DOI: 10.16452/j.cnki.sdkjzk.2023.04.011 文章编号:1672-3767(2023)04-0092-09

# GM 卫星测高数据海面时变校正程序的 I/O 并行优化研究

傅 游,徐方正,梁建国
(山东科技大学 计算机科学与工程学院,山东 青岛 266590)

**摘 要**:海面时变校正是建立平均海平面模型数据处理过程中最关键的一步,可以消除或削弱海面时变的影响。本研究实现了时空客观分析法对 GM 卫星测高数据进行海面时变校正的串行程序,针对该程序运行耗时超长问题,研究其 I/O 特征,提出两种多进程 I/O 并行方案,并使用消息传递接口(MPI)文件视口函数对其进行优化。在高性能集群系统上的实验结果表明,合并再分配方案具有更好的负载均衡度;I/O 并行优化后的合并再分配方案加速效果明显,最高开启 448 进程时,大数据量实验耗时 9 283.84 s,与 56 进程时相比,加速比为 7.21,并且具有良好的强、弱可扩展性。

**关键词**:卫星测高;海面时变校正;客观分析法;消息传递接口;并行计算;I/O 性能
**中图分类号**:TP311.5 **文献标志码**:A

## I/O parallel optimization of sea surface time-varying correction program for GM satellite altimetry data

FU You, XU Fangzheng, LIANG Jianguo
(College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China)

**Abstract**: Sea surface time-varying correction is the most important step in the data processing process of establishing mean sea level model, which can eliminate or weaken the sea surface time-varying effect. This study designed and implemented a serial program for the objective analysis of time and space to perform sea surface time-varying correction on GM satellite altimetry data. To address the problem of the program's excessively long processing time, the I/O characteristics were investigated and two multi-process I/O parallel schemes are proposed, which were optimized by using the massage passing interface (MPI) file view function. Experimental results on a high-performance cluster system show that the merging and redistributing scheme has better load balance. When 448 processes are used, the large-scale data experiment takes 9 283.84 s and the speedup ratio is 7.21 compared to that when 56 processes are used, indicating that the optimized merging and redistributing scheme has a significant acceleration effect as well as good strong and weak scalability.

**Key words**: satellite altimetry; sea surface time-varying correction; objective analysis; MPI; parallel computing; I/O performance

平均海平面(mean sea surface,MSS)是指相对于参考椭球在一定时间内的平均动态海面高,由平均海面地形和大地水准面两部分相加得到[1]。MSS 在研究地壳形变、大洋环流、海洋重力计算[2]、大地水准面起

**收稿日期**:2022-11-30
**基金项目**:山东省自然科学基金项目(ZR2022MF274)
**作者简介**:傅 游(1968—),女,山东聊城人,教授,博士生导师,主要从事高性能计算、并行优化、机器学习等方面的研究.
梁建国(1981—),男,山西霍州人,副教授,博士研究生,研究方向为高性能计算,本文通信作者.
E-mail:183549746@qq.com

伏确定和地壳形变[3]等问题中得到广泛应用,对地球科学和环境科学研究具有重要意义。

建立 MSS 模型面临的挑战是在有限的时间跨度内实现对时间海面变化的最精确滤波,同时获得最高的空间分辨率。通常结合来自精确重复任务(exact repeat mission,ERM)的数据与 ERS-1、GEOSAT 等早期的大地测量任务(geodetic mission,GM)测高数据实现[1]。校正 GM 数据中的海面时变信号时,需要读取大量的卫星轨迹数据文件[3],比如在构建日本海周边区域的平均海平面模型中对 GM 数据进行海面时变校正时,共使用了 5 颗卫星的 GM 卫星测高数据及其对应的 ERM 测高数据[4]。以 Crystat-2 卫星(2011.01.28—2019.12.12)为例,其 GM 卫星测高数据包含 112 个周期,总共 92 669 个轨迹文件。生成结果的精度和分辨率要求越高,I/O 的数据越多,计算耗时和 I/O 耗时越长,必须利用并行计算技术进行加速。并行计算已在卫星数据处理领域发挥了重要作用[5-7]。在 I/O 性能优化方面,Schenck 等[8]提出将突发式数据缓存使用快速存储介质作为缓冲区,将进程之间由 I/O 引起的负载不平衡降到最低限度,同时加快 I/O 的整体速度;Thakur 等[9]提出的 ROMIO 使用双阶段 I/O 调度算法优化后的集中式 I/O,尽可能对同一文件但不同数据段的多次访问进行合并以减少访问次数;Behazad 等[10]使用遗传算法考虑影响 I/O 性能的各个参数,再进行全局空间搜索,从而寻找最优参数解;Chen 等[11]使用遗传算法对并行 I/O 性能进行自动调优;Guedes 等[12]针对运行在基于容器的服务器虚拟化集群上的 I/O 密集型应用进行研究,在虚拟环境下提供缓存服务,在大规模集群的存储文件系统中(如 Lustre、通用并行文件系统和 Panasas 文件系统),将单个文件分为多个子文件存储在多个数据服务器上,通过服务器的并发来提高 I/O 效率。这些方法在一定程度上确实能减少 I/O 耗时,但不同程序具有不同 I/O 特征,要取得更好的 I/O 优化效果,必须针对具体程序进行分析,制订具体优化策略。

