

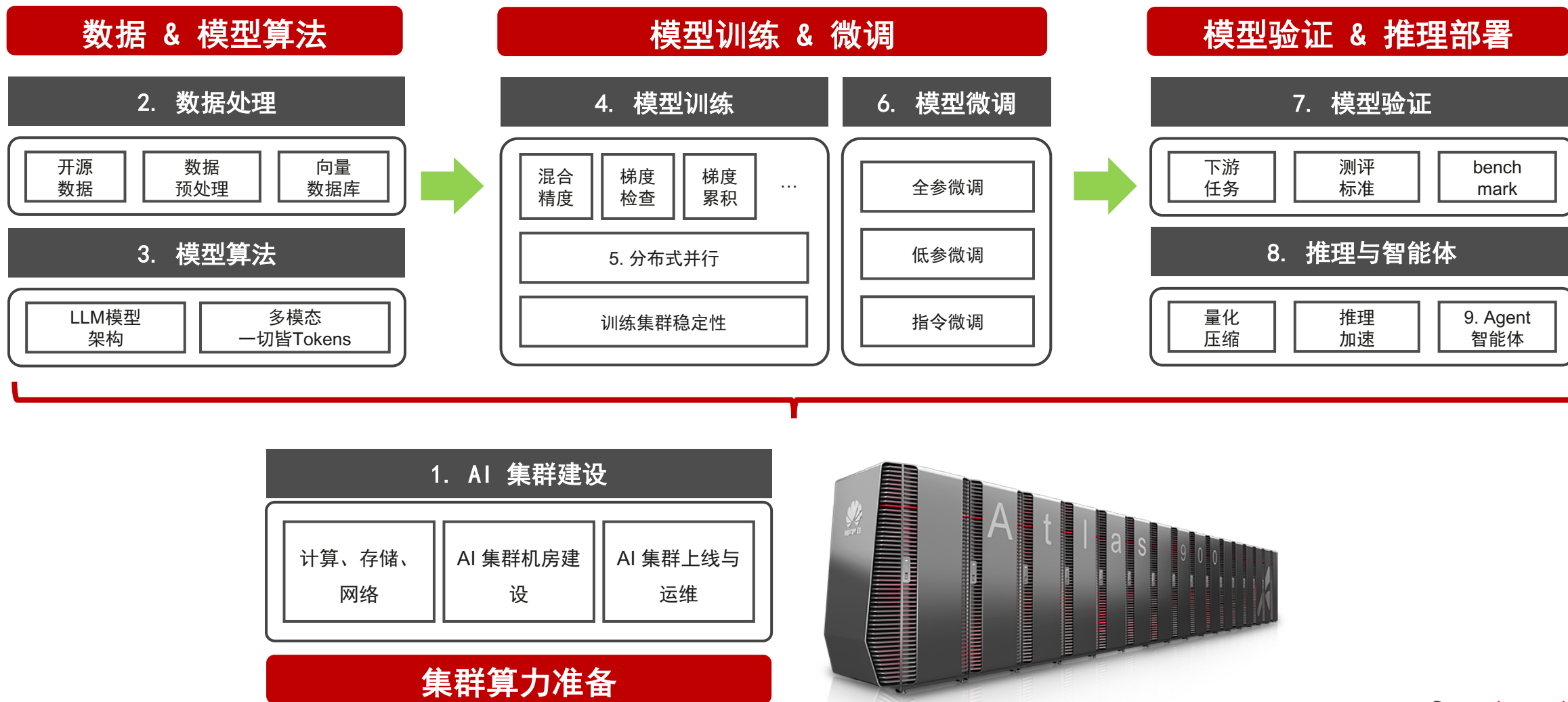
大模型-AI集群 (存)

# 数据存储 对象 & 应用



ZOMI

# 大模型业务全流程



# 关于本内容

## • 内容背景

- AI 集群 + 大模型

## • 具体内容

- **数据存储现状和场景**：存储软件类型、存储硬件类型的发展
- **大模型对存储的挑战**：存储性能指标、存储遇到大模型挑战与新机会点
- **大模型训练CKPT优化**：大模型训练过程、CKPT过程分解、CKPT优化
- **大模型时代对存储的思考**：什么样的存储架构才是AI大模型时代的选择？

