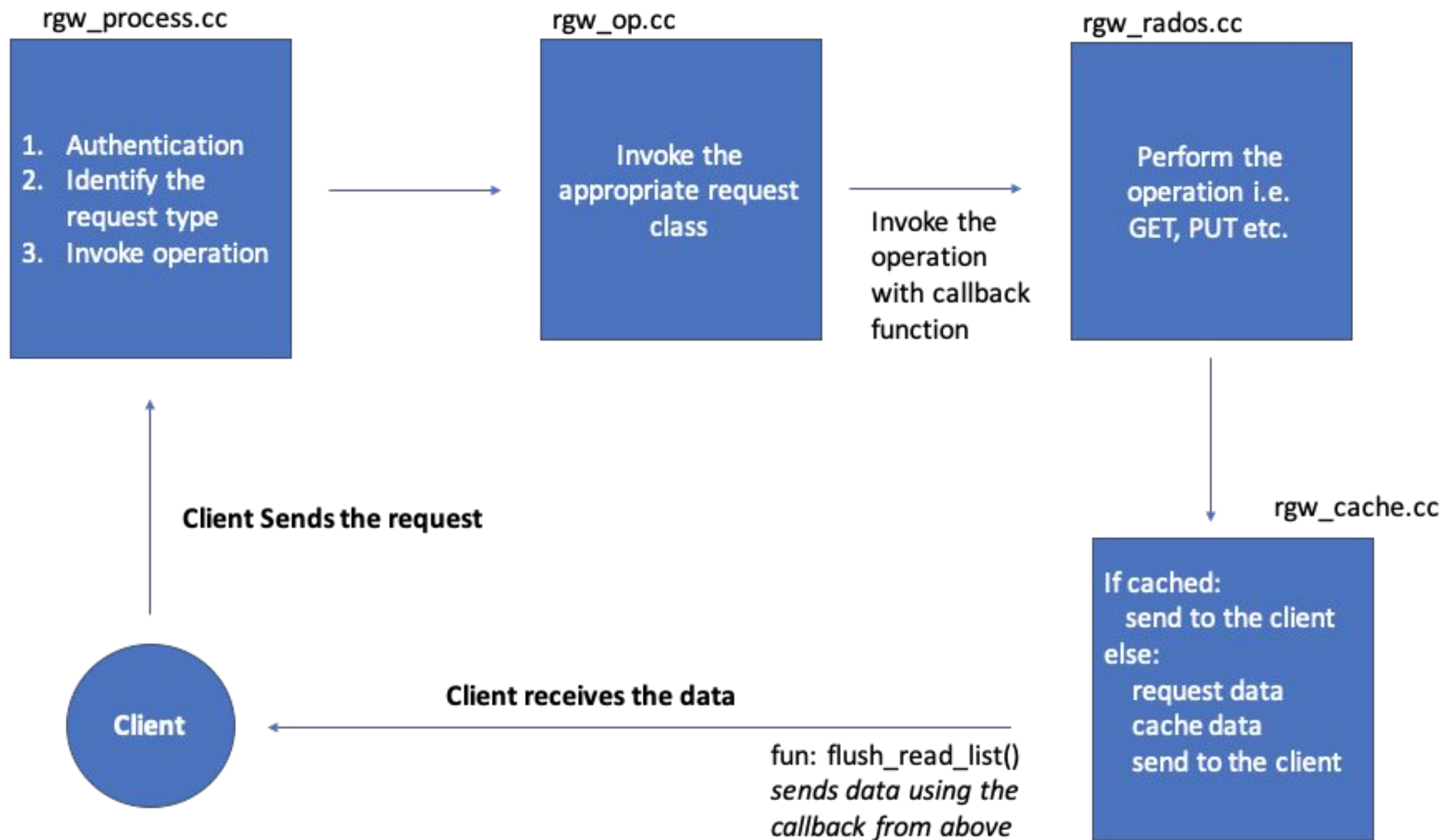


Prefetching in Ceph

- ❑ Current caching scheme is re-active
- ❑ Goal: Improve performance in first access
 - ❑ Lower latency
 - ❑ Higher throughput
- ❑ Two type of prefetching
 - ❑ on-demand prefetching
 - ❑ pro-active prefetching in rgw

What we have done ...