在全球平均海面模型的建立过程中,在 Yuan 等[13]研究基础上,开发了基于时空客观分析法的 GM 卫星测高数据海面时变校正的串行程序,完成了向高性能集群系统的移植。该串行程序读取的多源卫星总数据量约 2 TB,输出数据约 500 GB,总读取轨迹文件数约 10 000 万个。在 CPU 为 Intel i7-10875H、内存 16 GB 的个人电脑上需运行约 3 个月,严重影响研究进度。而在完成高性能集群系统移植后,程序计算时间大大减少,但 I/O 作为系统性能瓶颈的情况并未改善,影响了系统可扩展性。

为了缩短 I/O 耗时,实现系统可扩展性,本研究从两方面对 GM 卫星测高数据海面时变校正程序 I/O 特征进行分析;为了提高 I/O 效率,提出按周期分配方案,并针对该方案可扩展性不佳、易导致负载不均衡的问题,提出一种合并再分配方案;使用消息传递接口(massage passing interface,MPI)文件视口函数对合并再分配算法进行优化,进一步提高 I/O 效率。

## 1　GM 卫星测高数据海面时变校正程序

### 1.1　GM 卫星测高数据的海面时变校正程序计算过程

以进行海面时变校正的 GM 卫星测高数据在相同时间跨度的 ERM 测高数据为参考基准,先将 ERM 测高数据的海平面异常(sea level anomalies,SLA)与 GM 卫星测高数据进行时空匹配,再进行海面时变校正,即可得到校正后的 GM 卫星测高数据。采用时空客观分析法进行 GM 卫星测高数据海面时变校正的关键是如何选取与待校正 GM 卫星测高数据在时间和空间相匹配的 ERM 测高数据并计算 SLA。GM 卫星测高数据海面时变校正串行算法流程如图 1 所示,图中左侧为 ERM 的 SLA 数据的筛选和计算过程,右侧为 GM 卫星测高数据的筛选和计算过程。

### 1.2　GM 卫星测高数据的海面时变校正程序 I/O 特征分析

在对原串行 Fortran 程序使用插桩法进行热点检测后,发现该串行程序读写文件耗时占比达 80.44%,而数据计算部分仅占 10.37%。若能优化程序的 I/O 部分,则该程序的整体耗时会大幅减少。

时空客观分析法的特征是在考虑时空尺度的前提下,将沿测高轨迹的 SLA 数据格网化为规则的格网 SLA,再对同时空范围内的 GM 卫星测高数据进行校正。GM 和 ERM 卫星测高数据文件结构的特点是包含多个周期文件夹且每个周期文件夹包含多个轨迹文件,文件目录结构如图 2 所示。

