QCon 全球软件开发大会
INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE

BEIJING 2017

分布式海量二进制文件存储系统

夏鸣（LinkedIn）

主办方 Geekbang. 极客邦科技 InfoQ

Data Scale at LinkedIn

- 400+ million users
- 800 million puts and gets per day
- 120 TB data accessed per day
- Multiple data centers across the globe
- Request rates is doubled over the past 12 months.

Data source: Ambry: LinkedIn's scalable geo-distributed object store, *ACM SIGMOD,* San Francisco, June 2016.
Picture source: http://linkeddominator.com/blog/adding-linkedin-connections-url-can-grease-business/

Before the Ice Age

- Media Server
  - Monothilic
  - Not scalable
  - No full control
  - Expensive

Challenges

- Data volume can be huge
  - Grow capacity linearly as needed
- Number of objects can be huge
  - Fast and efficient indexing (O(1))
- Data can be critical
  - High survivability during partial failure
- Data need to be available all the time
  - Multi-9s availability
- Data access across globe
  - Users in U.S., E.U., and Asia should not see big difference in access time
- Etc...

Here Ambry Comes

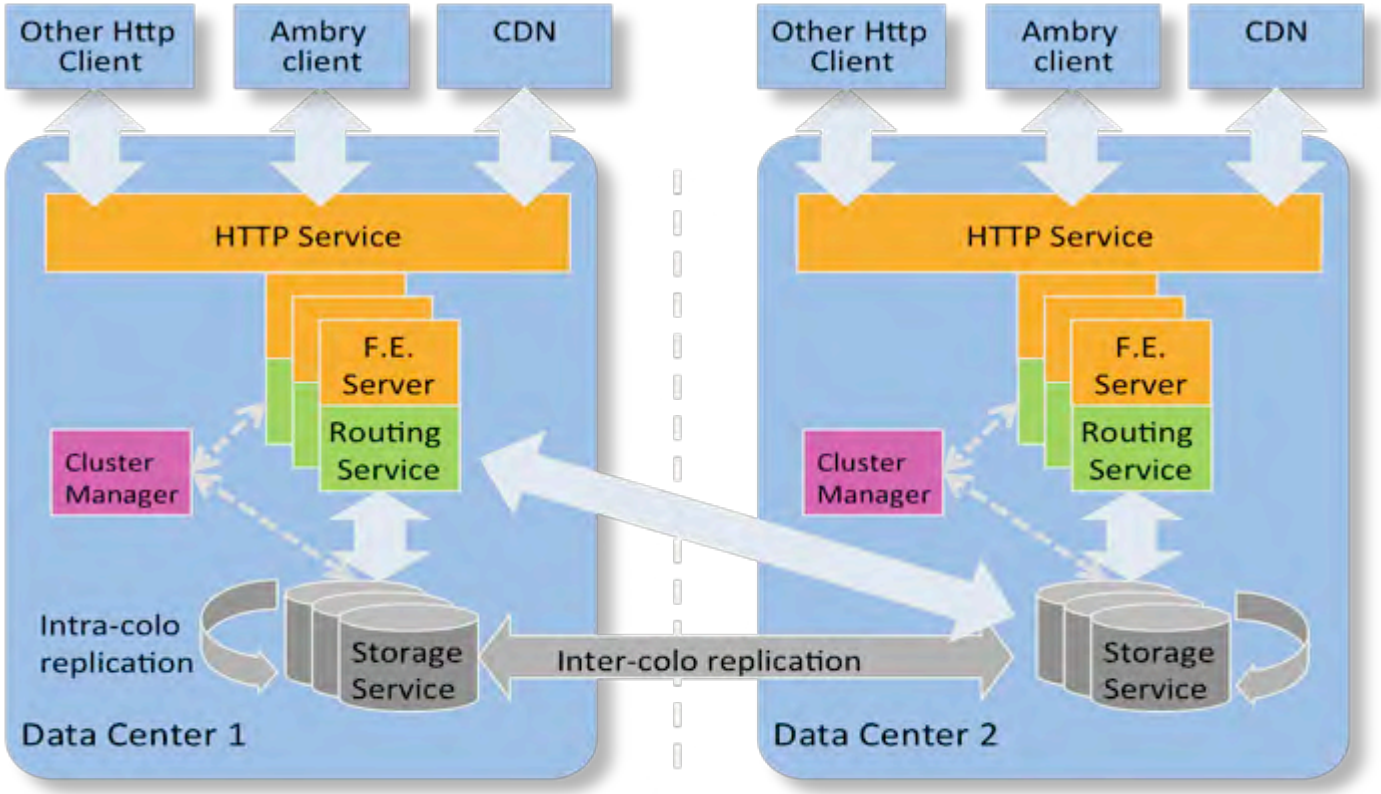
Ambry is LinkedIn's scalable geo-distributed object store

- **Scalable**: grow as needed
- **Geo-distributed**: multiple data centers across the globe
- **Object store**: immutable binary object with support of metadata

