

基于任务分解推送的计算机集群实现方案

成博韬¹, 曾丽娜^{2*}, 李林²

¹海南师范大学信息科学技术学院, 海南 海口

²海南师范大学物理与电子工程学院, 海南省激光技术与光电功能材料重点实验室, 海南 海口

收稿日期: 2022年6月5日; 录用日期: 2022年7月1日; 发布日期: 2022年7月8日

摘要

随着开发项目体量的扩大, 软件开发类项目组的成员容量逐渐增大, 数据传输与文件交换较为频繁。由于项目开发需要一定的安全性, 在很多的项目中, 一般不会选择一些公开的即时通讯软件。与此同时, 在文件传输过程中, 还需要追求传输的速度和容量, 因此一些网盘类软件也不能很好地满足需求。在一些大数据分析的项目中, 还需要参考计算机的硬件配置进行合理化分配。针对以上问题, 本方案提出了一套任务分解推送措施, 通过账号信息加密功能实现个人及项目安全保护, 使用户可以自定义传输目标, 根据个人需求选择传输空间。针对不同时效性的文件, 系统设置了短期文件处理回收站, 定期清理即时文件, 减少服务器的负担。分析不同配置计算机的性能情况, 给出任务分发的最优解。

关键词

GPU加速, 任务分解算法, 计算机集群

A Computer Cluster Implementation Scheme Based on Task Decomposition and Push

Botao Cheng¹, Lina Zeng^{2*}, Lin Li²

¹College of Information Science and Technology, Hainan Normal University, Haikou Hainan

²Key Laboratory of Laser Technology and Optoelectronic Functional Materials of Hainan Province, College of Physics and Electronic Engineering, Hainan Normal University, Haikou Hainan

Received: Jun. 5th, 2022; accepted: Jul. 1st, 2022; published: Jul. 8th, 2022

Abstract

With the expansion of the development project volume, the member capacity of the software de-

*通讯作者。

velopment project team gradually increases, and the data transmission and file exchange are more frequent. Because the project development needs a certain amount of security, in many projects, generally do not choose some public even communication software. At the same time, in the process of file transmission, but also need to pursue the transmission speed and capacity, so some network disk software can not well meet the needs. In some big data analysis projects, it is also necessary to refer to the computer hardware configuration for rationalization allocation. In view of the above problems, this scheme puts forward a set of task decomposition and push measures, through the account information encryption, the function realizes personal and project security protection, enabling users to customize transmission objectives and choose transfer space according to personal needs. For files with different timeliness, the system has set up a short-term file processing recycle bin to regularly reduce the burden of the server. The performance of different configurations is analyzed and the optimal solution for task distribution is presented.

Keywords

GPU Acceleration, Task Decomposition Algorithm, Computer Cluster

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着大数据、物联网、人工智能等新技术的发展，服务业、交通运输业、制造业等规模不断扩大，信息数据呈爆炸式增长。市场的发展驱动力一方面来源于人工智能市场的迅猛发展，另一方面来源于行业用户加大数据采集力度。从人工智能的角度看，人工智能开发中图像识别占较大比重[1]。图形处理器 GPU (Graphics Processing Unit)在进行图像处理时提供了大量并发执行运算空间。近年来，GPU 的研发速度也在不断加快，通过对此方向的研究，可以从过程上通过算力资源分配对人工智能技术进行过程优化，可以显著提高项目效率，也可以对数据进行有效的保护，加强数据的安全性。

针对文件传输的安全性，传统的解决方法是采用公司内网。基于公司内网搭建协同办公系统，可以实现传统办公模式的部分自动化[2]，实现包括监控网站浏览记录，邮件及其详细内容，软硬件资产管理，非法操作限制报警等，从过程上实现了管理，在信息安全建设[3]方面起到了重要的作用。针对单个 GPU 运算较慢的情况，一种较为有效的方法是搭建 GPU 集群[4]，GPU 集群可以在并行处理计算作业方面提供较为可观的支持。将 GPU 分开部署到多个计算机上，将计算机通过高速网络进行连接，形成一套高速的计算机集群，如日本的“Fugaku”，美国的“Summit”，中国的“神威太湖之光”等。除此之外，算力租赁如：国内的阿里云天池，百度飞桨，国外的谷歌 Colab，都是通过 GPU 租赁实现加速运算。