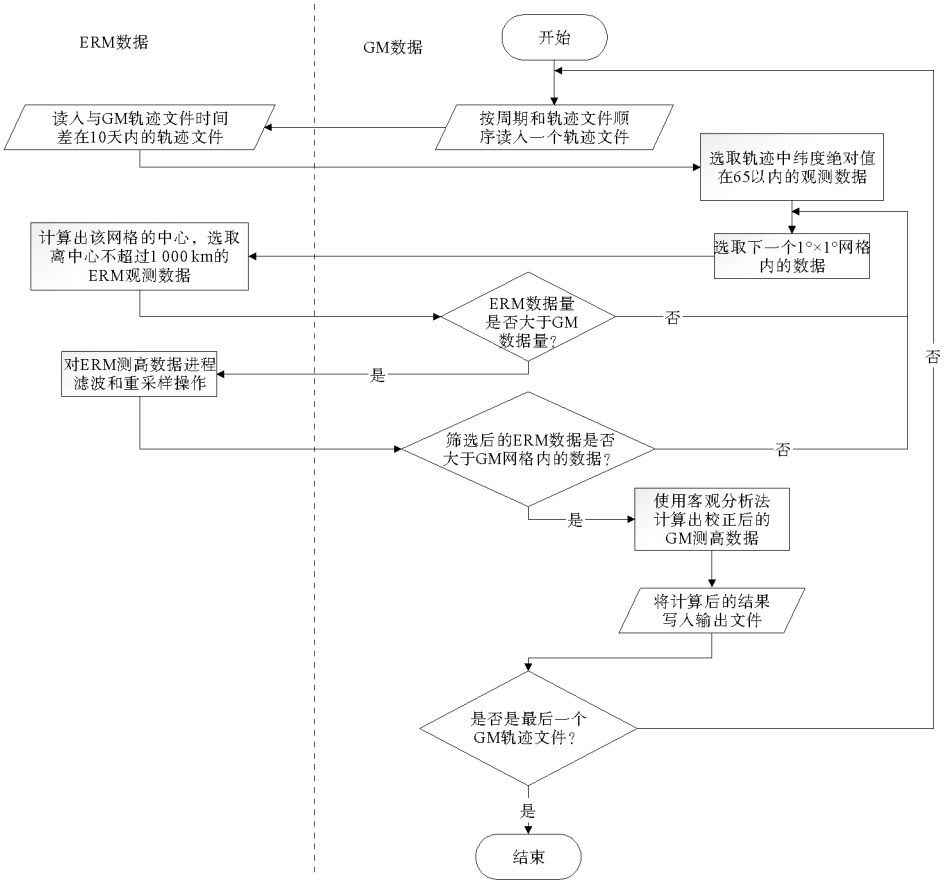


**图 1 串行算法流程图**

Fig. 1 Flow chart of serial algorithm

轨迹文件中以文本形式记录卫星测高数据，GM 卫星测高数据和 ERM 卫星测高数据文件内容结构分别如图 3、图 4 所示。图 3 和图 4 中，每一行代表一个观测点的信息，每一列代表一个属性，其他信息筛除。GM 卫星测高数据的观测点信息属性包括观测时刻、经度、纬度、动态海面高、大地水准面和平均动态海面高，其中大地水准面和平均动态海面高在本研究的海面时变校正中未被采用，但在后续海面高建模中采用，因此未去除。ERM 卫星测高数据包含时刻、经度、纬度和 SLA。

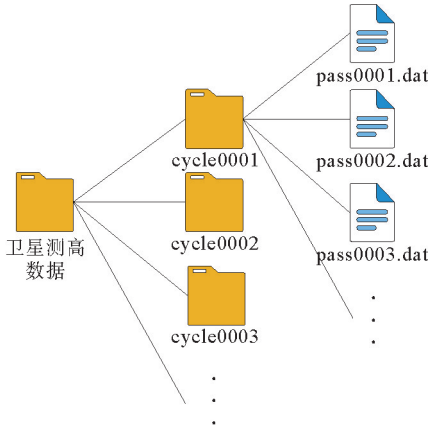


**图 2 卫星测高数据文件目录结构图**

Fig. 2 File directory structure of satellite altimetry data

GM 卫星测高数据海面时变校正程序 I/O 占比大的主要原因包括：①测高数据分布在数量繁多的轨迹文件中，如引言中提到的 Crystat-2 卫星包含 92 669 个轨迹文件，而对应的同时期 ERM 轨迹文件 83 058 个，共计 175 727 个轨迹文件；②读入 GM 卫星测高数据和查找时空对应的 SLA 数据过程中，需要多次读入不同文件，频繁切换文件句柄，切换频率约 5 000 次/s。每次完成一个观测点的计算均需要写入文件，每次写入的数据量仅 48 字节，输出文件时需写入约 10 000 次/s。可见 GM 测高数据时空客观分析法程序具有 I/O 密集型程序的特征，频繁的 I/O 导致程序运行速度受限于 I/O 带宽，无法充分发挥大规模集群中多核计算机性能。

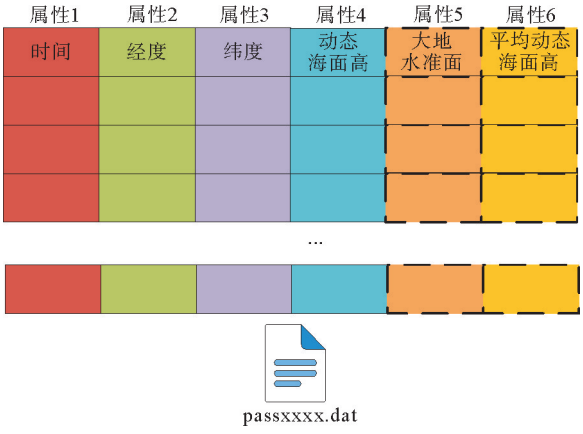


**图 3　GM 卫星测高数据文件内容结构图**

Fig. 3　File content structure of GM satellite altimetry data

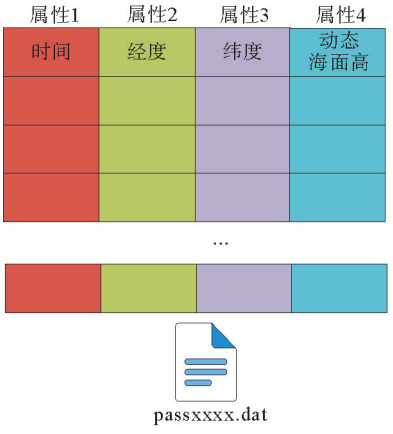


**图 4　ERM 卫星测高数据文件内容结构图**

Fig. 4　File content structure of ERM satellite altimetry data

## 2　GM 测高数据海面时变校正时空客观分析法程序的 I/O 并行优化

### 2.1　按周期分配数据的并行方案

为了将该程序移植到高性能集群系统,实现多进程并行 I/O,本研究提出一种按周期分配数据的并行方案。在该方案中,以一个周期的 GM 卫星测高数据为最小任务分配粒度,即每个进程处理整数个周期的 GM 卫星测高数据,尽可能把所有数据按周期数平均分配给每个进程进行计算,两两进程之间分配到的周期数差最大为 1。设 $n$ 为总进程数,$T$ 为总任务数,进程 $i$ 处理的任务数为 $t_i$,

$$t_i=\begin{cases}T/n+1, 0\leqslant i<T\%n;\\ T/n, \quad T\%n\leqslant i<n。\end{cases}\tag{1}$$

