

# “并行计算”文献综述

## 一、该领域的研究背景、研究意义和发展脉络

并行计算 (Parallel Computing) 是指通过同时使用多种计算资源从而在时间上或是空间上实现并发解决计算问题的过程。从过去的几十年来看, 得力于不断提升的计算能力, 人类在许多核心领域上有了极大的突破。然而随着工业界对运算量需求的不断增大, 传统串行算法的时间局限性愈加的突出, 为此许多研究者希望能够发明一种计算方式, 使得能够将一个需要大规模运算量的项目分割成若干小规模的项目同时进行运算, 在这样的需求驱动下, “并行算法”这一新兴研究领域应运而生。并行计算的出现, 不仅为许多基础学科领域 (如物理学、化学、生物学、材料学以及计算机科学) 提供了理论计算上的优势, 还大大减少了银行、医药、石油、气象、铁路等大规模工业领域的数据处理时间, 从而使得人们能够以更加高效的方式来进行科学研究和项目实践。

起初, 研究者主要的研究重心主要在于可以通过哪一些方式来实现并行编程的; 在对编程模式有一个较为深入的了解后, 就要开始研究如何进行大规模任务的分配和调度; 随着并行计算理论的不推推进, 基础学科中关于并行算法的研究成果被大量运用到实际的工程项目中, 如将并行计算运用到基因组学研究使其能够对海量基因数据进行快速计算[4]等等。

目前, 并行计算主要被应用在以下三个方面: 对于计算密集型应用 (如大规模的科学工程计算) 而言, 并行计算使得计算机能够在较短的时间内得到预期的结果; 对于数据密集型应用 (如大型数字图书馆, 计算可视化) 而言, 并行计算能够处理这些复杂的模式数据; 而对于网络密集型应用而言 (如远程控制, 远程医疗诊断), 并行计算能够对海量数据进行快速处理以及以较短的时间做出响应。

## 二、目前研究水平、存在问题及可能的原因

“Parallel Computing: Review and Perspective”一文作为一篇综述主要对并行计算的历史发展、编程模式等方面做了一个简要介绍, 指出了目前常见的编程模式有: “主从模式 (Master Slave Pattern)”、“分治模式 (Divide and Conquer)”、“流水线模式 (Pipelining)”以及“推测多线程模式 (Speculative Multithreading)”[1], 同时也指出了通过对具有这些编程模式的单处理机进行耦合, 可以得到具有并行运算功能的多处理机, 最后还简要介绍了并行编程的环境以及常见的编程模型 (如 Data Parallel Model 等)。但是这一篇文献作为综述性质的文献, 且标题中也写到了 Perspective, 但没有明确地提出对这一领域未来发展的展望。

“Study of multiprocessor task scheduling problem in network parallel computing system.”是对网络并行计算中多处理机任务调度问题研究成果的描述, 研究者在文章中针对多处理机任务需求的多步骤调度问题, 给出了“多处理机任务调度模型 (MTS)”、“作业车间调度模型 (JSP)”以及在这两者基础上建立的“多处理机任务作业车间调度模型 (MTJSP)”[6], 文中指出, MTJSP 模型相较于前两种模型而言, 能够保证在较短的时间内得到满意的近优解, 较好地实现解的质量与计算时间之间的平衡。但是这一篇文章的主要问题在于图表过于密集, 以及算法的证明过于粗糙, 对于初入门的读者而言难以快速理解。

“The OpenCL Specification”和“OpenCL: A Parallel Programming Standard for Heterogeneous Computing Systems”两篇文献主要对目前苹果公司推出的 OpenCL 这一工具的编程框架、编程语言、编程模型和 API 等做了一个简要的介绍, 前一篇文章中指出“OpenCL 提供了基于任务分割和数据分割的并行计算机制”[2], 该文中对于编程模型实现的代码没有注释, 加之底层语言的难理解性, 读者较难理解算法的含义。后一篇文章中指出“OpenCL 虽然具有较好的兼容性, 但是其编程语言仍然存在较为底层难以快速上手、不够直观等缺陷”

[10]。然而，针对这样一个难以快速上手、不够直观的编程语言上的缺陷，该文中没有提出可能可行的解决策略。

在研究的过程中，有一些的研究者考虑到不是所有人都会使用电脑和互联网，他们在“Parallel Computing Technology-based Mobile Search Engine”一文中提出了构建“基于并行计算技术的移动搜索引擎”[5]的方法，该文作者指出使用这种技术构建的产品能够使得高处理效率的设备可以给更多的人带来实际效益而不仅仅只是将高效率处理设备的面向对象局限于少数群体。但笔者认为就目前来说，这样的技术普及到现实应用中仍然存在着一定的技术上的难题，而且，不会使用电脑和互联网的群体很可能也不会使用移动设备，该文作者没有就这一方面的内容做一个考虑。

