实 验 报 告 内 容

一・实验内容

实验题目：Hadoop安装使用

实验内容与要求：

1. 在 PC 机上以伪分布式模式安装 Hadoop；
2. 访问 Web 界面查看 Hadoop 信息。

二・实验目的

1. 熟悉 Hadoop 的安装流程。
2. 熟悉 Hadoop 访问 Web 界等基本操作。

三・实验过程截图及说明

1. 实验环境与部署方案

为便于快速搭建可重复的实验环境，本实验使用 `Docker Compose` 在单机上搭建 Hadoop 伪分布式环境。整体由以下角色组成：

1. HDFS：`namenode`、`datanode`
2. YARN：`resourcemanager`、`nodemanager`
3. 实验客户端：`lab-client`（用于执行命令、编译与运行示例程序）

本次部署的核心目标是满足"伪分布式安装 + Web 界面访问验证"。对应的 `docker-compose.yml` 如下：

```yaml
networks:
  bigdata-net:
    driver: bridge

volumes:
  namenode:
  datanode:

services:
  # ----------------------------
  # 1. HDFS 存储层
  # ----------------------------
  namenode:
    image: bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8
    container_name: namenode
    hostname: namenode
    restart: always
    ports:
      - "9870:9870"
      - "9000:9000"
    env_file:
      - ./hadoop.env
    volumes:
      - namenode:/hadoop/dfs/name
    networks:
      - bigdata-net

  datanode:
    image: bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8
    container_name: datanode
    hostname: datanode
    restart: always
    environment:
      SERVICE_PRECONDITION: "namenode:9870 namenode:9000"
    env_file:
      - ./hadoop.env
    volumes:
      - datanode:/hadoop/dfs/data
    networks:
      - bigdata-net

  # ----------------------------
  # 2. YARN 计算层 (MapReduce)
  # ----------------------------
  resourcemanager:
    image: bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8
    container_name: resourcemanager
    hostname: resourcemanager
    restart: always
    ports:
```

```
      - "8088:8088"
    environment:
      SERVICE_PRECONDITION: "namenode:9000 namenode:9870 datanode:9864"
    env_file:
      - ./hadoop.env
    networks:
      - bigdata-net

  nodemanager:
    image: bde2020/hadoop-nodemanager:2.0.0-hadoop3.2.1-java8
    container_name: nodemanager
    hostname: nodemanager
    restart: always
    environment:
      SERVICE_PRECONDITION: "namenode:9000 namenode:9870 datanode:9864
resourcemanager:8088"
    env_file:
      - ./hadoop.env
    networks:
      - bigdata-net

  # ----------------------------
  # 3. Zookeeper & HBase
  # ----------------------------
  zookeeper:
    image: zookeeper:3.8
    container_name: zookeeper
    hostname: zookeeper
    restart: always
    ports:
      - "2181:2181"
    networks:
      - bigdata-net

  hbase:
    image: harisekhon/hbase:latest
    container_name: hbase
    hostname: hbase
    restart: always
    ports:
      - "16010:16010"
    depends_on:
      - namenode
      - zookeeper
    environment:
      - HBASE_MANAGES_ZK=false
      - HBASE_ZOOKEEPER_QUORUM=zookeeper
      - HBASE_ROOTDIR=hdfs://namenode:9000/hbase
      - SERVICE_PRECONDITION=namenode:9000 zookeeper:2181
    networks:
      - bigdata-net
```

```yaml
  # ----------------------------
  # 4. 工作台 (Client)
  # ----------------------------
  client:
    image: bde2020/hadoop-base:2.0.0-hadoop3.2.1-java8
    container_name: lab-client
    hostname: client
    restart: always
    volumes:
      - ./workspace:/workspace
    env_file:
      - ./hadoop.env
    networks:
      - bigdata-net
    environment:
      CORE_CONF_fs_defaultFS: hdfs://namenode:9000
      YARN_CONF_yarn_resourcemanager_hostname: resourcemanager
      SERVICE_PRECONDITION: "namenode:9000 resourcemanager:8088 hbase:16010"
    command: tail -f /dev/null
```

环境变量文件 `hadoop.env` 如下，用于指定 `fs.defaultFS`、关闭权限校验、设置副本数等：

```
CLUSTER_NAME=test
CORE_CONF_fs_defaultFS=hdfs://namenode:9000
CORE_CONF_hadoop_http_staticuser_user=root
CORE_CONF_hadoop_proxyuser_hue_hosts=*
CORE_CONF_hadoop_proxyuser_hue_groups=*
HDFS_CONF_dfs_webhdfs_enabled=true
HDFS_CONF_dfs_permissions_enabled=false
HDFS_CONF_dfs_replication=1
YARN_CONF_yarn_log_aggregation_enable=true
YARN_CONF_yarn_resourcemanager_recovery_enabled=true
YARN_CONF_yarn_resourcemanager_store_class=org.apache.hadoop.yarn.server.resourcemanag
er.recovery.FileSystemRMStateStore
YARN_CONF_yarn_resourcemanager_fs_state_store_uri=/rmstate
YARN_CONF_yarn_nodemanager_remote_app_log_dir=/app-logs
YARN_CONF_yarn_log_aggregation_retain_seconds=-1
YARN_CONF_yarn_log_aggregation_retain_check_interval_seconds=-1
```