式中:/为取商运算,%为取余运算。

在多进程运行时发现,各进程运行时间会有较大差异。以 10 个不完整周期(包含约 100 个轨迹文件)的 GM 卫星测高数据为测试集,该方案并行程序分别开启 1 到 10 个进程的运行时间与加速比如图 5 所示,9 进程时各进程运行时间如图 6 所示。

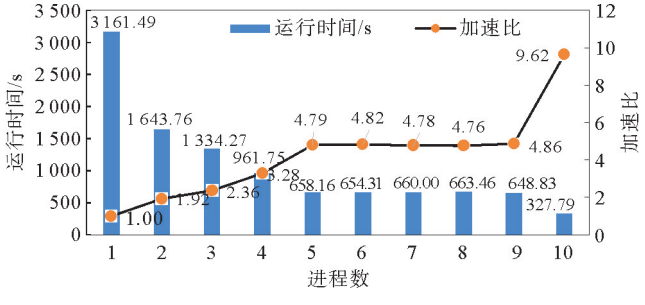


**图 5　并行程序运行时间与加速比图**

Fig. 5　Running time and speedup of the parallel program

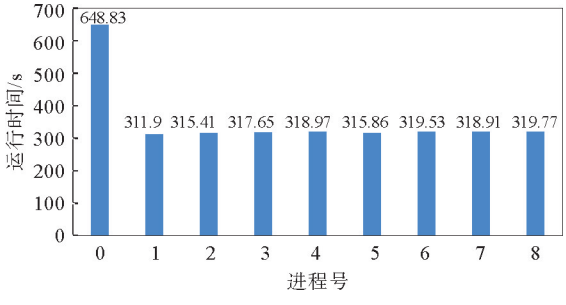


**图 6　并行程序 9 进程时各进程运行时间**

Fig. 6　Running time of the parallel part of the parallel program when the number of processes is set to 9

由图 5 可以看出,当进程数由 5 升至 9 时,运行时间基本保持不变;只有当进程数为 1、2、5 和 10 时,加速比与进程数接近。由图 6 可以看出,0# 进程的运行时间约为其他进程的 2 倍,产生了明显的负载不均衡。由式(1)可知,当 $T$ 不能被 $n$ 整除时,分配给各个进程的任务数不均等,任务差为一个周期的 GM 卫星测高数据校正任务。任务差越大,负载不均衡的情况越明显。由于 GM 卫星测高数据各周期的数据量不等,即使每个进程都分配相同数量的周期,实际的任务数据量差别也很大。除此之外,按周期分配方案的可扩展性不佳,能开启的进程数不能大于周期数。

### 2.2 合并再分配的并行方案

由 1.2 节中的数据文件结构分析可知,若将行数据均匀地划分给每个进程,需将所有文件的行数进行汇总,统计每个进程的行数。由于数据分布在多个轨迹文件中,在进行 SLA 数据匹配过程中依然存在频繁切换文件的问题,出现同一个轨迹文件数据被分配给不同进程的情况,降低任务分配的效率。为避免此类问题的发生,本研究提出合并再分配的并行方案。在实际的任务计算中,最小计算任务单位是一个GM卫星测高的观测点数据,即一行数据,若能以行为进程分配最小任务单位,可大大提高负载均衡程度。

合并再分配的并行方案将所有卫星测高数据和轨迹文件汇总为如图 7 所示的 $n_{att}$ 个属性文件。图 7 中,$c_{all\_input}$ 表示的汇总文件总行数,即各进程需要读入的行数之和,$c_{i\_input}$ 表示进程 $i$ 分到的卫星测高数据行数。将卫星测高数据汇总整合为数个文件,文件个数为需要测高数据中包含的属性数(如时间、经度、纬度等),并按时间顺序排序。ERM 数据的汇总属性文件除了图 4 所示的 4 个属性外,每行数据还需要加上该观测点的周期号和轨迹号,以便于时空匹配。其中,

$$c_{i\_input}=\begin{cases}c_{all\_input}/n+1, & 0\leqslant i<c_{all\_input}\%n;\\ c_{all\_input}/n, & c_{all\_input}\%n\leqslant i<n。\end{cases}\tag{2}$$