Not

- a key-value store
- a distributed file system
- transactional

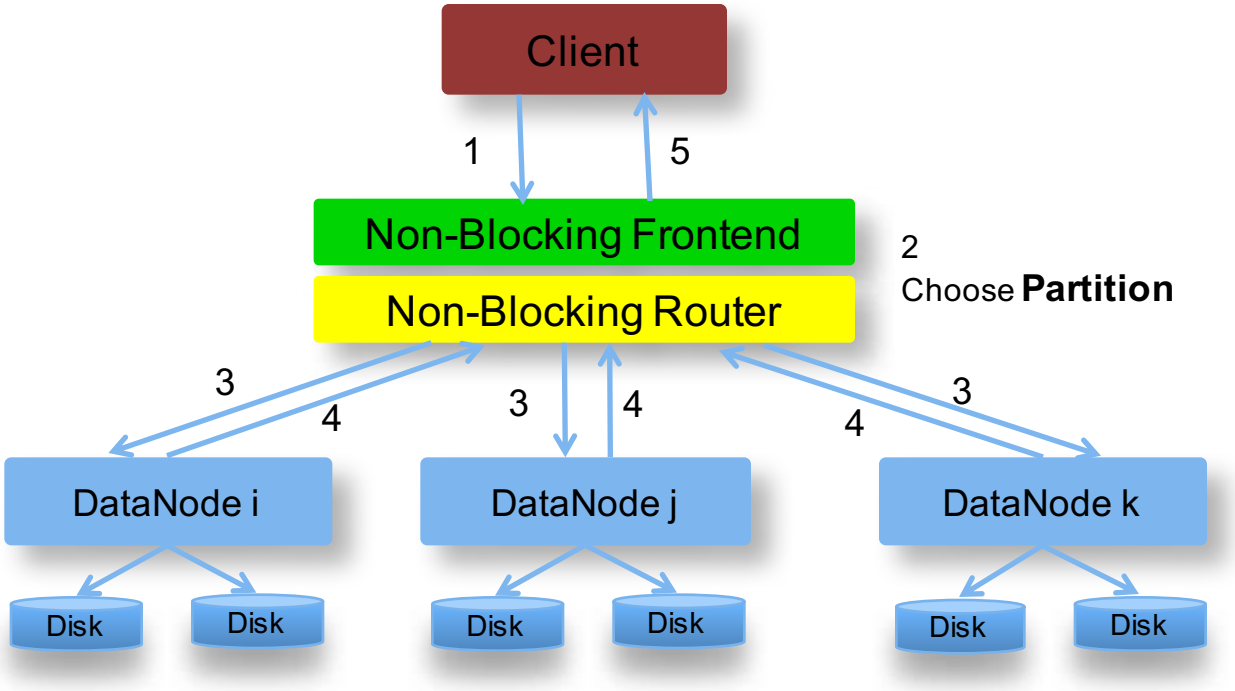
Architecture Overview



API

- **putBlob**
- **getBlobInfo**
- **getBlob**
- **deleteBlob**

Operation Flow



Static Cluster Management

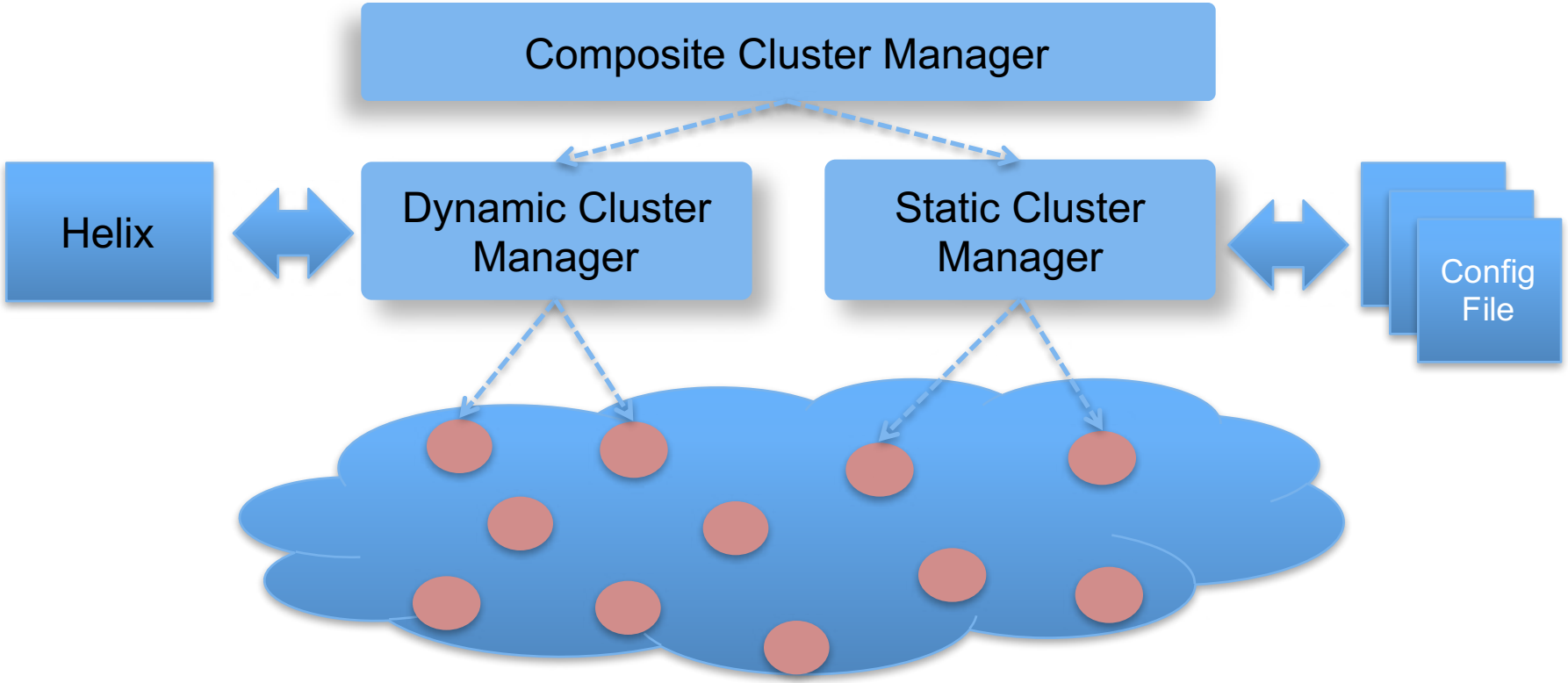
Hardware Layout

Node	Disks	Size	State
DC1: Node_1	Disk_0 Disk_1 … Disk_n	4 TB 4 TB … 4 TB	Up Up … Up
