# 分布式大数据的复杂任务调度与存储优化策略研究

宋俊苏 [1]
SONG Junsu

**摘 要** 面向大数据存储、复杂任务调度与处理的实际需求，基于 B/S 浏览器 / 服务器网络结构、Hadoop 分布式集群组件架构、NoSQL 数据库、HDFS 文件存储组件、元数据服务器、数据服务器、ECS 云服务器、Client 客户机等软硬件，设置涵盖基础层、网络通信层、数据存储层、任务调度层的多层网络架构，利用任务调度器、任务控制块标注不同任务的数据结构，文章设计应用复杂任务树算法模型进行简单任务组合、父（子）节点任务执行，任务执行成功后从消息队列取出执行后的二进制流数据，提升分布式复杂任务调度处理质量。实验结果表明，基于复杂任务树调度算法的数据块消息队列任务调度准确值 Precision 为 96.12%，数据块任务调度结果和真实标签之间的匹配程度值 AP 为 92.53%，F-Measure 值为 0.903，均显著优于 CNN 卷积神经网络算法的迭代训练结果。

**关键词** 分布式大数据；复杂任务调度；存储；优化策略

doi：10.3969/j.issn.1672-9528.2025.04.032

## 0 引言

根据不同任务树执行的复杂程度，基于元数据服务器、数据服务器、HDFS 文件存储组件、MapReduce 编程计算组件等软硬件，建立起涵盖并行任务、多任务、备选任务的分布式数据任务处理组合模型，由数据服务器响应客户端、元数据服务器承担数据托管（缓存）处理任务，由 HA Agent（高可用代理）在一定时间间隔内向所有服务器响应请求任务的类型信息，使用诸如 Task Base 任务类的静态调度函数设置任务起点、下一可执行的任务节点，利用各级任务节点作出父级（子级）任务判断、任务数据传输转换、数据副本读取和存储，实现跨节点的复杂任务类调度与存储操作 [1]。

## 1 分布式大数据存储系统涉及的重要技术

面对网络平台的海量化大数据资源，继续使用传统网络计算机内存数据存储的方式已难以满足需求，而使用 IBM、EMC 等高端小型机存储设备的代价高昂，因而采用以数据服务器为主的分布式并行数据读写方式，能够满足海量小文件存储的应用场景需求 [2]。

### 1.1 FUSE 文件系统框架技术

FUSE（filesystem in userspace）是用于网络访问控制、文件数据浏览链接的服务组件，主要包括内核模块（fuse.ko）、用户态库（libfuse.*）和挂载工具（fusermount）等组成结构。FUSE 框架提供面向应用程序的 posix 标准接口，根据不同用户的任务数据访问与浏览需求，提供用于引导的 FUSE 文件系统介质、允许用户与网络内核的静态链接，将外部 I/O 访问请求经过 VFS 虚拟文件系统传递至内核 FUSE 文件系统模块，FUSE 将请求转发给用户端的 hello.exe 程序进行处理，响应处理完成后结果沿原路返回 [3]。

### 1.2 VFS 虚拟文件系统技术

VFS（virtual file system）虚拟文件系统是基于 Linux 系统内核的文件接口层，包括 Ext2/3、Reiserfs、XFS 等文件系统的类型结构，主要负责提供统一的系统调用接口、文件操作目录来简化访问程序操作文件，无需使用任何物理硬件设备分配的存储空间，提升任务数据访问、应用程序服务执行的可移植性，并支持提供统一的文件对象结构视图 [1]。如 VFS 虚拟文件系统设置文件目录项数据所在 inode 的编号，用于建立起文件名、inode 之间的关联，查找文件子目录 usr 的 inode 数据段、找到名为 usr、bin 的目录项，根据 inode 编号定位 inode 数据段；在 usr、bin 目录下的 inode 数据段中查找名为 emacs 的目录，该目录内即包含着需查找的文件字符串内容。

### 1.3 分布式通信模型技术

分布式通信为面向任务消息队列的异步通信，在网络任务数据传输中不需要发送方、接收方随时保持运行状态。由任务发送方利用远程机器函数调用模型，向运行在不同的地址空间的被调用方发送参数列表请求，逐个读取、返回网络传输的消息的字节，并隐藏底层网络通信的细节。一般任务