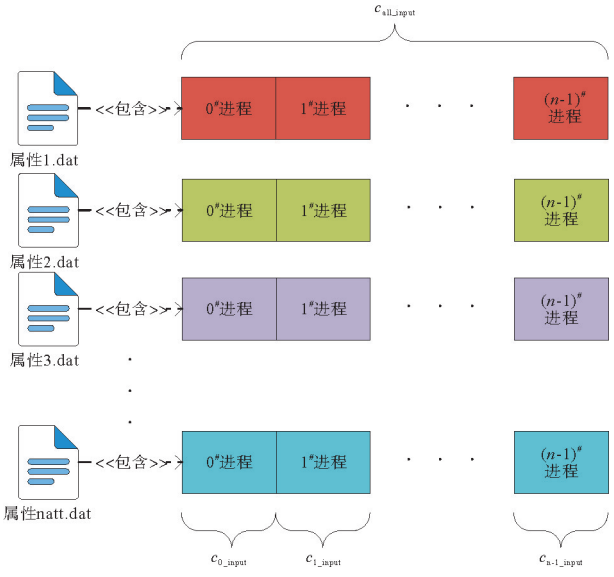


**图 7 合并再分配方案图**

Fig. 7 Merging and redistributing scheme

设进程 $i$ 在汇总文件开始读数据的起始偏移量

$$o_{i\_input}=\begin{cases}i\cdot c_{all\_input}/n+i, & 0\leqslant i<c_{all\_input}\%n;\\ i\cdot c_{all\_input}/n+c_{all\_input}\%n, & c_{all\_input}\%n\leqslant i<n。\end{cases}\tag{3}$$

各进程可以自行计算出 $c_{i\_input}$ 和 $o_{i\_input}$,无需通信。

基于合并再分配方案的并行 GM 卫星测高数据海面时变校正具体实现如算法 1 所示。

**算法 1** 基于合并再分配方案的并行 GM 卫星测高数据海面时变校正算法

**输入**:GM 卫星测高数据 *gm_data*,ERM 卫星的 SLA 数据 *sla_data*

**输出**:校正后的 GM 测高数据 *result_data*

```
Begin
MPI 及其常用参数初始化,获取进程号 i,进程数 n
多进程读取 sla_data 的不同轨迹文件,并将其汇总为 6 个 SLA 属性文件
多进程读取 gm_data,并将其汇总为 6 个 GM 属性文件
根据 i 和 n 确定本进程读取的 gm_data 在文件中的起始位置与长度
每个进程将本进程 gm_data 时间差在 10 天内的 SLA 数据读入内存
前 6 号进程读取 sla_data 并广播给所有进程,存入 6 个 SLA 属性数组
end_line_gm_data←start_line_gm_data + my_gm_data_length
for start_line_gm_data←my_gm_data_offset to end_line_gm_data by max_size do //每次循环取最多 max_size
    个观测点的 GM 数据,max_size 为一次最多可读取观测点数,自行设置
    block_gm_data←read_gm_data_to_array(start_line_gm_data,max_size,end_line_gm_data)//将本次循环要
    校正的 gm_data 读入内存
    for grid_gm_data in block_gm_data do //按顺序从当前的 block_gm_data 中选取 1°×1°格网内的 gm_data
        grid_sla_data←choose_sla_data(my_sla_data) // 筛选距离网格中心不超过 1 000 km 的 sla 数据
        grid_sla_data←filter_sla_data(grid_sla_data) // 进行滤波和重采样操作
        grid_result_data←objective_analysis(grid_sla_data,grid_gm_data)//使用时空客观分析法计算出校正后
        的 GM 卫星测高数据
```

```
15：        block_result_data←add_result_data(block_result_data,grid_result_data)//将当前格网的结果数据加入到
            当前周期结果数据
16：    end for
17：  write_block_result_data(block_result_data)//将一个 block 的结果数据写入磁盘
18：  end for
19：  0# 进程将所有 block 的结果文件汇总为一个文件并输出
20：  End
```

### 2.3　利用 MPI 文件视口函数实现 I/O 并行加速

为了减少合并再分配时对文件汇总的耗时,利用 MPI 文件视口函数实现多进程 I/O 并行加速。MPI 文件视口会给每一个进程定义一个独立文件指针,读写位置由当前文件指针确定[14],各进程可以同时读/写一个文件,互不干扰。

除了读/写的文件路径之外,每个进程在读/写前还需要获取自己在文件中进行读/写操作的偏移量和读/写的数据量。每个进程读操作的数据量依据进程数平均分配,并行读操作的文件起始偏移量可根据其进程号用式(3)计算得到。但是在并行写操作时,输出文件中的数据顺序需要与进程顺序保持一致,而每个进程写入输出文件的数据量并不相等,所以进程号大于 0 的进程进行写操作时的偏移量需由上一个进程传入。

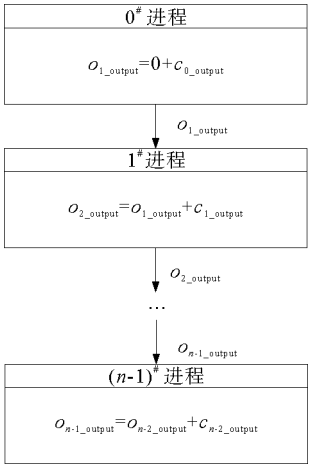


**图 8　写文件偏移量通信流程**

Fig. 8　Flow of write file offset communication

各进程偏移量通信流程如图 8 所示。图 8 中,$c_{i\_output}$ 为 $i$ 号进程需要向输出文件写入的数据行数,$o_{i\_output}$ 为 $i$ 号进程在输出文件中的起始偏移量,

$$o_{i\_output}=\begin{cases}0, & i=0;\\ c_{i-1\_output}+c_{i-1\_output}, & 0<i<n。\end{cases} \tag{4}$$