DC1: Node_2	Disk_0 Disk_1 … Disk_p	4 TB 4 TB … 4 TB	Up Down … Up
…	…	…	…
DC2: Node_i	Disk_0 Disk_1 … Disk_m	4 TB 4 TB … 4 TB	Up Up … Down

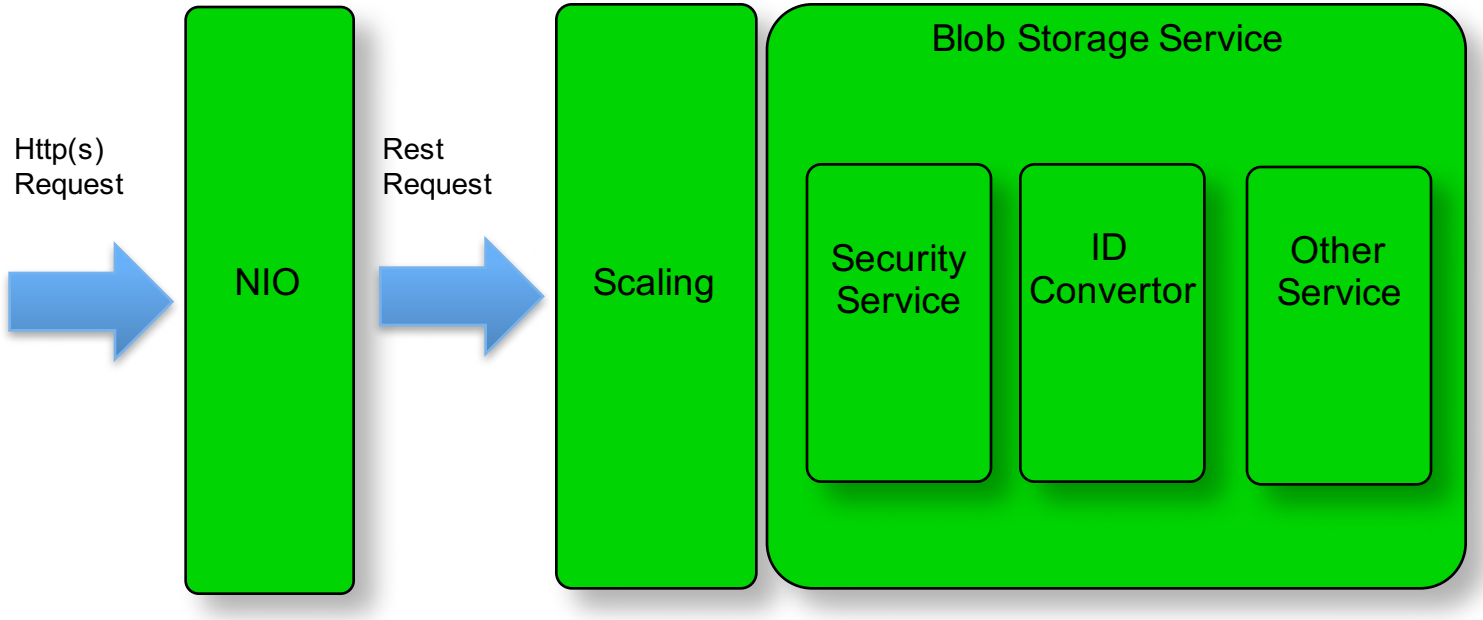
Partition Layout

Partition_id	State	Replica
Partition_0	Read-Write	DC1:Node1:Disk_0 DC1:Node4:Disk_2 … DC3:Node2:Disk_1
Partition_1	Read-Only	DC2:Node0:Disk_0 DC1:Node4:Disk_3 … DC2:Node2:Disk_2
…		…
Partition_k	Read-Write	DC1:Node1:Disk_1 DC2:Node1:Disk_2 … DC3:Node0:Disk_0