在本文所提到的方法中，结合了协同办公系统方便接入的特性，采取算力租赁式的积分制算力资源共享，通过计算机性能的分析，实现任务的分解。

2. 基本原理

本方案主要分为：硬件环境搭建、网络环境建设、任务分解推送算法的实现三个部分。

硬件环境搭建方面，为保证计算机集群中单个节点的一致性，将采用统一配置进行计算机组装。除此之外，应当有一台计算机搭载较高配置，方便进行整体管理。服务器的搭建选择公司私有本地服务器，

可配置为整体管理计算机的主机部分。为环境配置和协同调用方便,设备推荐使用相同的系统(如全部使用 Windows10 系统,在 python3.7 版本下运行)。

硬件配置中比较关键的一点是 GPU,但 GPU 的选配可以按需进行,无需过多的追求其一致性和性能。相关实现参考任务分解推送算法的分解部分。针对网络环境建设,本方案前期采取了 Python 的 Django 框架进行网络环境搭建。熟知的网络框架有 Tornado、Flask、Django、Pyramid、Falcon、Bottle 等。不同的网络框架性能不同,功能等也有差距。Tornado 的后端框架接收速度相较之下是最快的,这得益于 Tornado 自带异步特性,底层使用事件 + 循环协程,性能较为强大。Django 的速度相对较慢,这主要是慢在 Django ORM 与数据库的交互。但 Django 网络框架具有部署简单,上手难度低,成本较小的优点[5],与之相比 Tornado 的框架就比较原始了,很多部分需要开发者自己处理。使用 Django 框架进行网络环境搭建[6],将资金更多倾向于公司服务器的搭建,更有利于整体系统的完善。

在应用场景拓展后,将任务分发推送系统开放到互联网,实现整体远程办公等功能。此时需要在安全性等方面做出一定的改进,以规避信息泄露的风险。

任务分解推送算法,主要通过大数据及计算机配件官方参数进行函数描述,生成一套给不同型号计算机分配任务额度的分配方案。当算法被执行时,用户导入参与任务的计算机配置,或计算机安装的自检测插件自动检测并返回到主控计算机,调用函数进行运算。运算结构以 Log 的形式输出,并写出建议配额,经任务发送方和任务接收方共同确认后分发,如图 1 所示。

