计算机数据处理技术研究

王善厚
（郓城县双桥镇中学 山东 菏泽 274711）

摘 要 在计算机数据处理中，面对海量数据的处理要求，需寻找一种更加精准、有效的数据处理方法，来降低数据失真等问题的发生率。文中立足于大数据处理条件，以爬虫技术为研究对象，设计了一套完整的计算机数据处理技术，在阐述其技术要点的同时，对实验测试结果展开了分析。测试结果显示，爬虫技术可保证数据完整性且数据抓取效率较高，具有推广价值。

关键词： 计算机数据；爬虫技术；抓取率
中图分类号 TP274.2

Research on Computer Data Processing Technology

WANG Shanhou
（Yuncheng County Shuangqiao Town Middle School,Heze,Shandong 274711,China）

Abstract In computer data processing,in the face of the processing requirements of massive data,a more accurate and effective data processing method should be found to reduce the incidence of data distortion and other problems.Based on the conditions of big data processing,this paper designs a complete set of computer data processing technology with crawler technology as the research object.While expounding its technical points,the experimental test results of crawler technology are analyzed.The test results show that crawler technology can ensure data integrity and high data capture efficiency,and has popularization value.

Key words Computer data,Reptilian technology,Grabbing rate

0 引言

近年来，得益于大数据等现代计算机技术的发展，网络空间中的信息量快速增多，传统数据处理方案已经无法应对新的数据处理要求。在这一背景下，具有更高精准度、灵敏度的爬虫技术开始快速发展，并取得了令人满意的效果。但快速发展的爬虫技术也存在弊端，以分布式爬虫为例，该技术的可定制性差，需要投入大量的精力用于技术维护。为解决上述问题，本文提出了一种新的计算机爬虫数据处理技术。

1 计算机数据处理技术的要点分析

1.1 爬虫技术的功能需求

1.1.1 爬虫集群服务

在本次设计中，爬虫集群服务是在大数据模式下实现的，其关键技术服务内容包括以下几点。(1)用户通过爬虫服务模式可完成数据采集、处理等操作，可根据数据处理要求自定义处理任务，用户也可以按照集群伸缩服务与业务需求选择爬虫节点。(2)爬虫在执行数据处理任务的过程中，可生成诸多抓取任务，由爬虫节点依次完成，减少用户去冗余等一系列操作，提高数据处理效率[1]。(3)在爬虫伸缩服务中，可添加任务节点来强化数据处理能力(也可以降低数据处理成本)，并在计算机数据处理中不考虑爬虫状态。该方法可自动完成数据变化带来的新抓取任务与权重分配，快速适应爬虫节点的变化。

1.1.2 控制爬虫集群

在本次数据处理中，可按照自身意愿控制爬虫的行进过程，其关键在于以下几点。(1)在计算机上创建爬虫集群后，可根据自己意愿控制爬虫的运行过程，爬虫系统也可在已定的爬虫节点中部署新的爬虫数据集，降低后期系统维护难度。(2)用户可以根据使用需求设定爬虫集群任务，也可解决用户处理爬虫时的资源回收问题，提升数据资源处理效率。(3)用户可根据数据处理需求选择订阅爬虫集群，并在系统中核对爬虫节点信息，提高系统的响应效率。

1.1.3 爬虫数据处理任务调度方案

在计算机数据处理中，爬虫系统的主要功能是在数据采集任务中快速、明确地抓取任务。为实现这一目标，需要选择一种科学的任务调度算法，保障爬虫节点可通过处理时间差缩短任务执行时间，提升整体数据运行效率。在综

作者简介：王善厚(1986—)，专科，研究方向为计算机。

合比较不同任务调度方案的合理性后，本文决定应用加权轮询调度算法，该方法的主要特征是可以将数据处理任务随机（或指定）分配至爬虫节点上，并兼顾爬虫节点的数据处理能力，最终确定任务分配比例赋值，保障每个爬虫节点上的任务数量理想化[2]。

1.2 爬虫控制机制

1.2.1 爬虫控制流程

在计算机数据处理中，爬虫节点作为独立的运行个体，可支持同步控制与非同步控制两种模式。其中，在同步控制机制下，操作者可通过控制节点发送控制指令，也可等待节点反馈消息，在采集反馈信息后发送控制指令，此时控制节点也可向操作者展示数据处理结果。而在非同步控制模式下，控制节点向所有被控节点发送指令后，立即向操作者显示被控节点的状态，系统也可利用控制节点反馈监控信息。基于上述控制机制，爬虫系统可通过诸多爬虫节点快速响应用户操作指令，解决响应时间不可预知的问题。