2. 启动 Hadoop 伪分布式环境

在宿主机项目目录下执行启动命令：

```
docker compose up -d
```

为后续实验统一入口，使用 `lab-client` 作为操作终端：

```
docker exec -it lab-client bash
```

3. 访问 Web 界面查看 Hadoop 信息

按实验要求访问 Web 界面查看 Hadoop 信息：

1. HDFS 管理界面：`http://localhost:9870`
2. YARN 管理界面：`http://localhost:8088`

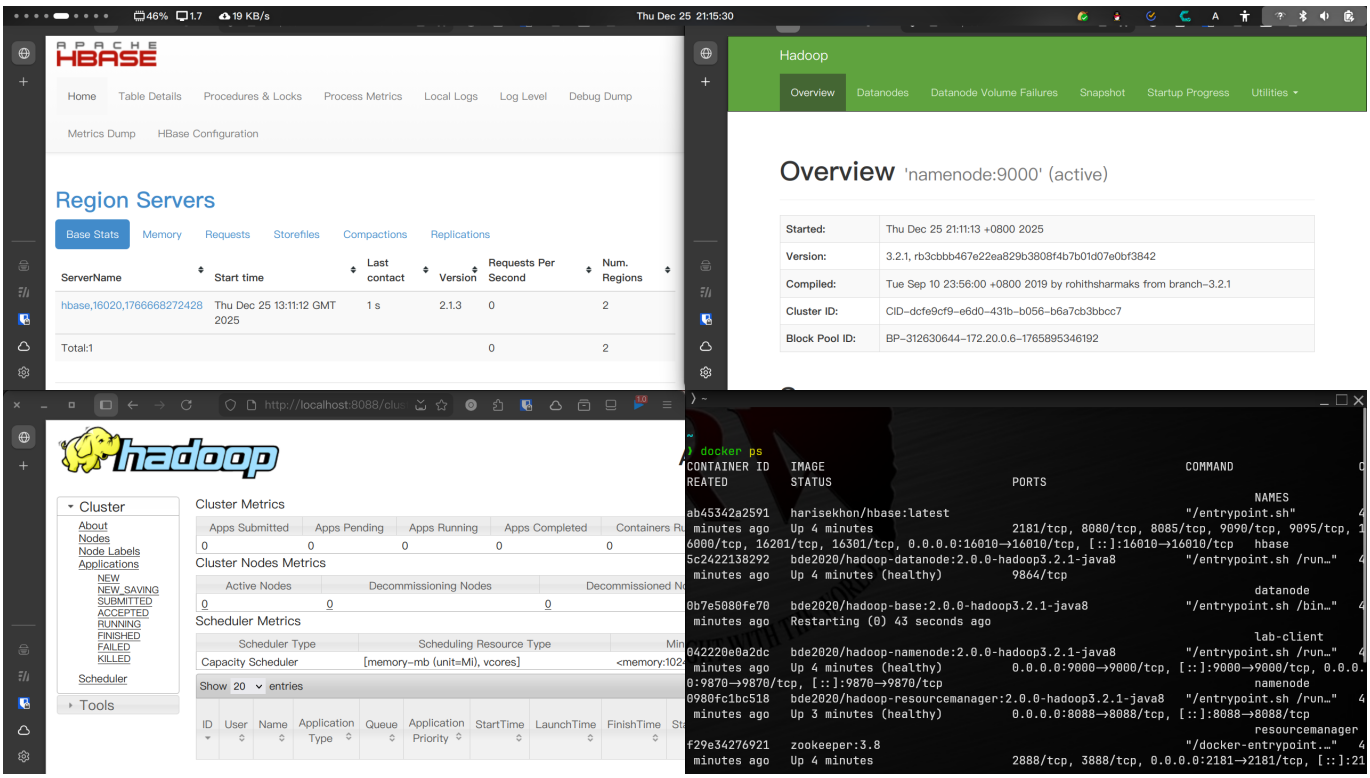

图 1 Hadoop/HDFS/YARN Web 界面与容器状态

四、实验总结及心得体会

实验总结：本实验以 `Docker Compose` 在单机上完成 Hadoop 伪分布式部署，并通过 `http://localhost:9870` 与 `http://localhost:8088` 分别验证 HDFS 与 YARN 的 Web 管理界面可访问，满足实验内容与要求。

心得体会：在容器化部署场景中，端口映射与服务启动顺序是影响可用性的关键因素；通过为关键服务设置 `SERVICE_PRECONDITION` 可以降低"先访问后就绪"导致的误判。同时，将命令执行统一收敛到 `lab-client` 便于后续实验复用环境与工具链，减少在宿主机与容器之间切换带来的不确定性。