所有进程的偏移量传输过程按进程号从小到大依序进行,在获得其输出文件的偏移量后,通过 MPI 文件视口输出函数将结果数据并行写入输出文件。

传统的主从并行方案(以下简称传统方案)中,往往只将计算任务并行化,输入数据时需要由主进程读文件并向从进程分发数据;输出数据时需要从进程向主进程发送数据,再由主进程写文件。并行 I/O 方法可避免主从进程间进行大量数据通信,理论上将读/写时间缩短为传统方案的 $1/n$ 。

## 3　实验测试

### 3.1　实验平台

实验平台为 17 台多核计算机组成的集群系统,其中 16 台为计算节点,1 台为管理节点。计算节点的整体配置相同,均包含 2 个 14 核 Intel Xeon Gold 6132 处理器和 96 GB 的 DDR4 内存。因实验平台使用多处理器计算(multi-processor computing,MPC)技术,每个核心可开启一个进程,可开最大进程数为 16×28=448。实验平台软件环境如表 1 所示。

**表 1　实验平台软件环境**

Table 1　Software environment of experimental platform

| 操作系统 | 编译环境 | 运行环境 | 并行环境 | 作业管理 |
|---|---|---|---|---|
| CentOS7 64 bit Linux | GCC 4.8.5 | MKL 库 | MPICH 3.2 | PBS |

### 3.2　实验数据集

实验数据集来源于 AVISO(archiving, validation and interpretation of satellite oceanographic data)发布的沿测高

轨迹的 Level-2+(L2P)SLA 数据产品(以下简称 L2P)。L2P 包含多个卫星测高数据,实验使用其中的 T/P、Jason-2 和 Cryosat-2 测高卫星数据集。L2P 中每个测高卫星均包含若干周期数据,每个周期数据中又包含若干轨迹数据,而每一个轨迹数据均包含了观测时间、纬度、经度、卫星轨道高度、卫星到星下点距离、SLA、平均海面高、有效数据标志以及各项误差改正项等。实验数据集对于 GM 数据,保留了观测时刻、经度、纬度、动态海面高、大地水准面和平均动态海面高;对于 ERM 数据,保留了观测时刻、经度、纬度和 SLA,将该数据集称为大数据集。小数据量数据集系从 L2P 中按固定时间间隔抽选的文件得到。两组数据集的具体信息如表 2 所示。

**表 2 卫星测高数据信息表**

Table 2 Satellite altimetry data

| 数据量 | 数据种类 | 测高卫星 | 起止周期编号 | 观测时间跨度 | 轨迹文件数(周期) |
|---|---|---|---|---|---|
| 小数据集 | GM 数据 | Cryosat-2 | 14～25 | 2011.01.28—2011.12.31 | 90～99 |
| | ERM 数据 | T/P | 674～708 | 2011.01.20—2012.01.02 | 37 |
| 大数据集 | GM 数据 | Cryosat-2 | 14～125 | 2011.01.28—2019.12.12 | 830～840 |
| | ERM 数据 | Jason-2 | 94～421 | 2011.01.20—2019.12.17 | 254 |

### 3.3 实验结果与分析

1) 负载均衡度测试

为了验证合并再分配方案的负载均衡度的优势,使用小数据量数据集,收集了在单节点上开启 6～10 进程时,按周期分配方案和合并再分配方案的各进程空闲时间之和,结果如图 9 所示。

由图 9 可以看出,按周期分配方案的进程空闲时间之和,在 6 到 9 进程时随着进程数的上升而增加,而在 10 进程时大幅降低。这是因为 10 进程时,周期数刚好能被进程数整除。而合并再分配方案的进程空闲时间在 6 到 9 进程时远低于按周期分配方案,只有在 10 进程时相近,且随着周期数上升幅度很小,可见合并再分配方案的负载均衡度优于按周期分配方案。

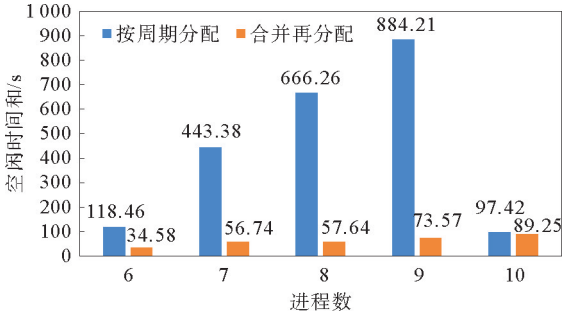


**图 9 多进程下不同并行方案进程空闲时间和**

Fig. 9 Sum of the process idle time of different parallel schemes under multiple processes

2) 两种方案和 I/O 并行加速效果测试

使用原版串行程序利用小数据量数据集进行实验,耗时为 1 027.89 s。

为验证两种并行方案以及 I/O 并行加速的优化效果,使用小数据集分别进行测试。小数据集包含 10 个周期,使用按周期分配方案时,若进程数超过 10 会出现闲置进程,故最大进程数不超过 10。合并再分配方案中待匹配的 ERM 数据有 6 个属性(时间、经度、纬度、SLA、所在周期、所在轨迹),每个进程读入一个属性,故进程数至少为 6。测试结果如图 10 所示。