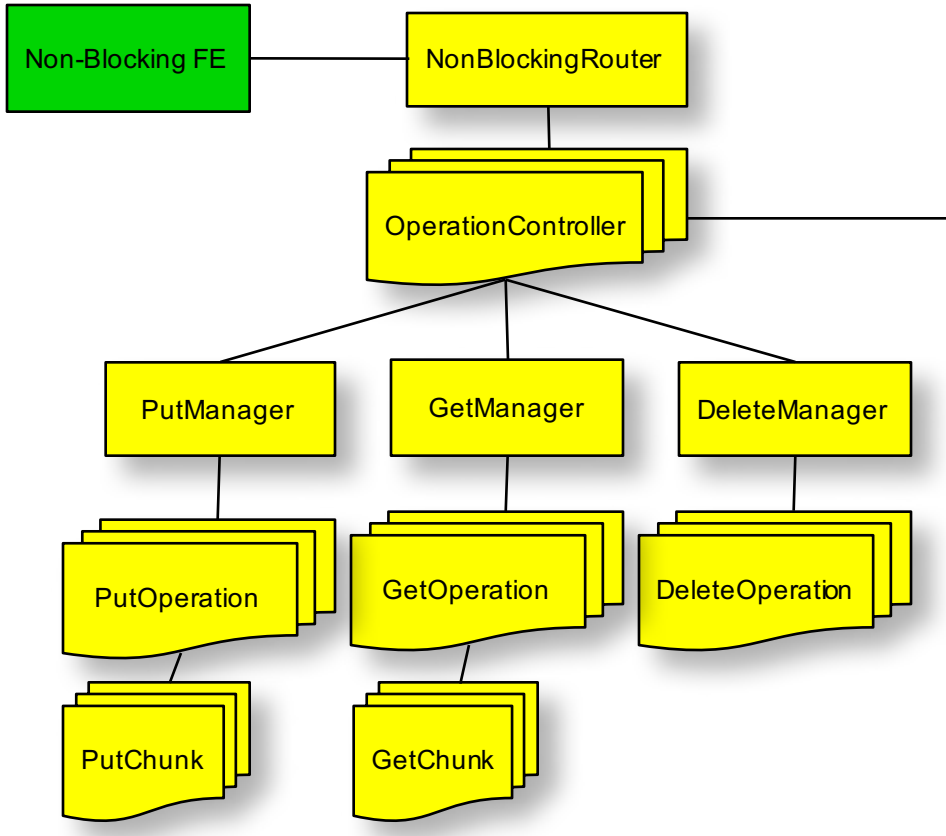
Dynamic Cluster Management



Non-Blocking Frontend

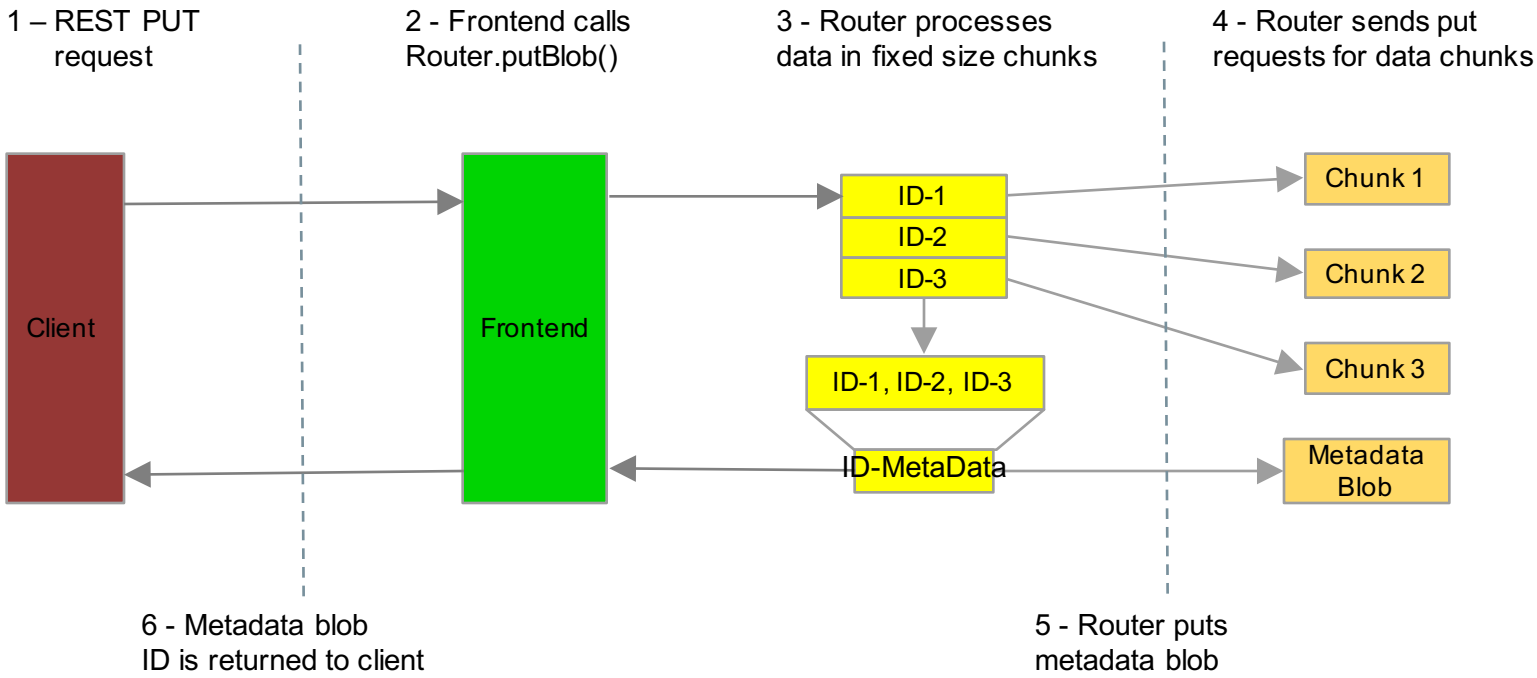


Non-Blocking Router

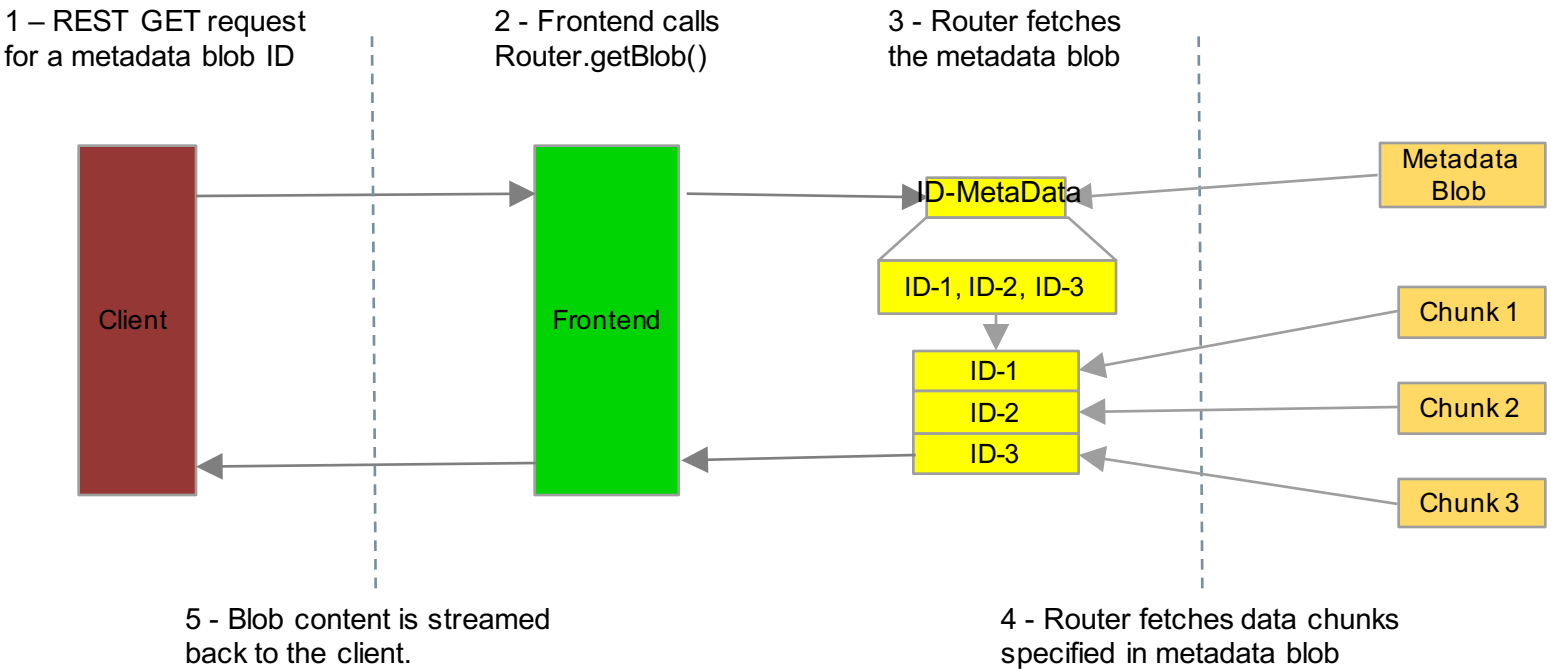



```
while (isOpen.get()) {
  List<RequestInfo> requestInfoList =
     pollForRequests();
  List<ResponseInfo> responseInfoList =