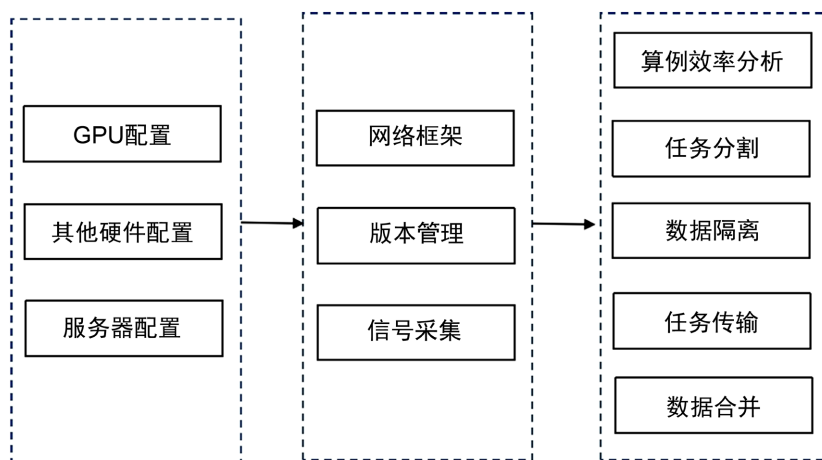


Figure 1. Scheme implementation process

图 1. 方案实现流程

3. 设计方案

3.1. 应用系统原理

本论文前端采用 VUE 框架,后端采用 Django 框架,可以更好的进行前后端的数据交互,后端 django 具有强大的数据库访问组件。Django 自带一个面向对象的、反映数据模型(以 Python 类的形式定义)与关系型数据库间的映射关系的映射器(ORM),使用户可以更轻松更便捷的访问数据库同时 Django 具有健全的后台管理系统,通过管理员可以更轻松地进行网站的维护。同时 Django 部署在 Nginx 会更加兼容,使服务器效率更高。前端采用 vue 框架,Vue 采用是一个 MVVM 框架,可以快速的进行数据之间的绑定,并且进行前端网站的快速搭建,同时 Vue 使用基于依赖追踪的观察系统并且使用异步队列更新,所有的数据都是独立触发的,提高了数据处理能力。第一次使用系统时需要先进行账号的注册,注册登录成功

后将会跳转首页并引导用户进行项目的创建,成功创建项目后,将会给该项目创建人分配一个邀请码,以便项目创建人通过此邀请码邀请其他用户参与进此项目,同时用户也需要进行填写自己电脑的配置信息,当用户将数据集进行上传后,本网站会通过自制的搜索引擎,寻找一台合适的主服务器,同时系统会通过独创数据分割算法[6]和分布式技术[7]将数据分配给不同的主机进行训练,并采用 CAP 原理使保证数据的稳定性。分配完数据集后,采用联邦学习算法进行不同主机之间与主服务器进行数据交互,这样充分保证了训练数据集的速度及稳定性,最后训练完成之后,将模型返回给用户。

3.2. 任务分解推送算法

计算机的运算速度和数据传输速率很大程度上受到了硬件设备的影响,选择合适的硬件设备,可以大大提高计算机的使用效率。NVIDA 官方说明,当 GPU 计算能力大于 5.0 时,适合用来进行神经网络运算。显存提高, batch size 相较之下更大,直接意味着性能更加强大, CUDA 核可以更接近满负荷的工作。从 CUDA 核心数和功率来看,单块 RTX 3090 的核心数为 10,496,功率为 350 W; RTX 2080 核心数为 2944,功率为 215 W。因此,相同功耗下, RTX3090 的核心数约为 RTX2080 的 2.18 倍,意味着相同时间内搭载高配置显卡的计算机可以承担更大的运算量。任务分解推送算法据此进行搭建。

3.3. 任务分解算法与流程实现

对于任务的分解算法,本文提供“同时结束任务分解”实现方法,具体实现思路如下:

设终端为 PC,总控终端为 PC0,则参与终端从 PC1 开始计数到 PCn。以总控终端作为参考进行任务分解,设总控终端 GPU 加速下的处理速度为标准处理速度(此处的标准仅指算法集合内的相对标准)为 v_0 ,根据 v_0 得其余终端处理速度为 $v_1 \cdots v_n$,处理效率 m 由标准处理速度与各处理速度比例得来,即 $m_1 = v_1/v_0$; $m_2 = v_2/v_0 \cdots$ 设任务总量为 s ,理想状态下,任务完成时间 $t_{\text{总}}$ 为 $s/v +$ 任务整理时间 t' 。在应用任务分解时,还需要加上任务分解时间,任务发放时间,任务回传时间,任务组装时间,用大写 T 来表示。

同时结束任务分解算法:同时结束分解算法的目的是让各终端的结束时间相同,即需要对 s 进行分解。此算法是较为简单的算法,只需要按照处理效率分配各终端的任务量即可,最终的任务时间为:

$$t_{\text{总}} = t' + T + \frac{s}{\sum_{i=0}^n v_i}$$

其中 T 表示 the time of Disassembly, distribution, return, assembly。在任务时间中,单次任务量是确定的,此刻影响任务完成速度的因素有每一个终端的处理速度,任务分解、发放、回传、组装等时间。在确定设备的情况下,网络与传输算法直接影响了任务的传输,进而影响任务效率。网络属于硬件功能,不作深入讨论。针对传输算法,本文的解决方案是通过将一些数据以 JSON 的形式传输,压缩数据体积,相对加快传输效率。