# 增长的存储主要靠数据中心和存储服务器拉动

## 存储 技术 架构

### 大模型&应用

LLAMA等LLM

多模态大模型

AI Agent智能体

### 大模型训练

多级存储

CheckPoint

训练数据

### 存储对象 存储协议

文件存储

NFS

HDFS

块存储

NVMe

SCSI

对象存储

Tensor

Vector

其他

Table

### 连接方式

SAN

NAS

DAS

### 存储架构

集中式存储

分布式存储

### 硬件介质

磁带

光盘

机械硬盘

固态硬盘

# 1. 数据存储 の类型

# 存储类型

- **存储的数据格式**：块存储、文件存储、对象存储。
- **块存储**：可获取任何数据（文件或数据库item），将其分为大小相同的数据块；
- **文件存储**：以文件 and/or 文件夹的层次结构来规整和呈现数据；
- **对象存储**：非结构化格式 and/or 对象的单位存储和管理数据；

# 存储类型与 AI 的关系

- **存储的数据格式**：块存储、文件存储、对象存储。
- **块存储**：可获取任何数据（文件或数据库item），将其分为大小相同的数据块；
  - —— AI 大模型中会以 bin 或者 拆分后的 bin 块把模型权重参数存储起来
- **文件存储**：以文件 and/or 文件夹的层次结构来组织和呈现数据；
- **对象存储**：非结构化格式 and/or 对象的单位存储和管理数据；

# 存储类型与 AI 的关系

 **Hugging Face**

[Models](#) [Datasets](#) [Spaces](#) [Docs](#) [Solutions](#) [Pricing](#) [Log In](#) [Sign Up](#)

 **THUDM/glm-10b**   like 32

[Feature Extraction](#) [Transformers](#) [PyTorch](#) [English](#) [glm](#) [thudm](#) [custom\\_code](#) [arxiv:2103.10360](#)

[Model card](#) [Files and versions](#) [Community](#) [Train](#) [Use in Transformers](#)

[main](#) [glm-10b](#) [1 contributor](#) [History: 5 commits](#)

|                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <b>zxdu20</b> Fix batch beam search <a href="#">696788d</a> 9 months ago                                                                                                                                                                                                                                                            |
|  <a href="#">.gitattributes</a> 1.48 kB  initial commit 9 months ago                                                                                                                                                                              |
|  <a href="#">README.md</a> 2.25 kB  add tag 9 months ago                                                                                                                                                                                          |
|  <a href="#">added_tokens.json</a> 174 Bytes  init commit 9 months ago                                                                                                                                                                            |
|  <a href="#">config.json</a> 955 Bytes  update path 9 months ago                                                                                                                                                                                  |
|  <a href="#">configuration_glm.py</a> 6.25 kB  init commit 9 months ago                                                                                                                                                                       |
|  <a href="#">merges.txt</a> 456 kB  init commit 9 months ago                                                                                                                                                                                  |
|  <a href="#">modeling_glm.py</a> 39.8 kB  Fix batch beam search 9 months ago                                                                                                                                                                  |
|  <a href="#">pytorch_model.bin</a>  19.8 GB   init commit 9 months ago |
|  <a href="#">tokenization_glm.py</a> 15.9 kB  init commit 9 months ago                                                                                                                                                                        |
|  <a href="#">tokenizer_config.json</a> 466 Bytes  update path 9 months ago                                                                                                                                                                    |



# 存储类型与 AI 的关系

- **存储的数据格式**：块存储、文件存储、对象存储。
- **块存储**：可获取任何数据（文件或数据库item），将其分为大小相同的数据块；
- **文件存储**：以文件 and/or 文件夹的层次结构来组织和呈现数据；
  - —— 已经 Label 好的标签数据和训练数据，通过文件夹来管理，提供基础的封装格式。
- **对象存储**：非结构化格式 and/or 对象的单位存储和管理数据；