     networkClient.sendAndPoll(requestInfoList);
  onResponse(responseInfoList);
}
```

Large-Blob Support – Put Flow



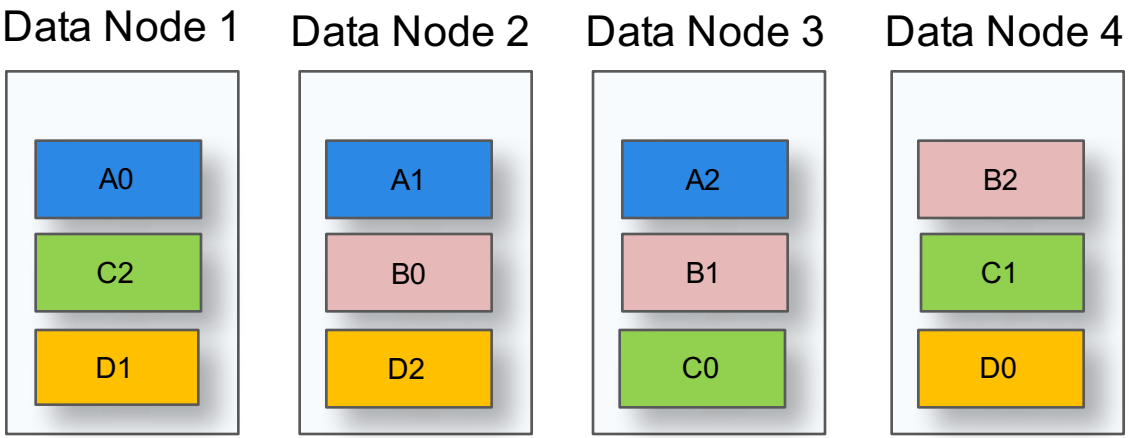
Large-Blob Support – Get Flow



Data Layer

- Data nodes with JBODs
- Support
  - Put, Get, Delete

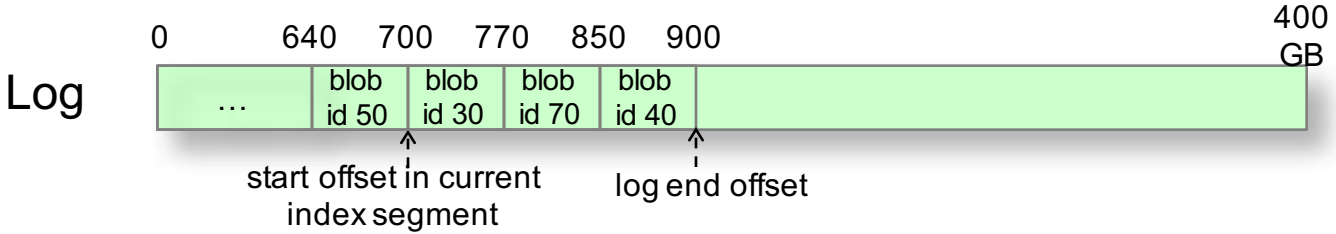
4 Data nodes
4 Partitions: A, B, C, D
3 Replicas per partition