1. 盐城农业科技职业学院 江苏盐城 210023

消息队列传输利用 put()、get()、notify() 等接口，发送方调用 pnt() 接口加入数据请求至发送队列，调用 get() 接口从发送队列内提取相应消息，调用 notify() 接口设置回传调用函数，运用消息转换器作为任务数据的消息格式转换，这一分布式数据通信过程被称为序列化过程 [4]。

## 2 分布式大数据存储系统的总体架构设计

分布式大数据存储系统为异步消息队列处理的架构，主要基于 B/S 结构、Hadoop 分布式集群组件架构、NoSQL 数据库、HDFS 文件存储组件、元数据服务器、数据服务器、ECS 云服务器、Client 客户机等软硬件，建立 TFS（taobao file system）可扩展分布式文件系统，设置涵盖基础层、网络通信层、数据存储层、任务调度层的系统结构，其中 B/S 结构、Hadoop 分布式集群架构为系统建设的底层结构，负责设置系统数据任务调度、存储的不同功能模块；元数据服务器、数据服务器为网络数据处理与管理层，负责响应外部的数据调用与任务处理请求，并完成数据访问的检索、存储和维护操作；ECS 云服务器主要将云计算平台的软硬件资源做出虚拟化，为系统任务调度处理过程中的负载均衡控制提供支持具体的分布式存储、任务调度系统总体架构如图 1 所示 [5]。

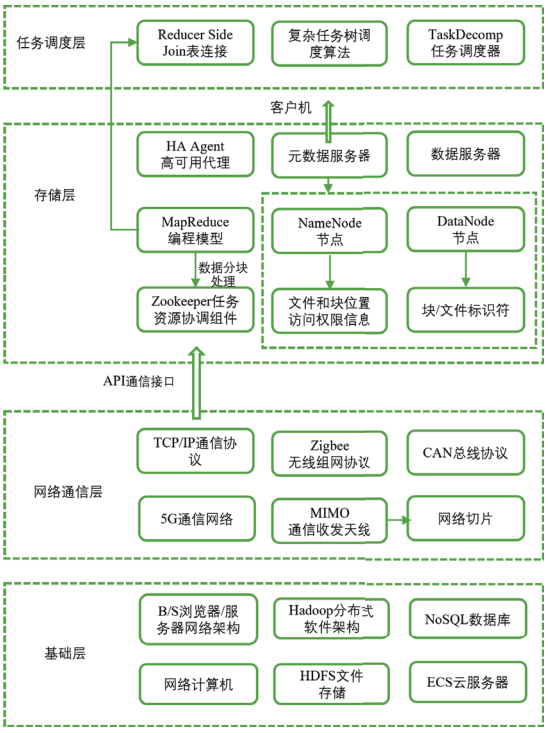


图 1 分布式存储、任务调度系统组成结构

基于 Linux 平台建构的 TFS 高可扩展分布式文件系统，通过设置基础层、网络通信层、数据存储层、任务调度层的多层网络架构可提供海量结构化、非结构化数据存储服务。

（1）网络通信层的用户接入。外部 Client 客户机主要依托 TCP/IP 通信协议、CAN 总线协议接入系统内网，而后基于 Zigbee 无线组网协议、消息队列中间件技术建构起多个 Client 客户机之间的通信互联，通过远程调用（remote procedure call, RPC）方式提供完整消息队列输送、存储通信服务 [6-7]。

（2）存储层的数据任务处理与存储。在分布式数据存储层设置元数据服务器、数据服务器，元数据服务器内设置涵盖 MapReduce 编程模型、Zookeeper 任务资源协调的功能组件，用于记录和管理数据文件来源、结构、格式、访问权限等关键信息；数据服务器则主要将存储单元设为 64 M 大小的数据块（Block），多个小文件数据可通过合并后存储至同一个数据块中，数据块拥有全局唯一的块标识符（Block id）[8]。

外部 Client 客户机的数据管理通常分为数据调用读取、数据存储的两个执行流程，在数据存储时基于元数据服务器中的 MapReduce 编程模型，使用 MapReduce 模型的 RecordReader 读取器读取输入 {0,1,2,...,$n$} 等数据片段，解析出每个数据片段的 {"Tower:Line:ID", Key:"Mac:Date", Value:($V_1$,$V_2$,...,$V_n$)} 键 / 值对，而后利用 MapReduce 模型的 Map、Reduce 函数，将多个小的数据片段（大小为 2 MB 左右）合并为 {($K_0$,$V_0$),($K_1$,$V_1$),……,($K_n$,$V_n$)} 的数据集合，经由 TCP/IP 通信协议、CAN 总线协议将数据信息上传至后台数据服务器完成存储 [9]。