# 存储类型与 AI 的关系

- **存储的数据格式**：块存储、文件存储、对象存储。
- **块存储**：可获取任何数据（文件或数据库item），将其分为大小相同的数据块；
- **文件存储**：以文件 and/or 文件夹的层次结构来组织和呈现数据；
- **对象存储**：非结构化格式 and/or 对象的单位存储和管理数据；
  - —— 训练数据以对象存储为主，把图像、视频、语料以张量作为对象给AI框架输入

# 存储类型对比

|       | 块存储                             | 文件存储                                  | 对象存储                   |
|-------|---------------------------------|---------------------------------------|------------------------|
| 操作对象  | 块（卷 and/or 硬盘），按 Byte 访问        | 文件和文件夹                                | 规定数据结构的数据单元作为对象        |
| 存储结构  | 非连续存储的结构化固定 Block               | 层级结构                                  | 地址空间                   |
| 数据结构  | 数组                              | 二叉树                                   | 哈希表                    |
| 文件管理  | 存储到硬盘的文件为主，需要额外的资源和管理来支持块存储上的文件 | 支持常见的文件协议和权限模式，可提供配置为使用共享文件存储的一个用程序使用 | 将视频、音频、图像、网络元数据文件存储为对象 |
| 元数据管理 | 使用的关联元数据很少                      | 仅存储与文件相关的有限元数据                        | 可以存储任何对象的无限元数据，自定义字段等  |
| 存储性能  | 高性能、低延迟和快速的数据传输                 | 为共享文件访问提供高性能                          | 存储海量不规则数据              |
| 物理存储  | SSD、HDD                         | 本地NAS服务器或者SSD/HDD                     | 分布式存储节点                |

# 存储类型对比

|       | 块存储                             | 文件存储                                  | 对象存储                   |
|-------|---------------------------------|---------------------------------------|------------------------|
| 操作对象  | 块（卷 and/or 硬盘），按 Byte 访问        | 文件和文件夹                                | 规定数据结构的数据单元作为对象        |
| 存储结构  | 非连续存储的结构化固定 Block               | 层级结构                                  | 地址空间                   |
| 数据结构  | 数组                              | 二叉树                                   | 哈希表                    |
| 文件管理  | 存储到硬盘的文件为主，需要额外的资源和管理来支持块存储上的文件 | 支持常见的文件协议和权限模式，可提供配置为使用共享文件存储的一个用程序使用 | 将视频、音频、图像、网络元数据文件存储为对象 |
| 元数据管理 | 使用的关联元数据很少                      | 仅存储与文件相关的有限元数据                        | 可以存储任何对象的无限元数据，自定义字段等  |
| 存储性能  | 高性能、低延迟和快速的数据传输                 | 为共享文件访问提供高性能                          | 存储海量不规则数据              |
| 物理存储  | SSD、HDD                         | 本地NAS服务器或者SSD/HDD                     | 分布式存储节点                |
| 跟AI关系 | CKPT权重文件                        | 基础数据格式                                | 训练数据(图像、音频、语料等)        |

