SDCC 2017深圳站
大数据技术实战峰会

CSDN

Alluxio系统的最新原理特性、案例分析与发展方向

顾 荣 博士
南京大学 计算机系 助理研究员,
Alluxio项目PMC, Maintainer
2017/06/11@SDCC 2017(深圳)

内容

- **Alluxio系统简介**
- Alluxio 1.5版本新特性介绍
- Alluxio应用场景与案例分析
- 总结

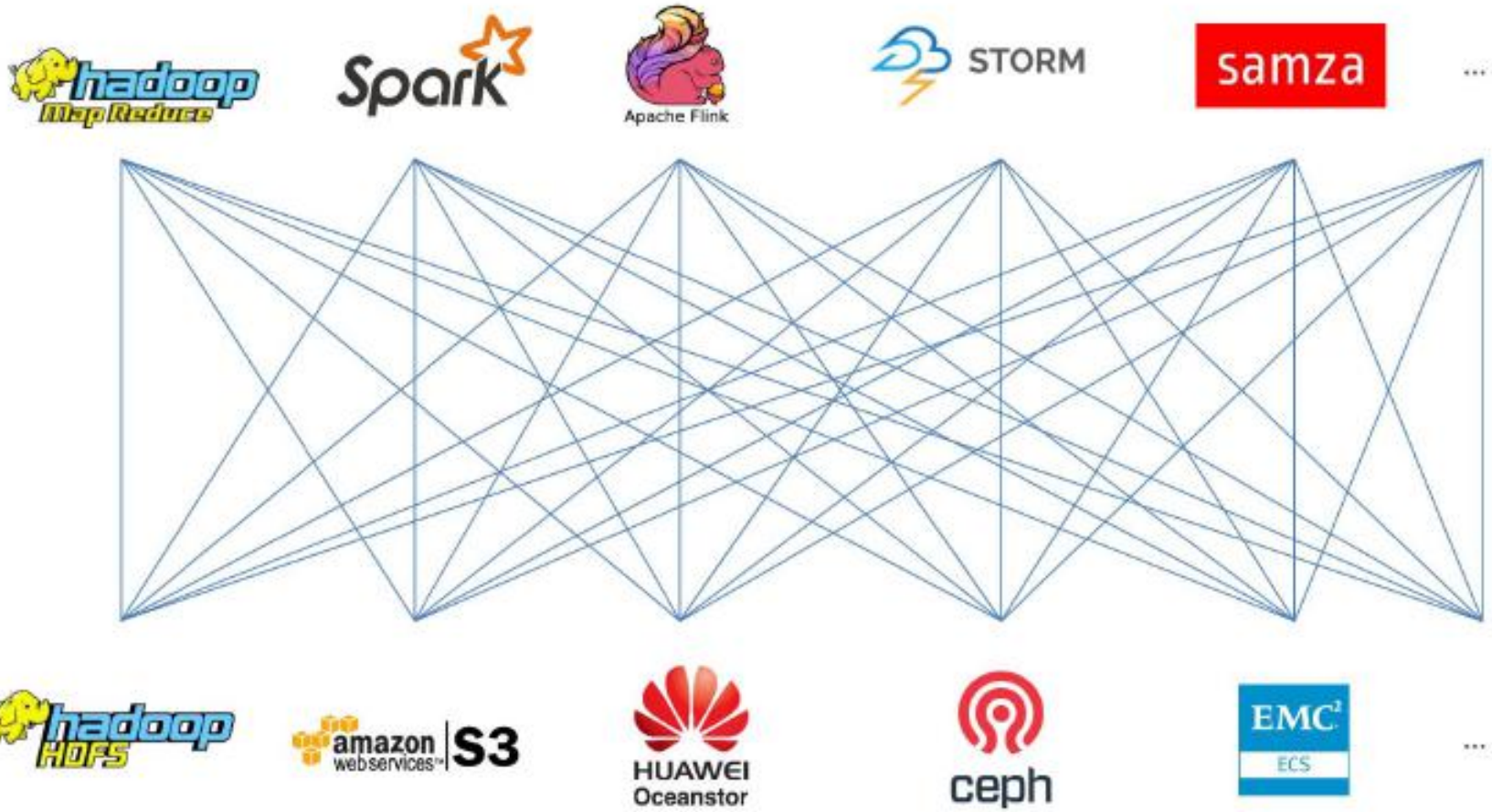
BIG DATA ECOSYSTEM Yesterday

BIG DATA ECOSYSTEM Today

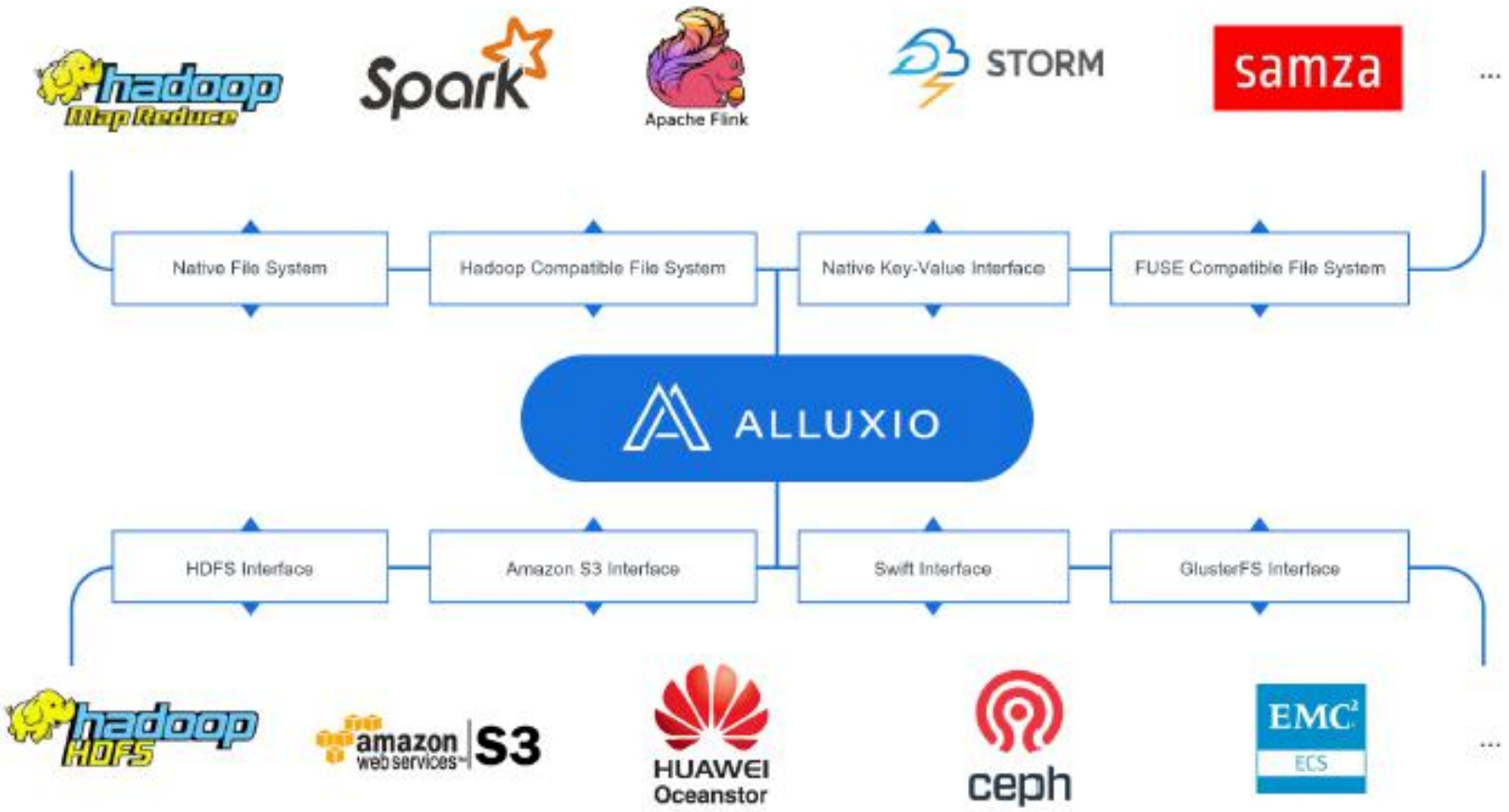
...

...