- ❏ Uploaded & retrieved files to Ceph storage
 - ❏ S3cmd
 - ❏ Swift
 - ❏ Boto3
- ❏ Understanding the code
 - ❏ Deployed Ceph on multiple machines
 - ❏ Data flow
 - ❏ Debugging the code with GDB
 - ❏ Brainstorming with mentors to find the best development strategy
- ❏ In Progress:
 - ❏ Prefetching based on user request
 - ❏ Pro-active prefetching on block level
 - ❏ Evaluation



Prefetching based on user request

Design Decisions

- Introduce “new” type of request
- Overload the GET request (add header)

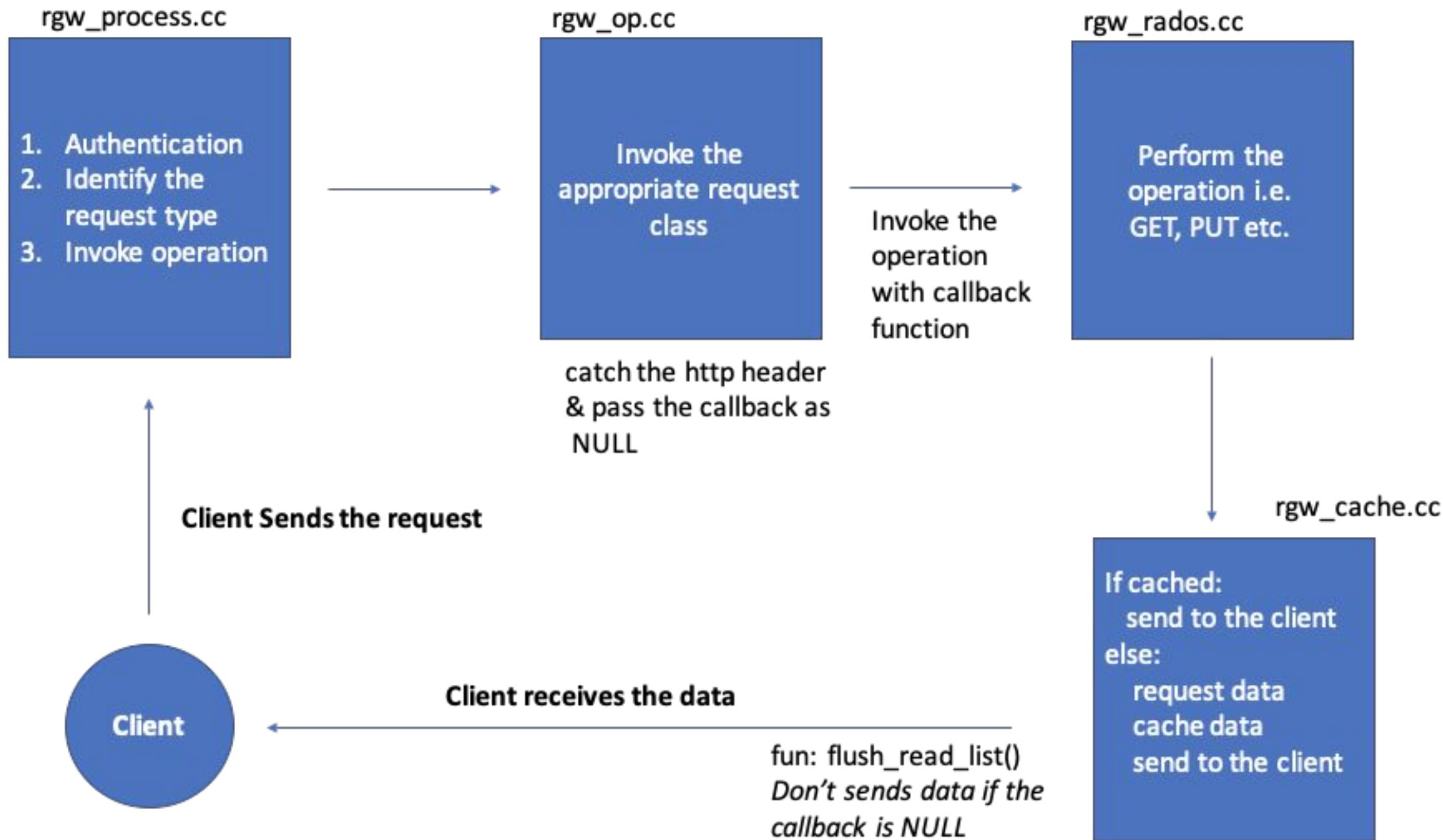
Prefetching based on user request

Design Decisions

- Introduce “new” type of request
- **Overload the GET request (add header)**

Why?

- Ease of implementation and usage
 - no need to change the client side
 - no more than 3 lines of code to add an extra header



Initial Progress

- Successfully caught the header in rgw_op.cc
- Invoked the correct function

Issue:

- Client still receives the data back

Debugging with GDB and talking to mentors from Redhat, as well as using the IRC channels

Pro-active prefetching

- ❑ GitHub blk-prefetching branch
- ❑ A prefetching C++ class to talk with cache