除此之外剩下的参考文献是关于并行计算在工程实践和其他领域中应用的有关文献。文献4介绍了如何将并行计算运用到基因组学研究使其能够对海量基因数据进行快速计算，文中对“要处理太字节和数PB大小的数据，科学家如果使用自己的工作站可能需要数周或数月”这一个问题提出了一个可行的解决策略“应用并行技术和高性能计算（HPC）环境来减少总处理时间并简化对这些数据的管理，处理和分析”[4]同时也对最近发表的涉及基因组学和并行计算的研究的文献进行了系统回顾。这篇文章的问题在于没有对这些系统回顾的文献做一个简要地总结，也没有提出这些文献的可借鉴之处，只是分别对其做了简要介绍。

文献7指出了并行计算在大气科学中的应用。文章中指出“随着能够处理具有精细尺度空间和时间分辨率的模拟的计算资源变得更容易获得，模拟输出的大小继续增长。而输出大小的增加，使得串行数据分析方法变得不堪重负，导致处理过程中的长延迟或由于内存限制导致的完全失败”[7]，针对这样一个问题，可以在大气模拟的大数据分析中使用并行处理的算法，使用消息传递接口（MPI）和Python来并行执行分析。

文献3介绍了一种减少大规模并行计算的延迟的方法，由于在并行运行的云计算工作中，最慢机器上的任务往往会成为完成工作的瓶颈。为了解决这样的问题，作者提出了“添加分散任务的副本并等待最早的副本完成”这一基本策略和“启发式搜索最佳复制”的策略[3]，同时对这种策略的有效性加以实验验证，证明了与MapReduce中使用的复制策略相比，该策略在时间延迟上有着明显的减少。

文献9主要介绍了通过并行计算，可以将实现于单核处理器上的“完全顺序的排序和选择（R&S）程序”，通过“向量填充程序和渐近并行选择程序”[9]这两种顺序过程，使得其能够可以利用多个并行处理器高效地解决大规模的R&S问题。这篇文章的问题主要在于

文献11介绍了并行计算在概念—认知学习（Concept-cognitive learning）中的应用，文章的作者提出了基于“粒度计算和信息融合的大数据以及多源数据”的“概念—认知学习并行计算技术”[11]，并通过了一些数值实验来评估所提出的并行计算算法的有效性。以更好地对形式概念进行分析以及对粒度的计算。

文献8针对目前指数级增长的非结构化数据泛滥的情况，提出了一种“混合存储架构”来存储这些“存储普遍的非结构化数据”，同时该文作者指出“这种混合架构在统一框架内集成了各种类型的数据存储，其中每种类型的非结构化数据都可以找到其合适的放置策略，并且对用户是透明的”[8]此外文章还提出了几种基于统一框架的分区策略，作者通过实验数据来表明，这些策略有利于基于MapReduce的这些非结构化数据的批处理，且可以构建高效智能的系统。

以上几篇文献的内容组织都较为充分，基本上没有出现常见的一些问题。

从这些文献的研究成果来看，目前国内外的研究从以往的单一算法模式研究——即仅仅只是研究如何用编程语言来实现准确高效的并行算法，逐渐过渡到如何将并行算法实践于各个领域之中。这也说明了目前学术界对并行计算的理论研究呈现出了多元化的发展趋势，无论是研究的深度还是广度都较以往的研究有了比较大的提升。

### 三、进一步研究的课题、发展方向概述

对于并行计算而言，进一步可以研究的课题主要有：

(1) 能否找出新的高效准确的并行算法的范式：由于并行计算在真正意义上的发展时间较短，目前理论研究上对于并行算法也没有能够提供较多优良的范式作为算法的参考，加之计算平台这一领域百家争鸣，没有一个统一的标准，导致并行计算的编程模型大多都较为分散，缺乏统一性，因此找到一个高效准确的并行算法范式成为了当下理论研究的一大重心。

(2) 如何保证并行计算时处理机内部能够有效地进行处理机资源的计算任务分配：由于并行算法的多样性，不同的模式会带来不同的效率，算法的执行过程中也可能会出现意外情况（如不同的进程对资源进行争夺，进程之间、不同处理机之间不能进行有效通信），并行计算时系统资源的调度问题是一个值得商讨的问题。