由图 10 可以看出,对比串行程序,按周期分配方案在单进程时耗时更长,是因为启用 MPI 需要耗时并且多了任务分配的步骤。6 到 9 进程时,按周期分配方案的耗时基本不变,这是因为出现了任务分配不均,只分配到 1 个周期的进程需要等待 2 个周期的进程执行结束。

合并再分配方案在 6 和 10 进程时比按周期分配方案耗时更长,是因为分配任务的步骤更加复杂,当按周期分配方案中周期数可以被进程数整除或者余数很小时,按周期分配方案负载较为均衡,合并再分配方案的负载均衡优势没有体现出来。当进程数为 7 到 9 时,按周期分配方案的负载产生了很大的不均衡,耗时大于合并再分配方案。同时,I/O 并行加速对于按周期分配方案也有一定的加速效果,在 6 进程时加速最多,缩短耗时 32.47 s,加速 13.4%。

3) 强、弱可扩展性测试

为了验证经 I/O 并行加速后的合并再分配方案的强、弱可扩展性,分别使用小数据量和大数据量数据集进行测试。

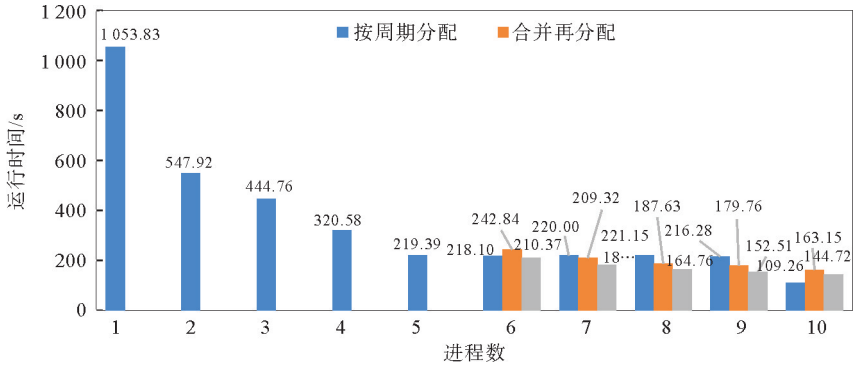


**图 10　多进程下不同并行方案运行时间**

Fig. 10　Running time of different parallel schemes under multiple processes

许多并行计算平台(例如高性能计算集群)能更高效地处理进程数为 2 的次方数的并行程序,这是因为此类平台通常使用硬件结构来组织二叉树或超立方体拓扑中的节点,使用 2 的次方数作为进程数允许进程自然映射到硬件拓扑中的节点,可以减少通信延迟并提高性能。因合并再分配方案的进程数不得小于 6,故最小进程数设置为 8。

小数据量实验组的加速效果如表 3 所示。表 3 中,加速比 1 由串行程序运行时间除以测试算例运行时间得到,加速比 2 由 8 进程并行程序运行时间除以测试算例运行时间得到,并行效率由加速比 1 除以进程数得到。由表 3 可知,运行时间在 64 进程时最短,为 86.41 s,且加速比 1 为 11.90,加速比 2 为 1.91,但是并行效率最差,为 0.19。并行效率在进程数为 8 时最高,为 0.78。

**表 3　小数据量实验组加速效果**

Table 3　Acceleration effect of experimental group with small amounts of data

| 进程数 | 使用节点数 | 使用核心数/节点 | 运行时间/s | 加速比 1 | 加速比 2 | 并行效率 |
|---|---|---|---|---|---|---|
| 8 | 8 | 1 | 164.76 | 6.24 | 1.00 | 0.78 |
| 16 | 16 | 1 | 95.24 | 10.79 | 1.73 | 0.67 |
| 32 | 16 | 2 | 89.95 | 11.43 | 1.83 | 0.36 |
| 64 | 16 | 4 | 86.41 | 11.90 | 1.91 | 0.19 |

小数据量实验组中,在进程数达到 16 后,运行时间再无明显缩减,并行效率随着进程数的增加而减少。8 进程到 16 进程加速比 1 和加速比 2 上升明显,但是进程数继续增加后,没有明显提升。原因是小数据量实验组的数据量较小,符合 Amdahl 定律,小数据量实验不具有良好的强可扩展性,难以体现出并行计算的优势。

大数据实验组中数据量大,最小进程数若设置太小会导致运行时间过长,并且可设置最大进程数为 448,故设置并行运行的最小进程数为 448 的 1/8,即 56。表 4 为大数据量实验组的加速效果,加速比由并行程序 56 进程运行时间除以测试算例运行时间得到。

**表 4　大数据量实验组加速效果**

Table 4　Acceleration effect of experimental group with largeamounts of data

| 进程数 | 使用节点数 | 使用核心数/节点 | 运行时间/s | 加速比 |
|---|---|---|---|---|
| 56 | 8 | 7 | 66 957.45 | 1.00 |
| 112 | 16 | 7 | 37 254.21 | 1.80 |
| 224 | 16 | 14 | 20 229.49 | 3.31 |
| 448 | 16 | 28 | 9 283.84 | 7.21 |