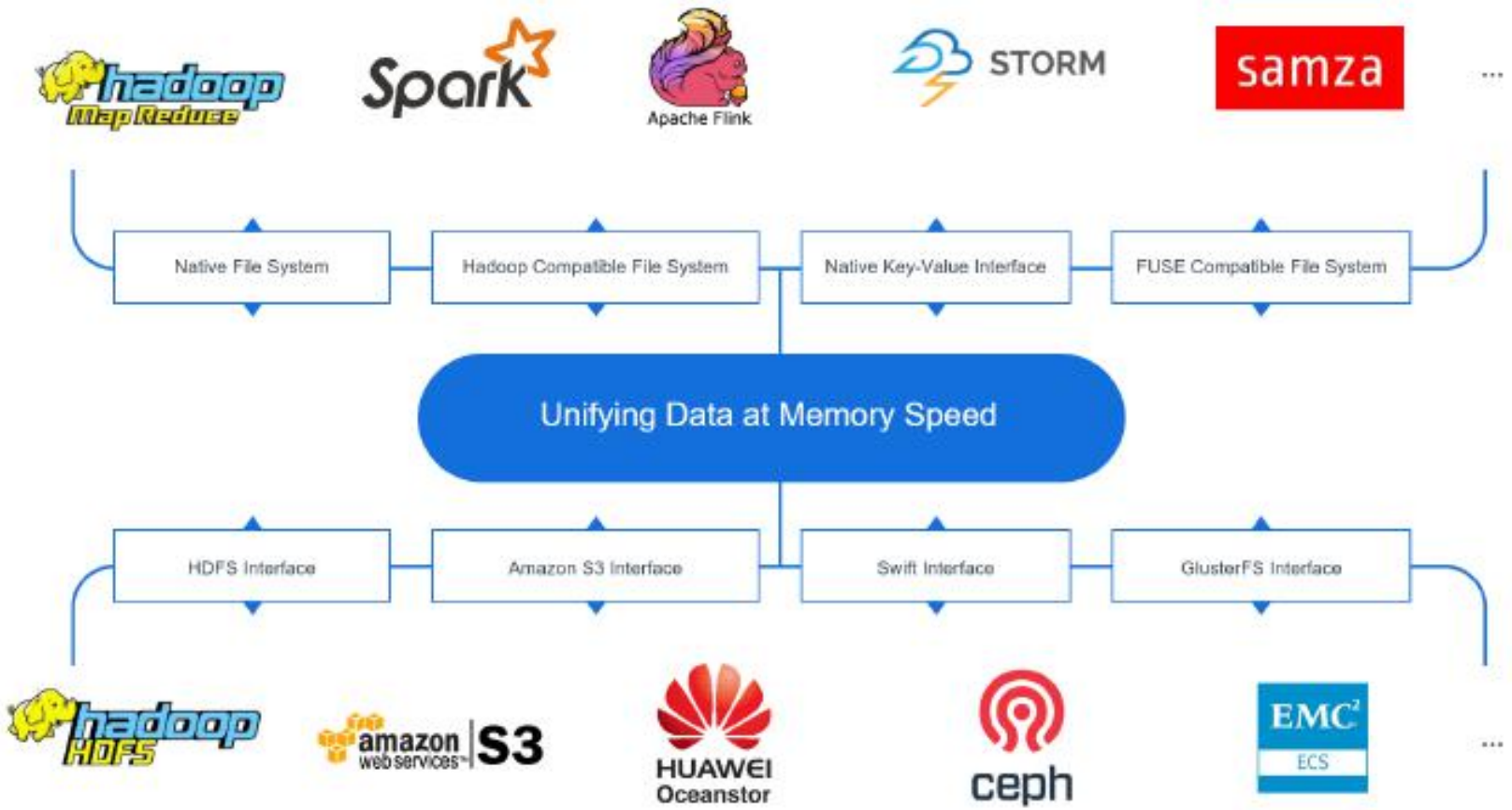
BIG DATA ECOSYSTEM Issue



BIG DATA ECOSYSTEM With Alluxio



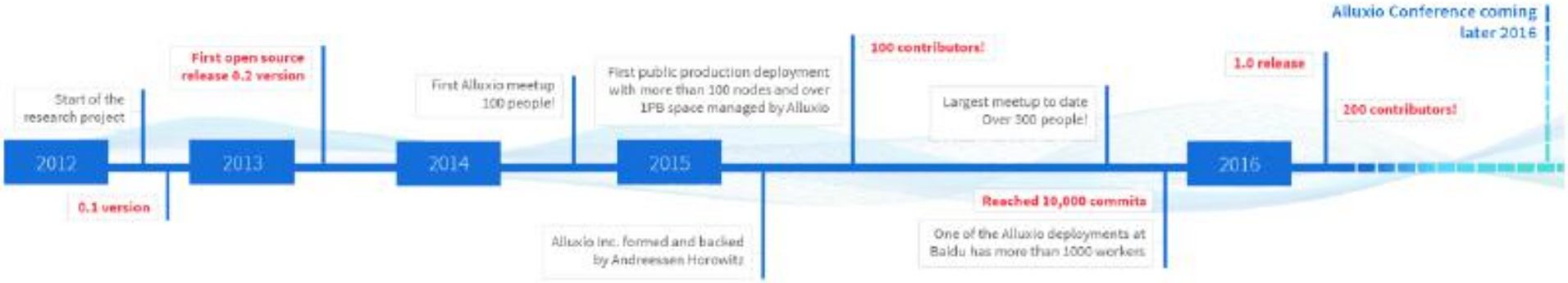
BIG DATA ECOSYSTEM With Alluxio



- Alluxio是世界上第一个以内存为中心(memory-centric)的虚拟的分布式存储系统。
- Alluxio介于计算框架和现有的存储系统之间，为大数据软件栈带来了显著的性能提升。