 - ❑ We have implemented:
 - ❑ The requested File, size of requested chunk and its offsets
 - ❑ The size of the File and passing it to the prefetching class
 - ❑ User's authentication
- ❑ A thread pool for prefetching in addition to caching thread pool

Evaluation

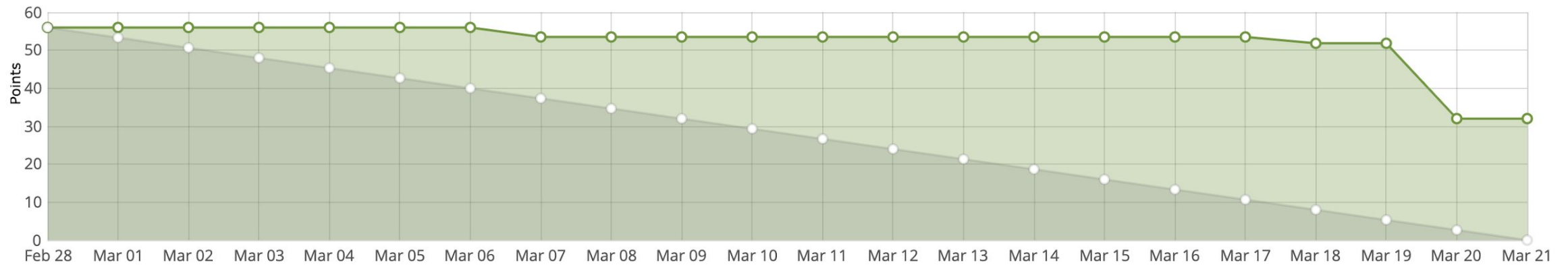
- COSBench
 - the software to evaluate the Ceph performance
 - With/without Prefetching
- Meeting with RedHat mentors
 - Customizing COSBench to evaluate Ceph performance

Burndown chart

CEPH-RGW-PREFETCHI... SPRINT 3 28 FEB 2019-21 MAR 2019



 21% ∨ 56 total points | 12 completed points | 3 open tasks | 7 closed tasks ⇄ |  0 iocaine doses



Challenges

- Request based caching
 - In case of over loading http GET request (shouldn't send the data back)
- Pro-active prefetching
 - We are prefetching parts of a file
 - What if we want to prefetch other files based on the current file?
- Do things in the right way
 - This project would be part of Ceph
 - So we need to design and develop

What we will be working next Sprint

[Hopefully :D]

- Finishing up request based prefetching
- Finishing up the pro-active prefetching
- Start using COSBench and changing it