Key Concepts in Data Layer

- Data node: the machine that hosts objects
- Partition: a *logical* container that stores objects
- Replica: the *physical* container that stores objects
  - One partition is constituted by multiple replicas.
  - Each replica eventually stores the same copy of data.
  - Replicas of the same partition may geo-distributed for parallel reads, and serve as backup.

Data Structure of Log

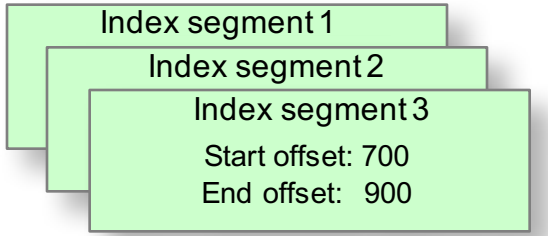


Index

blob id	offset	flags	TTL
id 30	700	-	∞
id 40	850	-	1/1/16
id 70	770	-	∞

Sorted by blob id

Index Format



Journal

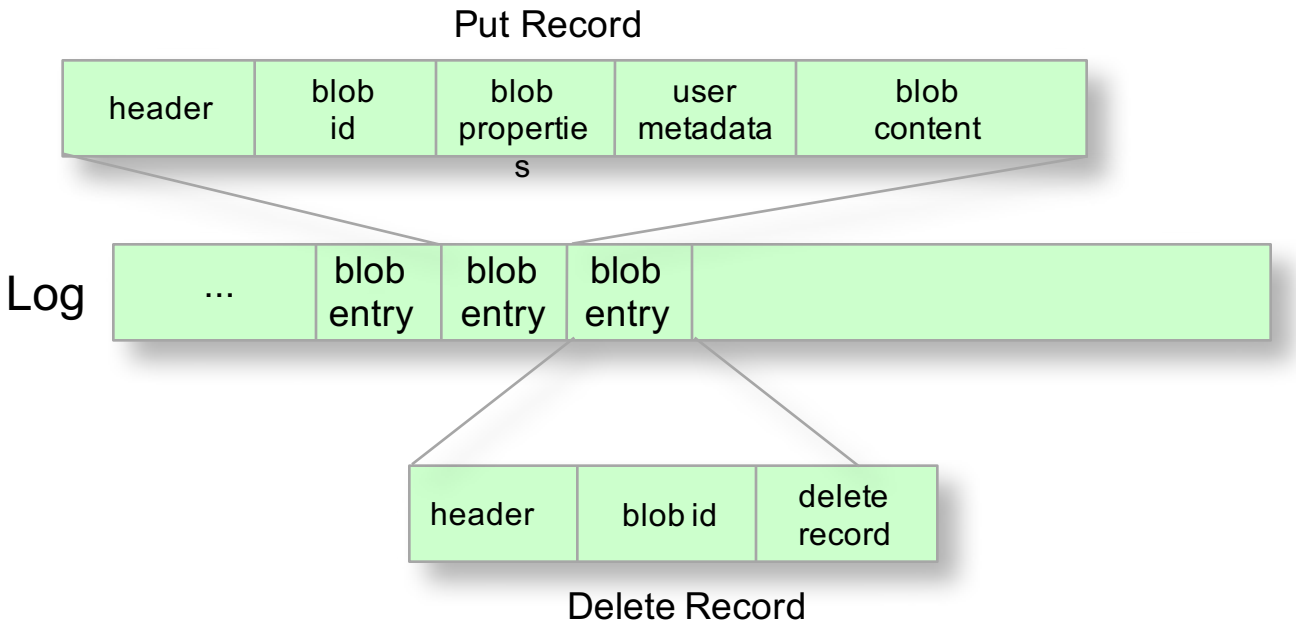
offset	blob id
640	id 50
700	id 30
770	id 70
850	id 40

Sorted by offset

Latest index segment in memory
- writable

Older index segments on disk
- read-only
- memory mapped
- bloom filter for each segment