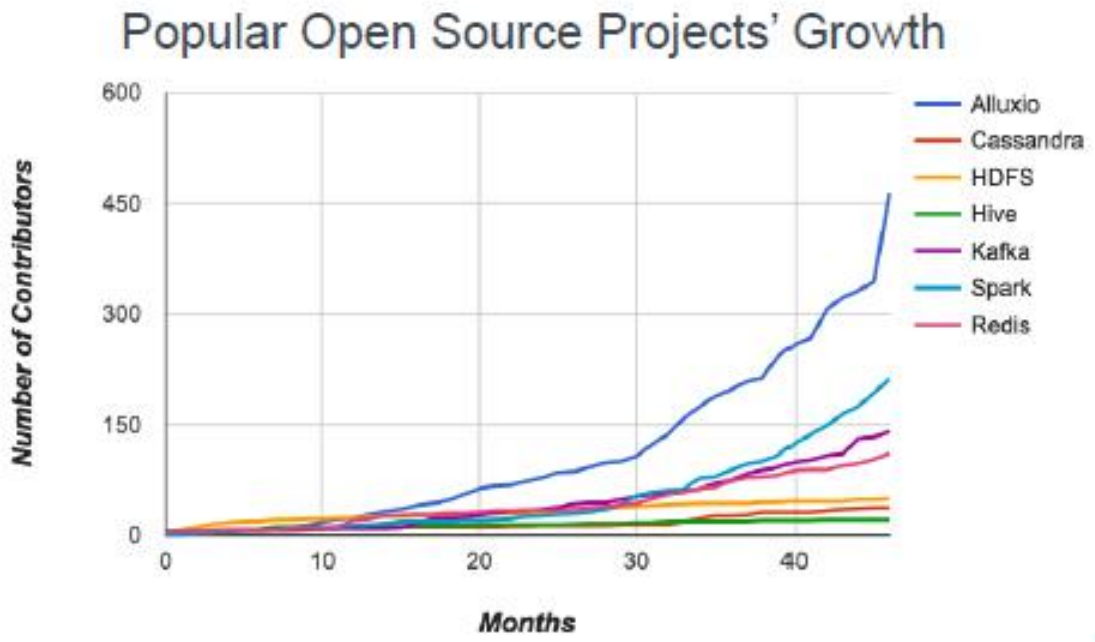
Alluxio的发展

- 2012年12月，Alluxio（Tachyon）发布了第一个版本0.1.0
- 2017年1月，Alluxio的发布1.4版本
- 2017年6月，Alluxio的发布1.5版本



Alluxio的发展

- 自2013年4月开源以来，已有超过100个组织机构的500多贡献者参与到Alluxio的开发中。包括阿里巴巴，Alluxio，百度，卡内基梅隆大学，IBM，Intel，南京大学，Red Hat，UC Berkeley和Yahoo。
- 活跃的开源社区



INDUSTRY ADOPTION

The Register®
Biting the hand that feeds IT

Data Center ▸ Storage

Multi-silo data-sucker Alluxio inks deal with Dell EMC

Follows startup's agreement with Huawei

Huawei and Alluxio Jointly Release Big Data Storage Acceleration Solution, Speeding Up Big Data Application Popularization

DATA + FOLLOW THIS TOPIC

8 data trends on our radar for 2017

From deep learning to decoupling, here are the data trends to watch in the year ahead.

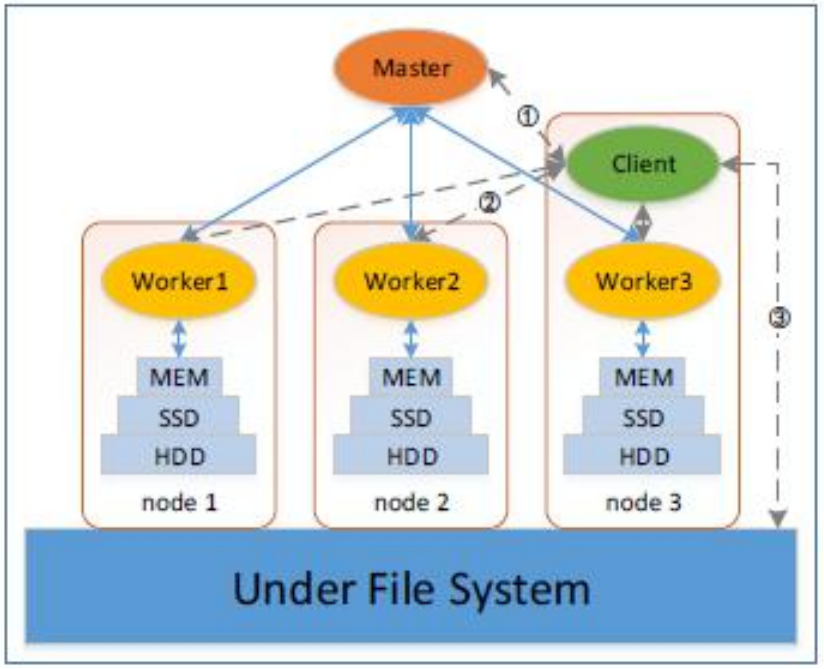
By Ben Lorica, January 3, 2017

6. The decoupling of storage and computation will accelerate.

The UC Berkeley AMPLab project ended last November, but the team behind Apache Spark and Alluxio are far from the only ones to highlight the separation of storage and computation. As noted above, popular object stores in the cloud and even some recent deep learning architectures emphasize this pattern.

Alluxio整体架构

- Master-Worker
  - Master
    - 管理全部元数据
    - 监控各个Worker状态
  - Worker
    - 管理本地MEM、SSD和HDD
- Client
  - 向用户和应用提供访问接口
  - 向Master和Worker发送请求
- Under File System
  - 用于备份



内容

- Alluxio系统简介
- **Alluxio 1.5版本新特性介绍**
- Alluxio应用场景与案例分析
- 总结

Ecosystem Integrations

- Golang Client
  - A native client has been developed to enable Golang applications to communicate with Alluxio. This allows applications in Golang to interact with Alluxio without needing to write custom code to invoke Alluxio's REST interface or Java client.
- Presto Integration
  - Support for running Presto on top of Alluxio. Alluxio can provide up to 10x faster queries for Presto.
- Docker Integration
  - Support for deploying various Alluxio services in a Docker container.

Ecosystem Integrations

- Azure Blob Store Integration
  - Support for running Alluxio with Azure Blob Store as an under storage.
- Alternative Ceph Connector
  - In 1.5.0, Alluxio can connect to Ceph under storages using the S3A connector which provides significant functionality and performance improvements. Users can expect up to 3x read performance improvement.
- Improved Mesos Integration
  - Improved support for deploying Alluxio in a Mesos and Marathon environment. This includes convenience functionality for deploying Alluxio servers as well as configuring Alluxio clients.