而在数据调用读取过程中，通常 Client 客户机会访问 MapReduce 编程模型的 NameNode 节点、DataNode 节点，获取涵盖 IP 地址、MAC 地址、UDP 报头、内容描述等属性的数据包信息 [6]。MapReduce 编程模型中存在 1 个 NameNode 主节点、多个 NameNode 从节点、多个 DataNode 节点，主从 NameNode 节点用于存储数据目录树结构、数据文件和块的位置信息、访问权限信息，DataNode 节点用于存储数据块标识符（Block id）、文件标识符（File id）信息，具体的数据调用读取流程如下：

（1）由客户机向 NameNode 节点发出文件读取请求，NameNode 查找出文件所属的 DataNode 返回至客户机。

（2）客户机向 DataNode 节点发出文件读取请求，DataServer 查找出可读取的 Block id、并访问数据服务器读取相应的文件或块信息，将结果返回至客户机。TFS 系统会在后台数据服务器中复制存储多个数据块（Block）副本，一旦读取的数据块出现损坏则索引至相同的数据块副本，以保证数据读写的高可靠性。

## 3 面向分布式大数据的复杂任务树调度逻辑设计

### 3.1 分布式大数据的复杂任务分解

传统复杂任务调度与存储模式，是将多个子任务分发至

多个主机服务器分别执行，然而这一串行任务分发模式所消耗的系统负载过高、任务处理耗时过长。本文采用将多个子任务分发至不同节点执行的并行方式，在单个服务器内建构父（子）节点的复杂任务树调度算法模型。按照复杂任务下子任务串行连接、并行连接形式，设置父子任务节点的控制逻辑关系，用递归任务树形结构表示复杂任务的执行逻辑，则包含多个任务树的复杂任务树 [10-11]。

假设复杂任务树调度算法的父节点为 $x_0$，分解的多个子任务节点为 $\{x_1, x_2, \cdots, x_i, \cdots, x_n\}$，则可用 $T_i=(x_i, r_i, \Delta t)$ 的三元组表示复杂任务调度算法模型的任务树属性，其中 $r_i$ 表示当前节点的任务执行状态，且 $r_i \rightarrow \{R, S, F\}$，其中 $\{R, S, F\}$ 分别表示正在执行 (Running)、执行成功 (Success) 以及执行失败 (Failure)，$\Delta t$ 表示时间步长。在此基础上引入 $k \in N$ 的离散时间变量，建构起复杂任务树调度算法模型，具体公式 [12] 为：

$$T_i=(x_i, r_i, \Delta t) \qquad \begin{cases} R_i=(r_i(x)=R) \\ S_i=(r_i(x)=S) \\ F_i=(r_i(x)=F) \end{cases} \tag{1}$$

$$x_{k+t}(t_{k+1})=f(x_k(t_k)) \qquad t_{k+1}=t_k+\Delta t \tag{2}$$

并行任务中的每个子任务 {1,2,...,*n*} 互相独立，子任务执行成功或失败都会上传结果至父任务，某一子任务执行失败不会影响其他子任务的执行，但一个子任务执行失败表明父任务失败，该任务需退回至父节点、加入至待处理任务队列 ReturnTaskQueue 中，并被重新分配给其他子节点继续作出执行。将两个或者多个任务树组成的复杂任务树调度执行方案，具体公式 [13] 为：

$$\text{if} \quad T_i=\text{Sequence}(T_1, T_2)=\text{Success}(T_1) \qquad x_k \in S_1 \tag{3}$$

$$r_0(x_k)=r_2(x_k) \qquad f_0(x_k)=f_2(x_k) \tag{4}$$

式中：父节点 $x_0$ 的子节点 $x_1$ 的任务执行成功、$x_2$ 的任务执行失败，则将 $x_2$ 节点的任务返回至父节点，并标记为 $r_0(x_k)=r_2(x_k)$、$f_0(x_k)=f_2(x_k)$，重新将其分配给其他子节点继续作出任务执行。根据复杂任务树的层次化结构，当增加新的待处理任务类型时，需设置新消息队列、返回任务消息队列两种存放结构，将根节点待处理的返回任务消息下发至子任务节点，子任务节点继续分解为多个叶子进行更细分任务的执行，若 Scheduel(aaa) 返回为空，则表示没有执行的任务，直至所有子任务执行完成后停止 [14]。