Message Format



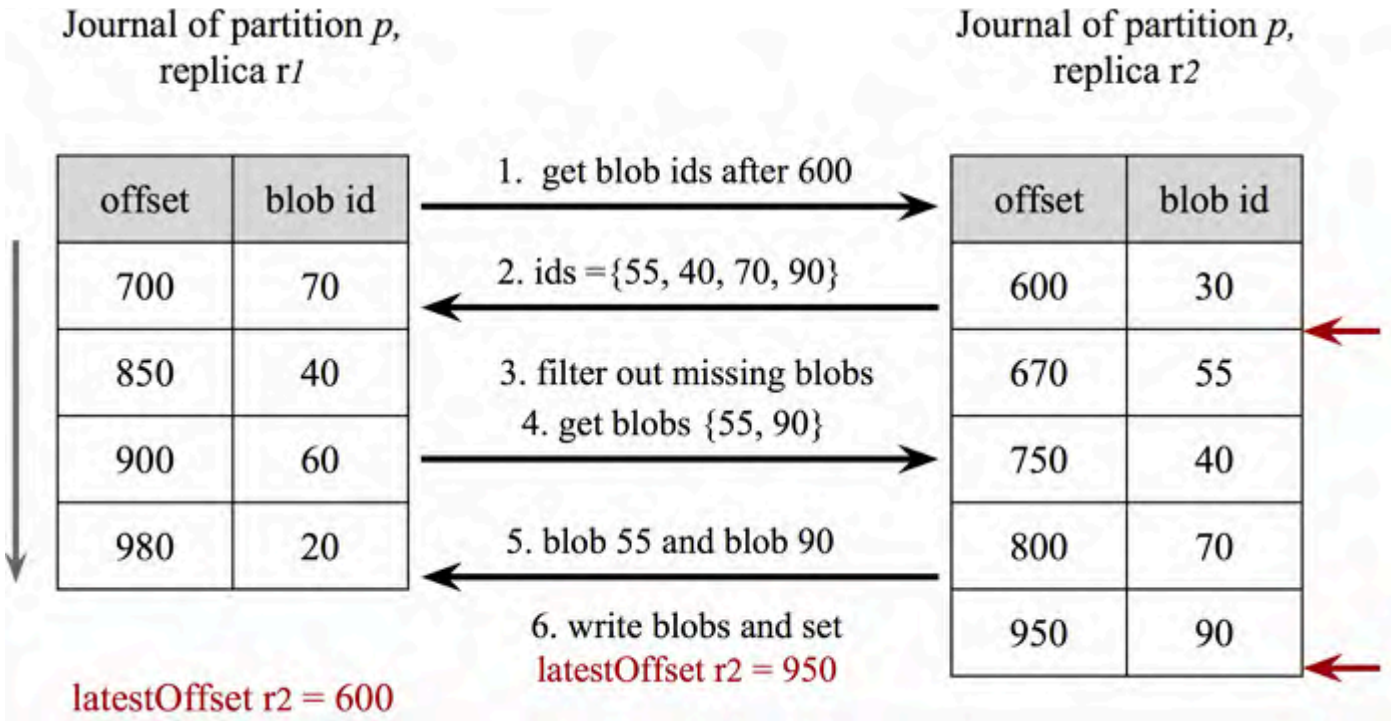
Features in Data Layer

- O(1) I/O for writes
- Bloom filter for index segments
- Rely on OS page cache
- Zero copy for gets

Data Replication

- Multi-master replication (not master-slave)
- Asynchronous (eventual consistency)
- Pull based
- Optimizations
  - Batching requests
  - Inter and intra-DC thread pools
  - Prioritization for lagging replicas
  - In-Memory Journaling
    - Map<offset, blob_id>