Performance Improvements

- Periodic Journal Checkpointing
  - Alluxio periodically compacts journal edit logs to update the journal checkpoint. This reduces restart and failover (if in HA mode) times, which were previously unbounded, to less than a minute. Periodic journal checkpointing is enabled in both HA and non HA modes.

- Packet Streaming Based I/O
  - All network I/O now use the packet streaming protocol, providing up to 3x performance improvement. In addition, the metadata management between client and worker is greatly simplified, reducing the number of connections between the two components by about half.

Performance Improvements

- Domain Socket Based Short Circuit I/O
  - Alluxio 1.5.0 introduces an option to use domain sockets for short circuit I/O to write data to local Alluxio storage through a worker process instead of the client. This mode is recommended for users running write heavy workloads in a container environment.
- Optimized Alluxio to Under Storage Metadata Operations
  - The internal logic and concurrency mechanism of Alluxio is greatly improved for handling metadata heavy workloads with under storage systems, most notably object storages like Amazon S3, resulting in up to 10x performance improvement in metadata heavy stages of compute tasks.

Usability Improvements

- High and Low Watermark Space Reserver
  - Asynchronous space reservation on Alluxio workers are now triggered at a specified high watermark and evict until the user specified low watermark is reached. It is highly recommended to use the space reserver in workloads with bursty writes. See the documentation for more details.

- Improved Object Store Directory Management
  - The need to use dummy metadata files suffixed by `$_folder_$` has been removed. This enables Alluxio to be more compatible with other systems interacting with the backing object store and allows users to mount buckets without having write permissions or the bucket set up with dummy files to work with Alluxio.

Usability Improvements

- Alluxio Client Module Isolation
  - Alluxio 1.5.0 introduces a runtime module specifically designed for drop-in use of an Alluxio client. This client is equivalent to the core-client uber jar in previous versions. Alluxio's other client interfaces such as HDFS and Alluxio's native file system are in independent modules making it much easier to control dependencies for downstream projects.

- Mount Point Configuration Properties
  - Richer configuration settings are exposed at a mount point granularity, enabling users to mount under storage systems with different properties. This enables use cases like using different credentials to connect to the same type of under storage. For example, mounting in Alluxio two Google Cloud Store buckets which require different credentials.

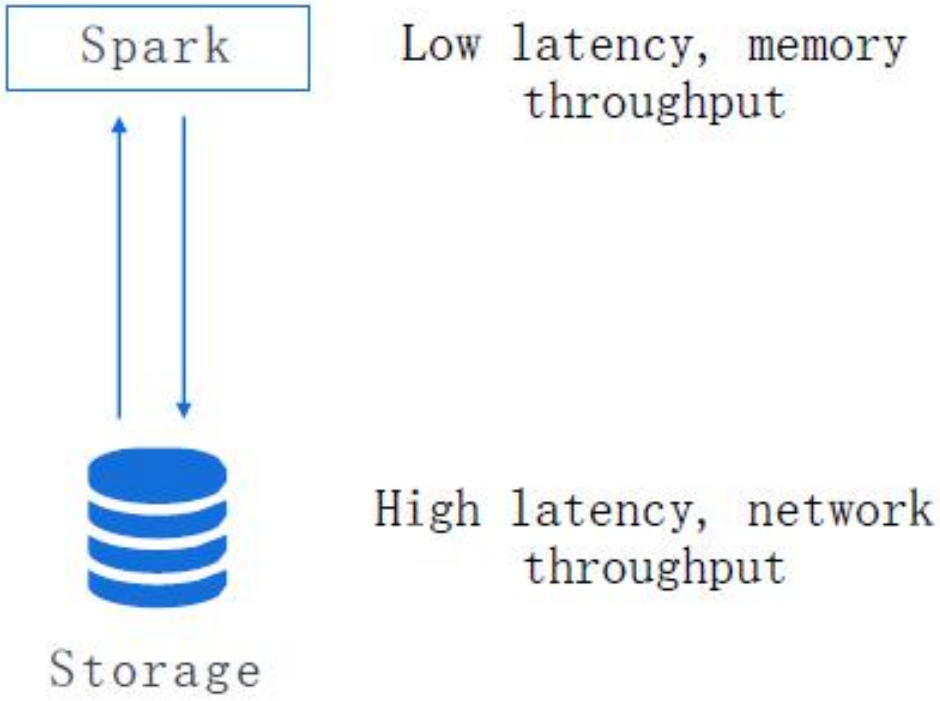
内容

- Alluxio系统简介
- Alluxio 1.5版本新特性介绍
- **Alluxio应用场景与案例分析**
- 总结

应用场景1：加速远程存储的I/O访问

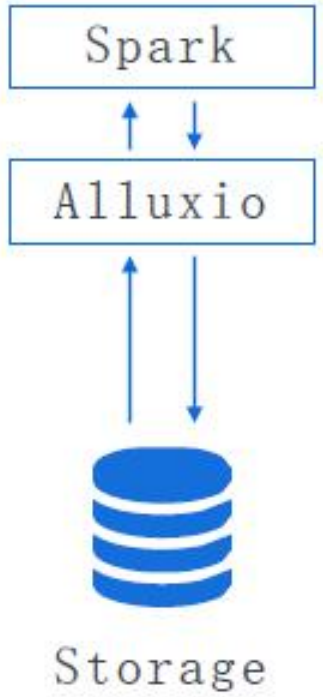
- 场景: 计算和存储分离
  - 可以满足不同的计算和存储硬件要求
  - 能够独立地扩展计算层和存储层
  - 将数据存储在传统的文件系统/SAN和对象存储中
  - 通过大数据计算框架分析现有数据
- 限制：
  - 访问数据需要远程I/O