### 3.2 复杂任务树的任务调度算法实现

TCB 任务控制块为任务数据属性结构的标注块，为保证分布式大数据异步消息队列的处理，通常利用 TCB 任务块将不同任务处理状态设置为 Running 状态、Wait 状态、Success 状态、Failure 状态，分别表示任务未执行、远端进程任务等待、任务执行成功、任务执行失败。

基于复杂任务树调度器的子任务组合模式，在 TaskBase 类任务执行框架内设置 Running()、RunWait()、RunSuccess()、RunFailure() 的控制接口，使用面向对象编程完成各任务类型之间关系的建模；设置中间任务节点的控制接口 Adds()、Remove()，定义并行任务、串行任务执行成功或失败的条件，利用控制接口操作复杂任务、简单任务的执行 [9]。

以当前输入任务作为开始，引入 TaskBase 类任务、NextAvailTask 类任务的静态函数 TaskBase*、TaskBase* NextAvailTask，连续执行下一子任务节点的任务，复杂任务树调度算法的执行代码 [15] 如下：

```
Input: 任意一个任务节点 ;
连接下一可执行的任务节点 TaskBase*NextAvailTask;
{
while 循环继续执行 (task) 任务节点的任务 ;
{
if(task->_childTaskNum>0&&task->_initChildTaskNum==0)
// 若节点子任务数大于 0 且未初始化，返回第一个子任务 ;
if (task=>_state==S_SUCCESS)
    return task->_ nextSibling&parent
else if(task->_state==S_ ERROR)
     return task->_parent// 当并行任务执行成功时返回下一兄弟节点，任务执行失败时返回父节点 ;
if(task->_parent&&task->_parent->_ type// 当该任务节点的父任务为备选任务 ;
if(task=>_state==S_ ERROR)
return task->_ nextSibling&parent
else if(task->_ state==S_ _SUCCESS)
return task->_ parent// 当任务失败时返回下一兄弟节点或父节点，成功时返回父节点 ;
if( task->_ parent->_ completeChildTaskNum)
return task->_ parent// 所有并行任务任务完成时，返回父结点 ;
```

## 4 仿真实验测试及分析

基于 linux 网络操作系统、元数据服务器、数据服务器、ECS 云服务器等软硬件虚拟出 10 个 VMWare 虚拟机，设置 VMWare 虚拟机的多个网络节点 AP，为局域网分布式任务调度处理的接入作业提供支持。基于 Matlab 2022b 仿真软件执行复杂任务树调度算法的仿真计算，先根据单个数设置根节点、父（子）节点控制逻辑执行的优先级，选定 ROS Behavior-Tree 行为树库作为被测数据对象，依照数据块标识

符（Block id）、标记标签（tag）、内容描述（*.value），设立多级文件目录，利用 put() 函数将数据读取 / 写入请求加入任务消息队列、用 get() 函数提取任务消息队列信息、用 notify() 函数作出执行失败数据回传，得到分布式并行任务处理的执行结果如表 1、图 2 所示 [16]。

表 1 分布式并行任务处理的执行结果

| 算法模型 | Precision/% | AP/% | F-Measure | LBD 负载均衡度 /% |
|---|---|---|---|---|
| 复杂任务树调度算法 | 96.12 | 92.53 | 0.903 | 73.65 |
| CNN 卷积神经网络算法 | 85.70 | 83.24 | 0.875 | 84.12 |