3.4. 节点之间的信息交流

信息交流平台: kubernetes 1.18.0,在 kubernetes 1.18.0 中部署的 Nginx 1.12.0。选择使用 Nginx 服务器而不是传统的 web 服务器,前者在短时间内的占用线程后,就直接的交予相应的模块进行处理,而后者则一直占用进程资源。所以 Nginx 服务器的模块化设计可以使得网络性能、用户感知的请求时延都得到大大提升[8]。

传输媒介是一个共享的存储卷。一个容器内使用一个共享的存储卷来简单高效的地在容器间共享数据。大多数情况下,使用一个共享目录在 pod 里的不同容器间共享数据就够了。一个标准的 pod 内容器共享存储卷的用例是一个容器往共享存储卷里写入数据,其它的则从共享目录里读取数据,达到基本的

信息交流和传输。Kubernetes 将集群中的设备划分为一个 Master 节点和一群 Worker 节点(Node 节点),这和 Nginx 的工作模式不谋而合。所以因为这种 Master-Worker 工作进程模式,可以使得多个 worker 节点工作进程之间实现负载均衡,实现数据的高效处理和传输。

4. 总结

本文所实现的基于任务分解的数据推送算法,通过 Django 搭建局域网环境,结合 kubernetes 作为信息交流平台,实现了针对机器学习的算力资源分配。使得使用者可以在低成本,多终端的环境下实现机器学习及相关数据分析的提升。结合联邦学习的方法对集群进行测试,通过任务分解算法实现的“K8sPlus”,其效率在 CIFAR-10 数据集中相对于普通联邦学习算法较低,但在 CIFAR-100 的训练中达到了较好的成果。在较低的经济成本下,通过接入更多的终端进行任务分发,从而加快运算速度,可以让每个终端的负载较低,有利于设备的维护。

基金项目

海南省高等学校科学研究项目(Hnky2020ZD-12, Hnky2020-24),海南省自然科学基金(2019RC190, 2019RC192, 120MS031, 622RC671),大学生创新训练项目(202111658013)、大学生创新创业开放基金(202111658021X, 202111658022X, 202111658023X)和海南省院士创新平台科研专项资金(YSPTZX202127)资助。

参考文献

- [1] Ukita, J. (2020) Causal Importance of Low-Level Feature Selectivity for Generalization in Image Recognition. *Neural Networks*, **125**, 185-193. <https://doi.org/10.1016/j.neunet.2020.02.009>
- [2] 曹义新. 协同办公系统在企业中的设计与应用[J]. 计算机系统应用, 2013, 22(6): 39-43+94.
- [3] Kathi, M.G. and Shaik, J.H. (2021) An Optimised Allotment and Tracking Using Django and Opencv. *Ingénierie des Systèmes d'Information*, **26**, 201-209. <https://doi.org/10.18280/isi.260207>
- [4] Li, Z.J., Chang, V., Hu, H.Y., Fu, M.Z., Ge, J.D. and Piccialli, F. (2021) Optimizing Makespan and Resource Utilization for Multi-DNN Training in GPU Cluster. *Future Generation Computer Systems*, **125**, 206-220. <https://doi.org/10.1016/j.future.2021.06.021>
- [5] 孙建军, 李琪, 吕强. 浅析 Web 开发工具 Django 的 MVC 架构[J]. 品牌与标准化, 2021(6): 105-106+109.
- [6] 刘炜, 唐琮轲, 马杰, 刘宇昭, 田钊, 余维. 区块链在隐私计算中的应用研究进展[J/OL]. 郑州大学学报(理学版), 2022: 1-12. <https://doi.org/10.13705/j.issn.1671-6841.2022109>, 2022-06-09.
- [7] 彬建. 主机应用下移——分布式技术应用实践[J]. 金融电子化, 2018(12): 82-84.
- [8] 苗鑫. 基于 Nginx 的智能多媒体传输系统设计与实现[D]: [硕士学位论文]. 大连: 大连理工大学, 2018.