在爬虫数据处理中，基于控制指令完成集中爬虫节点操作指令即可，按照上述控制机制，爬虫可利用控制节点接收用户的操控指令，从第三方存储组件获取各个爬虫节点汇报的信息，并将其展示给用户。当爬虫完成操作后，会继续查看控制指令是否顺利完成，再根据操作指令完成信息更新，此时对于已经完成的数据处理指令，会自动删除执行结束的控制指令[3]。

考虑到控制指令在执行阶段存在时间序列要求，可采用redis的第三方存储组件来设置任务处理队列流程。但redis只支持“字符串”格式，因此在时间序列中，也可采用相同的数据结构，将其转换为条件相同的字符串。

1.2.2 爬虫控制指令

为保证数据处理效果，可采用节点控制指令模式，利用爬虫任务对应的ID，分别完成数据处理任务。其对应的控制指令如表1所列。

表1 爬虫控制指令分类方案

控制指令分类	任务内容描述
开始任务	启动爬虫数据处理任务并开始执行
结束任务	停止爬虫节点的数据采集任务
添加任务	将数据库中的数据加载任务添加到控制指令上
更新任务	实时更新爬虫数据处理方案
删除任务	删除爬虫的数据采集任务

在表1所列的爬虫控制指令的基础上，该系统可通过提供多功能接口完成数据交互，且该系统可采用“etcd”数据处理模式存储数据，并根据不同的数据处理路径更新存储区域，最终形成爬虫数据处理的绝对路径。

1.3 爬虫实现方案

1.3.1 节点管理模块的实现

在本文所提方案中，节点管理模块是主要的功能载体。本次设计将按照节点管理方案，主动从“etcd”集群中获取爬虫的信息资料，再将任务节点更新到对应的控制路径。同时，按照爬虫控制机制节点设计方法，该系统可自动执行“redis”存储单元的控制指令。

1.3.2 任务列队模块的实现

在任务列队模块中，可利用爬虫节点抓取任务链，该任务链应最大限度地保证任务数据链的完整性。在综合考虑计算机数据处理任务设定方案的基础上，将通过快速迁移的爬虫节点实现数据的快速处理与更换，并支持未抓取任务列队模块的迁移处理要求。

为实现上述要求，本文通过计算与存储分离原理，将从任务列队中抓取的存储任务添加至“redis”存储单元，使数据处理爬虫可在不保留本地控制任务的基础上实现数据更新，降低爬虫数据处理难度。同时，考虑到数据量增加可能会造成访问延迟增加的问题，因此爬虫可通过预先读取部分数据的方式，提升数据更新能力。基于爬虫节点的任务数据更新要求，可通过线性队列结构将其缓存至本地系统中，降低数据处理延迟。

1.3.3 任务调度解决方案

根据爬虫技术的任务调度机制，在解决任意一个节点的数据预处理方案后，每个节点负载均可分配到对应的控制节点中。假设爬虫数据处理的阶段组数据长度为n，存活节点数组长度为m，总分配任务为T，根据谷歌一致性哈希算法，每个爬虫对应的存活节点组数分配任务如式（1）所示：

$$g = \frac{T(n - m)}{n} \tag{1}$$

其中，g表示爬虫技术中任务节点的分配任务数量。

为保证所有爬虫在数据处理中均可完整进行数据更新，在保证全部节点组数满足g的情况下，存活节点数量为ratio时，则存活数组节点组合方案为[g*ratio]。

1.4 基于爬虫技术的系统实现

基于大数据的计算机数据处理系统主要分为两个部分，分别是管理子系统以及爬虫子系统。其中，爬虫子系统可为计算机大数据处理中的用户提供具有伸缩功能的爬虫集群服务，并将其与采集任务匹配起来[4]。

在爬虫部分的设定上，可通过爬虫任务模块、爬虫集群监视结构、集群控制等功能模块提供爬虫控制的详细操作，包括创设爬虫数据处理节点、启动或停止、爬虫的创建与扩容等。

2 系统测试方案

2.1 测试环境与测试内容

本次系统测试环境为计算机端，其CPU为2.4 GHz/4核，内存4 GB、网卡为千兆以太网卡。测试内容包括主要有3点。（1）爬虫集群测试，以了解爬虫集群的数据处理能力及数据抓取的精准度。（2）对爬虫数据处理的详细功能展

开测试，验证其数据创建、删除、查看状态等功能。(3)在非功能性测试中，需要了解爬虫数据处理的完整度，计算爬虫节点抓取率及数据完整性情况，检查经过爬虫处理后的数据是否存在数据丢失的问题。