(3) 并行计算时，处理机之间如何进行有效的通讯：由于并行算法的作业流程是先将任务分割成若干个子任务分发出去，再将处理完成的子任务收集回来进行合并处理，那么如果在发送任务的时候或是接收返回的任务时出现“丢包”（即没有成功发送子任务或接收到结果）的情况，究竟该如何处理，即集群中计算机如何有效地发送和接受成功或失败的消息。

(4) 如何有效地对并行算法的代码进行编译优化：由于程序员编程的水准参差不齐，不同的程序员对于相同的一个算法的实现可能会在代码效率之间存在差异，因此可以进一步研究如何在编译时，对并行算法的代码效率进行分析，并做一定的优化处理。

进一步的发展方向有：

(1) 开发便于开发者的并行计算的开发环境：由于并行计算的编程语言仍然不够成熟和系统化，开发环境和开发工具仍然缺乏足够的发展期；而目前苹果公司推出的 OpenCL 虽然具有较好的兼容性，但是其编程语言仍然存在较为底层难以快速上手、不够直观等缺陷。因此可以考虑开发具有优良功能、用户友好的并行计算的集成开发环境。

(2) 开发低延迟的网络接口，使之与高效率并行的处理机相配套：如果网络接口不能为与并行处理机提供低延迟、高效率的通讯，那么很有可能会因为延时问题产生一系列的错误结果。

(3) 将并行计算与当今产业相结合：将先进科学理论与实际生产实践相结合，能够最大程度地创造社会效益。

### 四、见解和感想

在最后想谈一谈个人对“并行计算”的见解。之前也介绍过了，并行计算主要分为“时间上的并行”和“空间上的并行”，而笔者认为，目前所说的并行计算通常都是指“空间上的并行”，因此本文中，主要是对“空间上的并行”这一方式做主要的叙述。为了更好地来区分这两种“并行”，举一个简单的例子，“时间上的并行”好比于一条 CPU 流水线，在同一个时钟内，CPU 既可以将指令从内存中取出，又可以更新程序计数器（PC）的值。而“空间上的并行”主要指在同一个时间内，集群中不同机器分别处理自己的任务，而这些子任务都隶属于同一个任务。

难以想象，在数据量如此庞大的今天，如果没有并行算法的出现，我们能否享受到大数据时代为我们提供的足不出户的快捷和便利，科学家们可能会对几百几千甚至几亿 TB 的数据束手无策。并行算法经过了几十年的发展，仍然还有很大的发展空间，如果能够将其成果广泛地应用在各个方面，那么将会带来一场更大的变革。

### 五、参考文献

[1] Yuxiang Li, Zhiyong Zhang. Parallel Computing: Review and Perspective. Published in: 2018

5th International Conference on Information Science and Control Engineering (ICISCE)

[2] Stone J E , Gohara D , Shi G . OpenCL: A Parallel Programming Standard for Heterogeneous Computing Systems[J]. Computing in Science & Engineering, 2010, 12(3):66-73.

[3] Wang D , Joshi G , Wornell G . Using Straggler Replication to Reduce Latency in Large-scale Parallel Computing (Extended Version)[J]. Acm Sigmetrics Performance Evaluation Review, 2015, 43(3):7-11.

[4] Kary Ocaña, Daniel de Oliveira. National Laboratory of Scientific Computing, Petrópolis, Rio de Janeiro, institute of Computing, Fluminense, Federal University, Niterói, Brazil .

[5] Juefu Liu , Lijuan Ju .Parallel Computing Technology-based Mobile Search Engine. 2010 International Conference On Computer Design And Applications (ICCD 2010)

[6] WANG Meng, FAN Kun, ZHAI Yafei, LI Xinning. Study of multiprocessor task scheduling problem in network parallel computing system. CEA, 2017, 53(10): 264-270.

[7] Sliwinski T S , Kang S L . Applying Parallel Computing Techniques to Analyze Terabyte Atmospheric Boundary Layer Model Outputs[J]. Big Data Research, 2017, 7:31-41.

[8] Lu W , Wang Y , Jiang J , et al. Hybrid storage architecture and efficient MapReduce processing for unstructured data[J]. Parallel Computing, 2017, 69:63-77.

[9] Jun Luo, L. Jeff Hong, Barry L. Nelson, Yang Wu (2015) Fully Sequential Procedures for Large-Scale Ranking-and-Selection Problems in Parallel Computing Environments. Operations Research 63(5):1177-1194.

[10] The OpenCL Specification[S]. Khronos OpenCL Working Group.2014:23.

[11] Jiaojiao Niu, Chenchen Huang, Jinhai Li, Min Fan1. Parallel computing techniques for concept-cognitive learning based on granular computing. Published online: 9 February 2018 Springer-Verlag GmbH Germany, part of Springer Nature 2018.