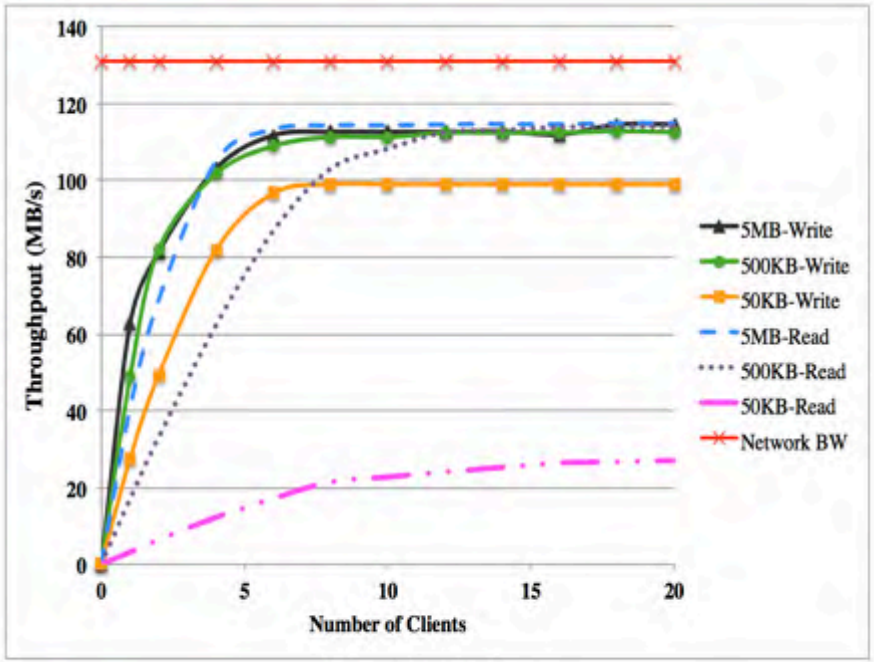
Example of Replication



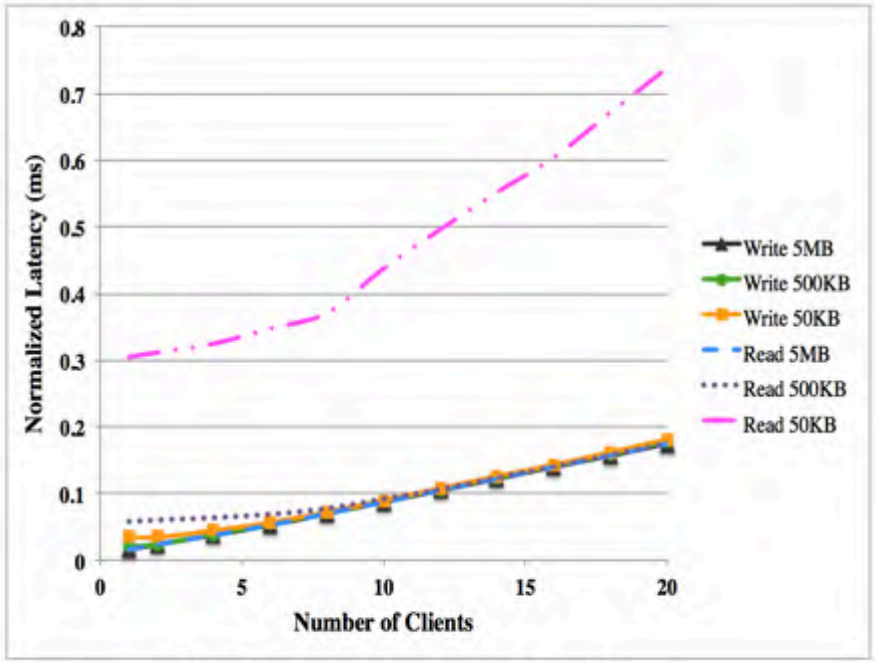
Performance Evaluation (Single-Node Setting)

# of cores	24	RAM (Ambry RAM)	64G (4G)
HDD size	1TB	# of HDD	14
Partition Size	100GB	# of partitions	8
Network BW	1Gbps		
Blob size	{25KB, 50KB, 100KB, 250KB, 500KB, 1MB, 5MB}		

Performance Evaluation (I)

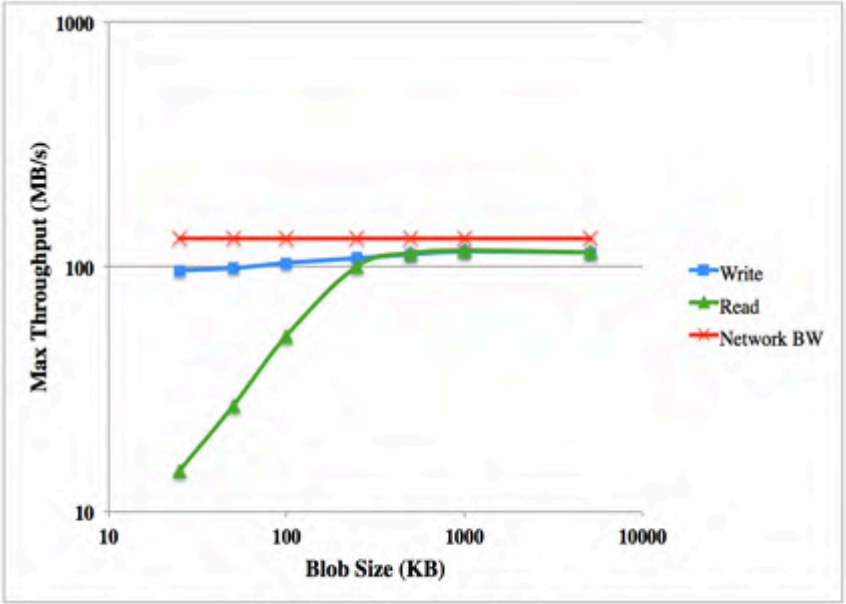


(a) Throughput

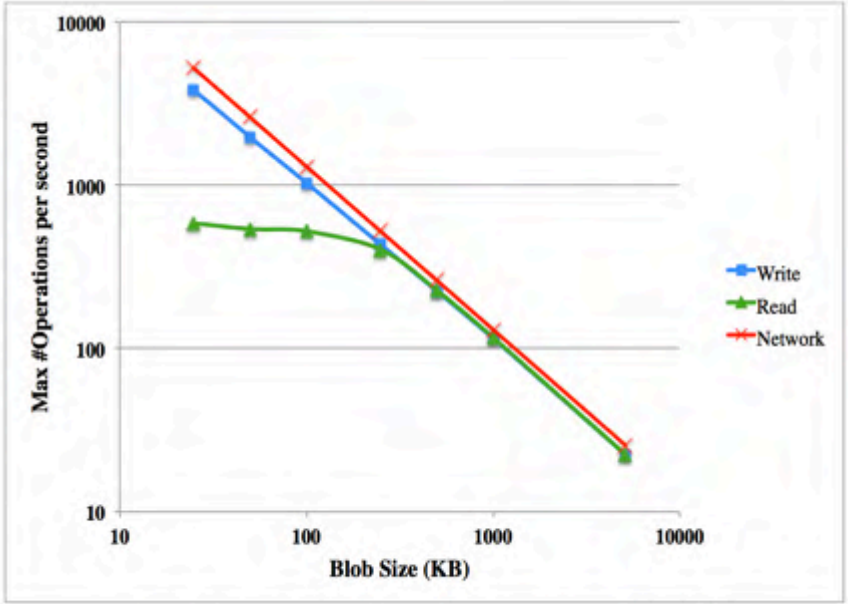


(b) Latency normalized by blob size

Performance Evaluation (II)



(a) Throughput (MB/s)



(b) Throughput (operations per second)

We have Gone Open Sourced!

- Ambry: LinkedIn's scalable geo-distributed object store, *ACM SIGMOD/PODS,* San Francisco, June 2016.
- A series of technical articles will be posted at: https://engineering.linkedin.com/blog
- **Github:** https://github.com/linkedin/ambry
  - Apache license
  - New features to be added
  - **Everyone is welcome to contribute**

New Features to Come

- Compaction
  - Automatically clean up deleted/expired blobs
- Containerization
  - Virtually grouping blobs for ACL
- Blob Stats Generator
  - Collect system-wide stats
- Async Client Library

https://github.com/linkedin/ambry