2.2 功能测试内容

2.2.1 爬虫集群服务测试方案

(1)爬虫数据处理任务测试

本次测试的目的是了解爬虫技术能否正常完成数据采集任务，整个测试的操作流程如下。

1)在计算机上新建爬虫集群，并启动任意一次数据采集任务。

2)启动爬虫集群的数据处理任务。

3)自动分配任务，将数据采集任务匹配至指定的爬虫集群。

4)由爬虫自动完成数据处理任务。

5)检查数据库中有无最终的结果数据。

通过现场测试后，现场测试结果与预期结果一致，通过比较数据采集前后的数据变化，发现在数据库中存在结果数据，与预期相同。

(2)爬虫添加节点测试

本次测试的目的是判断爬虫在数据处理中能否快速完成新节点添加等操作，测试操作流程如下。

1)创建新的爬虫集群，并订购任意节点。

2)启动爬虫集群任务处理程序。

3)在计算机上点击订购节点，并随机添加新节点。

4)检查是否正常添加了新节点。

按照上述方法展开现场测试后，结果显示爬虫集群可顺利新添节点。

2.2.2 爬虫集群控制测试

开展爬虫集群控制测试，是为了检验爬虫能否根据用户指令完成个性化处理操作。以爬虫状态更新检测为例，整个检测过程的步骤如下。

(1)用户登录系统，新增爬虫集群。

(2)点击新创建的爬虫集群。

(3)检查在计算机上是否正常更新爬虫集群状态，或爬虫是否可按照用户指令运动。

测试结果显示，用户可在计算机上查看爬虫集群状态，爬虫集群也可根据用户控制指令实现集群控制。

2.3 非功能性测试内容

2.3.1 性能测试

在爬虫数据处理中，性能测试的目的是了解爬虫集群的性能差异，通过比较爬虫在不同节点完成数据处理任务的情况，评估计算机处理技术的可行性。具体测试流程如下。

(1)创建5个爬虫集群，并为对应的爬虫集群分配数量相同的任务节点。

(2)启动任务，由每个爬虫集群自行完成数据采集。

(3)观察每组爬虫完成任务的时间。

按照上述测试步骤，节点数量越多、爬虫完成任务的时间越短，则证明其数据处理效率更高。根据现场测试结果发现，节点数量越多的爬虫集群完成任务的时间越短，且节点数量与完成同一任务的时间接近反比折线。根据该测试结果可以判断，在数据处理中合理增加节点数量，可显著提升爬虫的数据处理效率。

2.3.2 数据抓取率测试

本次测试的目的是验证不同规模爬虫数据集在页面数据抓取中的速率变化情况，测试步骤如下。

(1)创设新的抓取页面，确保其中有充足的数据抓取任务。

(2)随机创设5个爬虫集群，并分配数量相同的爬虫节点。

(3)由爬虫集群自行完成数据采集任务。

(4)每隔特定的时间记录集群抓取的数据数量变化情况，并计算每15 s的节点平均抓取速率。

在上述数据抓取流程测试中，应尽量确保爬虫集群总抓取速率随着集群节点个数的增加而增加，而现场实际测试结果也证实了上述目标的实现，相关测试结果如表2所列。

表2 爬虫数据抓取率测试结果

集群节点数量	1	2	3	4	5
数据抓取速率（页面数/分钟）	142.6	184.9	241.3	443.5	502.7

2.3.3 数据完整性测试

数据完整性测试直接关系到数据处理质量，其目的是判断爬虫技术在数据处理中是否会出现资料丢失的情况，测试流程如下。

(1)创设一个总抓取量为10 000条数据的采集任务。

(2)创设两个爬虫群，分别为测试集群与正常集群，并为集群各分配5个测试节点，由爬虫自行完成采集任务。

(3)执行至特定时间节点后关闭一个节点，重复上述操作，直至所有节点均关闭。

(4)统计两个集群中爬虫的数据抓取数量变化情况。

在测试中，数据抓取量越接近10 000就表明数据抓取数量越多。现场测试结果显示，正常爬虫集群和测试爬虫集群的数据抓取数量分别为10 000、10 001。经分析可知，测试爬虫集群之所以抓取数量达到了10 001，是因为在强制停止节点时，任务节点模块并未将数据库中的信息从redis任务列队中删除，导致爬虫重复执行了该任务，但经过数据处理校正后，多余数据已经消除。

3 结语

本文提出的计算机爬虫技术在操作上具有先进性，操作难度较低且效果显著。现场测试结果显示，本文提出的系统测试方案在爬虫集群服务测试、爬虫集群控制测试等方面具有优势，且数据抓取率和完整性更高。这证明本次

(下转第273页)