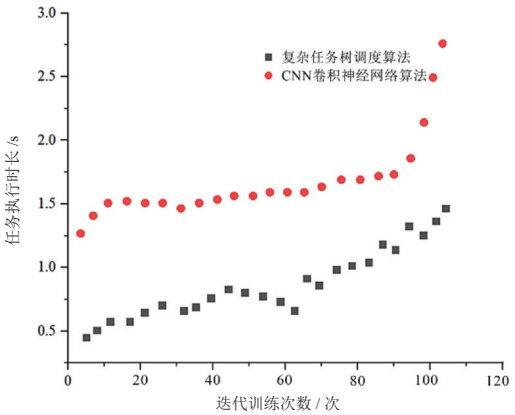


图 2 复杂任务树调度算法的任务执行时长

根据表 1、图 2 的仿真实验结果可得，在云计算任务调度中使用复杂任务树调度算法可缩短系统的虚拟机分配、数据工作流任务处理时长，并提升任务调度执行的负载不均值。基于复杂任务树调度算法的数据块消息队列任务调度准确值 Precision 为 96.12%，数据块任务调度结果和真实标签之间的匹配程度值 AP 为 92.53%，F-Measure 值为 0.903，均显著优于 CNN 卷积神经网络算法的迭代训练结果。且基于复杂任务树调度算法任务执行的 LBD 负载均衡度更优（73.65%），随着数据量不断增加的任务执行时长上升较慢，表现出良好的算法任务调度质量，可在保证任务调度执行质量的同时、降低单个虚拟机任务执行时长。

## 5 结语

面对海量化的复杂任务调度与存储管理需求，由分布式大数据存储系统利用 FUSE 文件系统组件响应用户请求，将任务消息队列交由应用程序客户端作出处理执行，利用任务树、任务树调度器等统一服务结构并发处理多个任务，针对不同的请求任务需修改复杂任务树的执行程序，执行不同数据计算任务调取、取出与返回的操作，在单线程完成所有类型任务的消息队列添加、处理与存储管理，以满足海量化数据任务调度和存储的业务需求。

**参考文献：**

[1] 王纬国，张子明，刘良勇．基于机器视觉的航空仪表测试方法研究 [J]. 中国设备工程，2021(21):155-156.

[2] 盛伟峰．基于黑盒测试技术的有线电视收视用户标签化系统测试方法研究 [J]. 广播电视网络，2022,29(5):69-71.

[3] 张宝斌．面向安全性分析的嵌入式软件测试方法研究 [J]. 电子测试，2020(11):117-118.

[4] 邝祝芳，陈志刚，陈清林，等．基于深度强化学习的多用户边缘计算任务卸载调度与资源分配算法 [J]. 计算机学报，2022,45(4)：812-824.

[5] 缪巍巍，王传君，张明轩，等．一种面向多物联代理在线应用的弹性资源调度算法 [J]. 重庆理工大学学报 ( 自然科学 ), 2022, 36(2): 151-161.

[6] 方霁，王红胜，刘伟东，等．云转播系统测试方法研究 [J]. 广播与电视技术，2021, 48(12):40-43.

[7] 董丽，赵琪，周健．嵌入式软件安全性分析和测试方法研究 [J]. 信息系统工程，2021(5): 70-71.

[8] 徐荣，魏莉．面向深度学习训练的异构任务调度研究 [J]. 西安文理学院学报 ( 自然科学版 ), 2023, 26(3):35-39.

[9] 康玲玲，丁立业，姜祁峰．RFID 读写器灵敏度测试方法研究 [J]. 中国集成电路，2022,31(7):80-82.

[10] 向啟苗．基于 Golang 的文件存储优化程序实现 [J]. 电脑编程技巧与维护 ,2023(5): 77-79.

[11] 金国栋，卞昊穹，陈跃国，等 .HDFS 存储和优化技术研究综述 [J]. 软件学报，2020, 31(1): 137-161.

[12] 刘深，王利朋，刘光享．一种可信的分布式异构存储系统 [J]. 电脑编程技巧与维护，2024(2):95-98.

[13] 赵军富，杜海渊，靳永胜，等．分布式 3D 打印服务的实时多任务调度研究 [J]. 制造技术与机床 ,2024(4):188-195.

[14] 张志强，徐泉，刘文庆，等．分布式实时数据采集与传输系统的研究 [J]. 控制工程，2020, 27(9) :1582-1588.

[15] 许富景，杜少成，张燕，等．应用混沌反向灰狼算法的多终端复杂任务调度策略 [J]. 中国测试，2024, 50(10): 73-80.

[16] 胡爱军，李楚进．基于分布式大数据的 Expectile 回归分析 [J]. 应用数学，2022, 35(4):974-981.

**【作者简介】**

宋俊苏（1973—），男，江苏盐城人，硕士，教授，研究方向：计算机教育教学、计算机网络技术等。

（收稿日期：2024-12-02）