由表 4 可知,运行时间随着进程数的每次翻倍,加速比的增长速度也接近翻倍,在进程数为 448 时最短,为 9 283.84 s,加速比也在此时达到最高 7.21,并且加速比的上升速率没有明显降低,说明该算法具有良好的强可扩展性。

分别以 8 和 56 为基准进程数,进程数增加为 8 倍后,大数据量实验组的加速比提升为小数据量实验组的 3.78 倍,这主要是因为数据量扩大后合并再分配数据的耗时占比更小了,可见该并行方法的加速比在随着数据量和进程数的增加而提升,也具有良好的弱可扩展性。

## 4 结论

本研究实现了时空客观分析法对 GM 卫星测高数据的海面时变校正的串行程序,分析了 I/O 密集型程序特性。使用按周期分配的并行方案对其进行并行化,提出合并再分配方案以保证负载均衡,并使用 MPI 文件视口函数进行 I/O 并行优化。实验结果表明:与按周期分配方案相比,合并再分配方案在多进程运行时耗时更少,并且在周期数不能被进程数整除时,负载均衡度更高,多进程可扩展性也更好;I/O 并行加速可缩短合并再分配方案的运行时间;I/O 并行加速后的合并再分配方案具有良好的强、弱可扩展性。后续工作可以进一步优化程序的协方差矩阵计算步骤,减少计算环节耗时。

**参考文献:**

[1] ANDERSEN O B,KNUDSEN P. DNSC08 mean sea surface and mean dynamic topography models[J]. 2009,14(C11):1-12.

[2] ZHU C C,GUO J Y,HWANG C,et al. How HY-2A/GM altimeter performs in marine gravity derivation:Assessment in the South China Sea[J]. Geophysical Journal International,2019,219(2):1056-1064.

[3] LE TRAON P Y,NADAL F,DUCET N. An improved mapping method of multisatellite altimeter data[J]. Journal of Atmospheric and Oceanic Technology,1998,15(2):522-534.

[4] YUAN J,GUO J,NIU Y,et al. Mean sea surface model over the sea of Japan determined from multi-satellite altimeter data and tide gauge records[J]. Remote Sensing,2020,12(24):4168.

[5] 刘志鑫,孟小红,王俊,等. 海域重力场解算中垂线偏差方法的并行化改进[J]. 地球物理学进展,2022,37(1):413-420.
LIU Zhixin,MENG Xiaohong,WANG Jun,et al. Parallelization improvement of vertical deviation method in the calculation of sea gravity field[J]. Progress in Geophysics,2022,37(1):413-420.

[6] 干永康. 基于卫星测高的逆 Stokes 反演重力场方法的实现[D]. 北京:中国地质大学(北京),2020:53-54.
GAN Yongkang. Realization of inverse stokes inversion of gravity field based on satellite altimetry data[D]. Beijing:China University of Geosciences,2020:53-54.

[7] FANG L Y,WANG M,LI D R,et al. MOC-based parallel preprocessing of ZY-3 satellite images[J]. IEEE Geoscience & Remote Sensing Letters,2015,12(2):419-423.

[8] SCHENCK W,SAYED S E,FOSZCZYNSKI M,et al. Evaluation and performance modeling of a burst buffersolution[J]. ACM SIGOPS Operating Systems Review,2017,50(2):12-26.

[9] THAKUR R,GROPP W,LUSK E. Data sieving and collective I/O in ROMIO[C]//Frontiers'99:Proceedings of the Seventh Symposium on the Frontiers of Massively Parallel Computation. Annapolis,Feb. 26,1999:182-189.

[10] BEHZAD B,LUU H V T,HUCHETTE J,et al. Taming parallel I/O complexity with auto-tuning[C/OL]//International Conference on High Performance Computing. New York:ACM,2013. DOI:10.1145/250321.2503278.

[11] CHEN Y,WINSLETT M,YONG C,et al. Automatic parallel I/O performance optimization using genetic algorithms[C]// Proceedings of the Seventh International Symposium on High Performance Distributed Computing. Chicago:IEEE,1998:155-162.

[12] GUEDES E A C,SILVA L E T,MACIEL P R M. Performability analysis of I/O bound application on container-based server virtualization cluster[C/OL]//2014 IEEE Symposium on Computers and Communications (ISCC). Funchal:IEEE,Jun. 23-26,2014. DOI:10.1109/ISCC.2014.6912556.

[13] YUAN J,GUO J,ZHU C,et al. SDUST2020 MSS:A global 1×1 mean sea surface model determined from multi-satellite altimetry data[J]. Earth System Science Data,2023,15(1):155-169.

[14] 杨伟光,李文. 使用 MPI 的并行 I/O 实现及性能分析[J]. 计算机工程与应用,2006(17):96-98.
YANG Weiguang,LI Wen. Implementation of parallel I/O using MPI and its performance analysis[J]. Computer Engineering and Applications,2006(17):96-98.

(责任编辑:齐敏华)