不使用ALLUXIO访问远程存储



使用ALLUXIO访问远程存储

将数据放在Alluxio中以
加速数据访问



实际案例：百度

Baidu的项目经理和分析师需要运行交互式的查询以获得关于他们产品和商业的insights

- 200+ nodes deployment
- 2+ petabytes of storage
- Mix of memory + HDD

Baidu 百度

> *The performance was amazing. With Spark SQL alone, it took 100-150 seconds to finish a query; using Alluxio, where data may hit local or remote Alluxio nodes, it took 10-15 seconds.*
>
> - Baidu

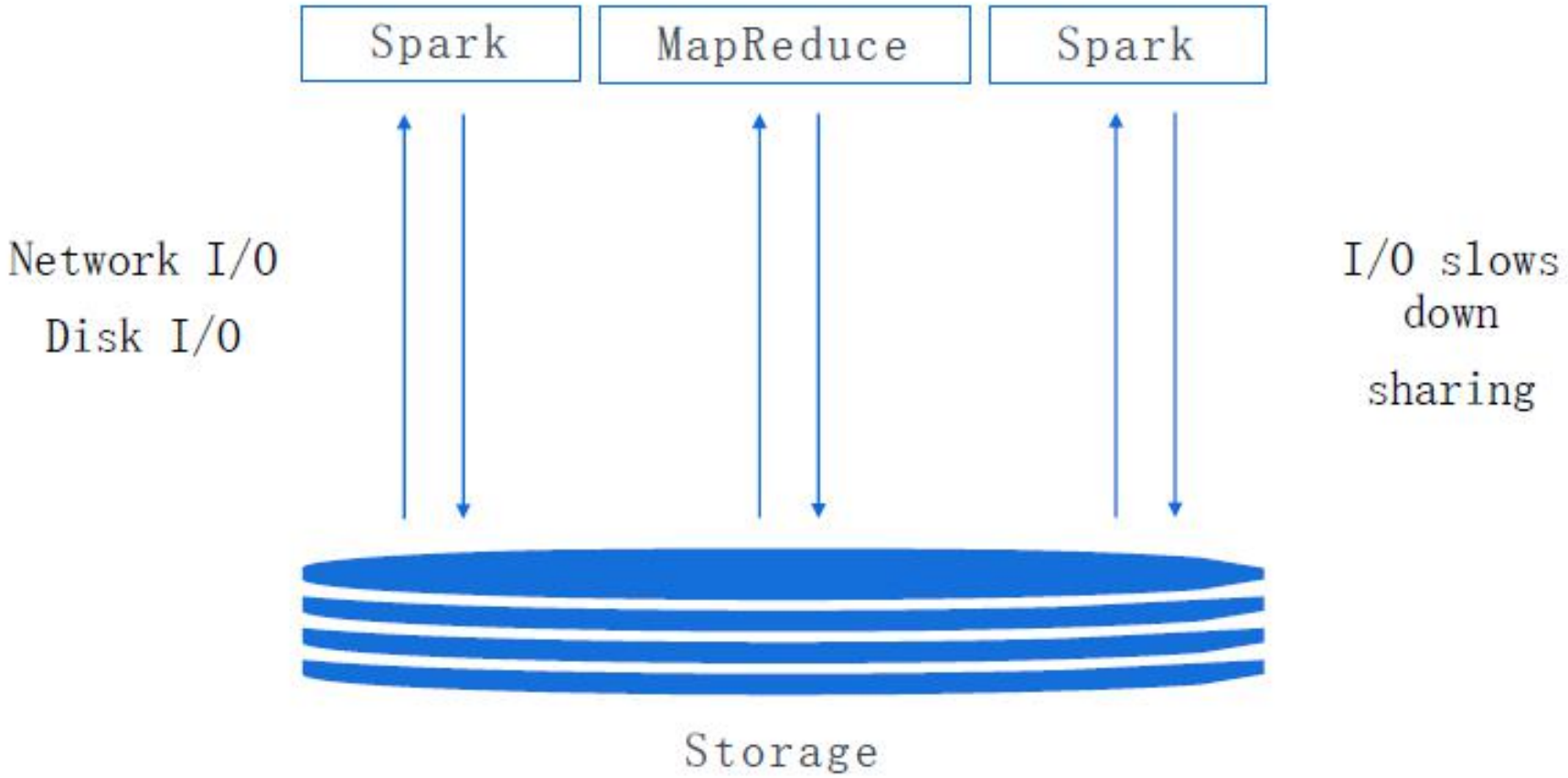
结果:

- 通过使用Alluxio，性能加速了30x
- Alluxio集群运行稳定，并管理着50TB的集群内存空间
- 通过使用Alluxio, 原先需要15分钟才能执行完的批处理查询，转换为了可以30秒内可以处理完的交互式查询

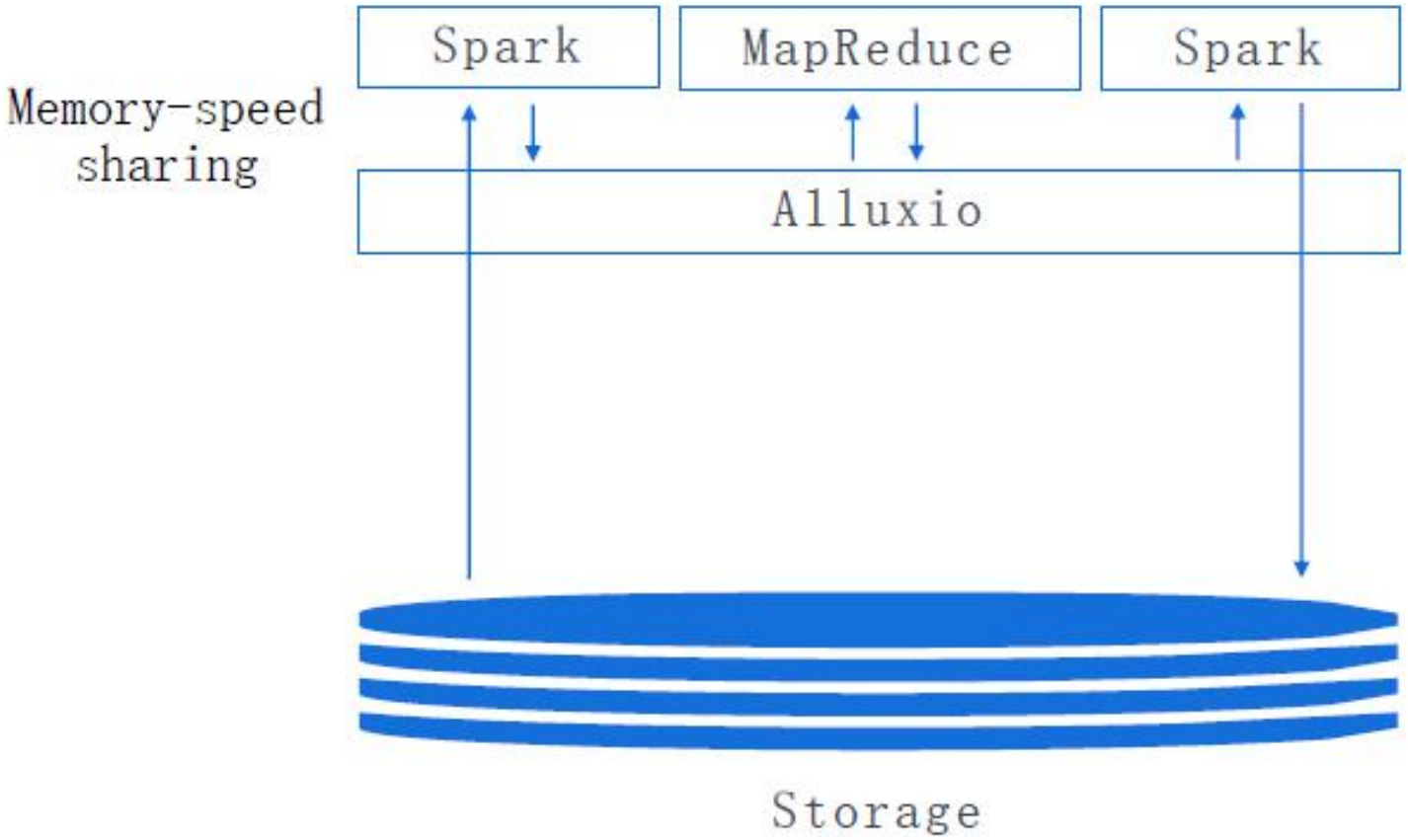
应用场景2：不同应用以内存速度共享数据

- 场景: 数据共享式架构
  - 流水线: 一个作业的输出是下一个作业的输入
  - 应用、作业、上下文（ contexts ）读取相同的数据
- 限制：
  - 共享数据需要I/O

不使用ALLUXIO进行数据共享



使用ALLUXIO进行数据共享



实际案例：巴克莱银行（BARCLAYS）

巴克莱银行通过查询抽取数据，并运行机器学习算法来训练风控模型

- 6 node deployment
- 1TB of storage
- Memory only

BARCLAYS

> *Thanks to Alluxio, we now have the raw data immediately available at every iteration and we can skip the costs of loading in terms of time waiting, network traffic, and RDBMS activity.*
>
> - Barclays