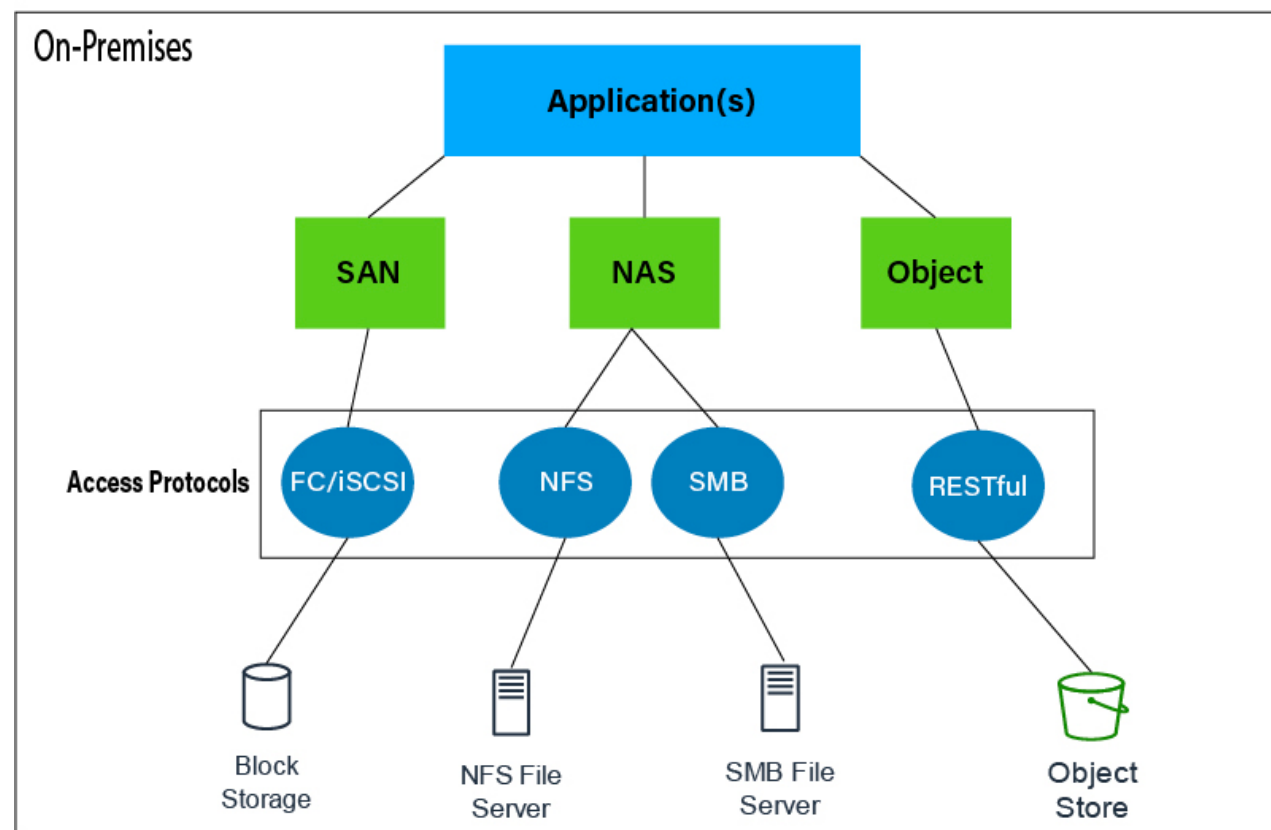
# 2. 使用场景

# 与存储连接方式的场景

- 块存储、文件存储、对象存的使用场景

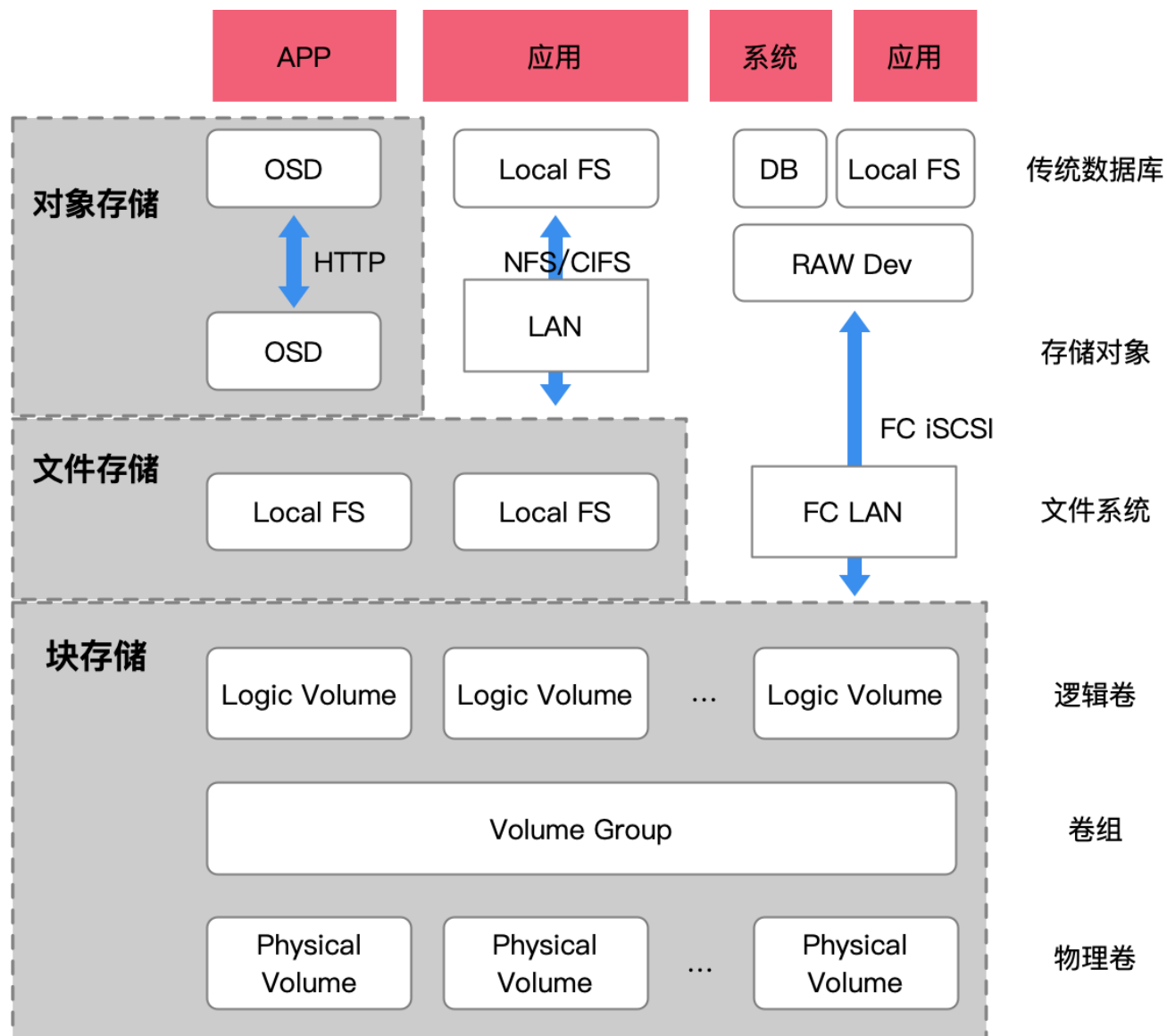


## Storage pattern decision matrix



# 使用场景

- 三者分别提供不同层级、不同粒度的存储方式，是一个逐层递增关系；
- 块存储是字节级存储，文件存储增加了文件层级语义，对象存储封装了对象语义，屏蔽底层细节。



## 三者关系

- **块存储**：为 Database 的结构化数据存储提供基础，满足 Database 高性能需求。低延迟高带宽场景，使用块存储是最佳方案。
  - 主要关心数据在存储介质上的存储位置，为后续的存储类型提供一切物理基础
- **文件存储**：基于块封装了文件、目录的语义结构，利用层级关系组织数据，通过 OS 实现文件系统封装（Ext4），基于大数据封装分布式文件系统（HDFS），应用层提供 XML/JSON 等格式。
  - 从单机、分布式到大数据，提供多粒度文件语义、满足不同层级的场景需求。主要关心文件组织关系
- **对象存储**：通过对象对数据的透明读写，多模态异构数据统一抽象为对象进行操作（AI 中为 Tensor）。

# 小结&思考



## 小结

1. 了解数据存储的主要三种类型（块存储、文件存储、对象存储）及其优缺点
2. 了解三种存储对象在不同场景的具体使用情况



# Reference 引用&参考

1. <https://www.linkedin.com/pulse/dram-scaling-challenges-grow-aken-cheung>
2. [https://nn.wikipedia.org/wiki/Random\\_Access\\_Memory](https://nn.wikipedia.org/wiki/Random_Access_Memory)
3. <https://aws.amazon.com/cn/compare/the-difference-between-block-file-object-storage/>



# Thank you

把AI系统带入每个开发者、每个家庭、每个组织，构建万物互联的智能世界

Bring AI System to every person, home and organization for a fully connected, intelligent world.

Copyright © 2023 XXX Technologies Co., Ltd.  
All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. XXX may change the information at any time without notice.



Course [chenzomi12.github.io](https://chenzomi12.github.io)

GitHub [github.com/chenzomi12/DeepLearningSystem](https://github.com/chenzomi12/DeepLearningSystem)