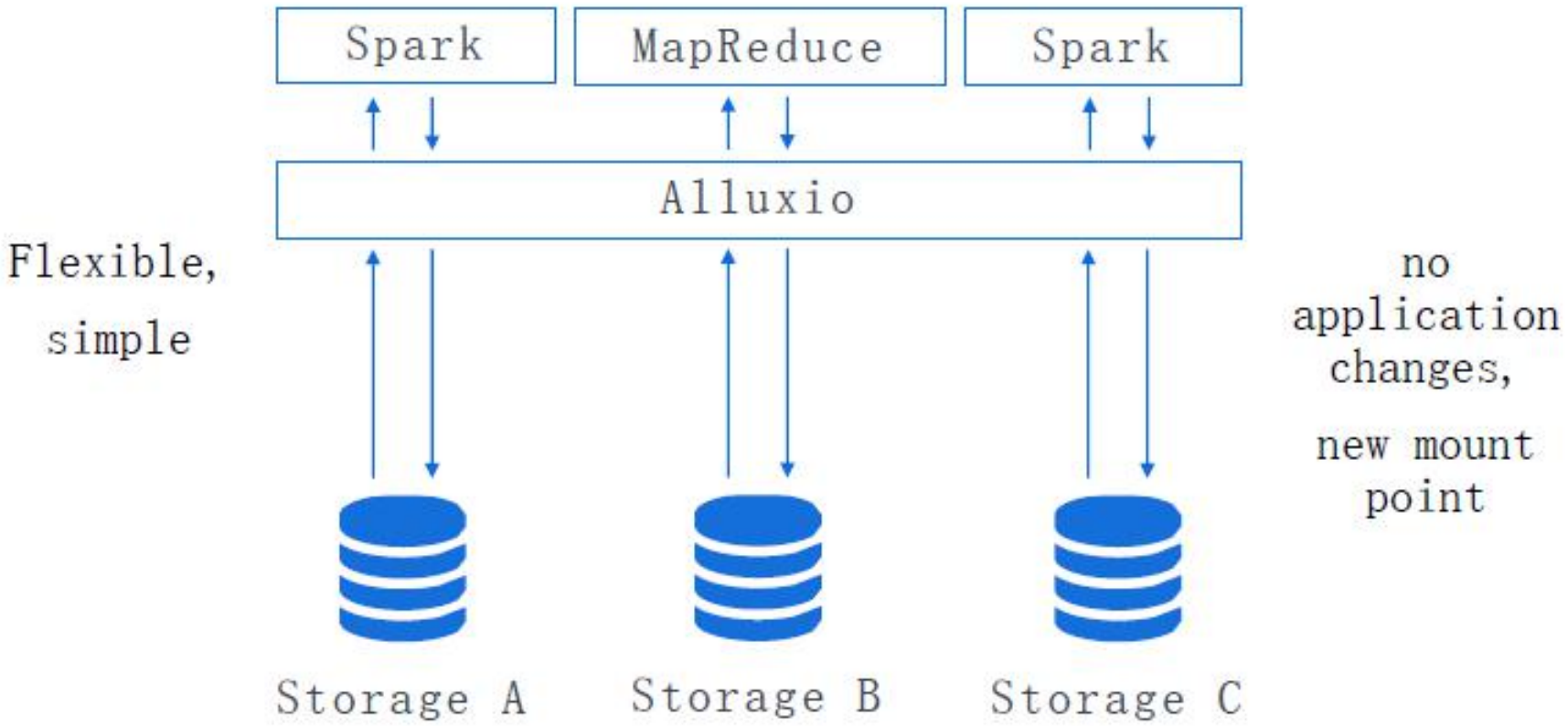
结果

- 巴克莱银行的处理流程的每轮的迭代时间从几小时降低至几秒钟
- Alluxio使得之前在给定时间内很难运行的工作变得可能
- 通过将数据缓存在Alluxio内存中，加载和存储的I/O开销变成只有几秒钟

应用场景3：统一不同存储的数据访问

- Scenario: Multiple Storage Systems
  - Most enterprises have multiple storage systems
  - New (better, faster, cheaper) storage systems arise
- Limitation
  - Accessing data from different systems requires different APIs

使用ALLUXIO访问数据



实际案例：去哪儿网

去哪儿网使用实时机器学习算法服务于他们的线上广告

- 200+ nodes deployment
- 6 billion logs (4.5 TB) daily
- Mix of Memory + HDD

去哪儿? Qunar.Com 即 明 你 的 旅 行

> *We've been running Alluxio in production for over 9 months, Alluxio's unified namespace enable different applications and frameworks to easily interact with data from different storage systems.*
>
> - Qunar

结果

- 通过使用Alluxio，在Spark流处理、Spark批处理以及Flink作业直接的数据共享变得高效
- 系统提升了大约15 – 300倍
- 采用的Alluxio层次化存储功能，对集群的存储资源（内存、HDD）进行了分层管理

内容

- Alluxio系统简介
- Alluxio 1.5版本新特性介绍
- Alluxio应用场景与案例分析
- 总结

技术特点小结

- 将计算与数据共同安置（co-located），提供内存级访问速度
- 通过统一的命名空间虚拟化底层不同存储系统
- 横向可扩展的系统架构
- 文件系统API（软件层的解决方案）

系统优势小结

Unification

新的应用可以访问任意存储里的任意数据

Performance

多个数量级以上的性能提升

Flexibility

计算和存储的选择变得独立，可以根据需要而进行购买

欢迎使用Alluxio中文文档！

- http://alluxio.org/documentation/master/cn/index.html

SDCC 2017深圳站
大数据技术实战峰会

CSDN

个人微博：顾荣_NJU
电子邮箱：gurong@nju.edu.cn