第34卷 第23期 计 算 机 工 程 2008年12月
Vol.34 No.23 Computer Engineering December 2008
·软件技术与数据库· 文章编号：1000—3428(2008)23—0076—03 文献标识码：A 中图分类号：TP301.6

分布式计算中可靠的数据放置方法

汤小春，胡 杰
(西北工业大学计算机学院，西安 710072)

摘 要：数据放置活动在以计算为主的分布式计算中被看作是次要的任务。文章将数据放置工作与计算工作区别对待，定义分布式计算过程中的数据放置模型，给出数据放置协议和可靠的数据传输算法，使得数据放置活动在分布式计算中像计算工作一样被排列、调度、监控、管理以及检测。对数据放置方法进行了系统的评价，取得了较好的效果。

关键词：数据放置；可靠传输；分布式计算

Reliable Data Placement Method for Distributed Computing Environment

TANG Xiao-chun, HU Jie
(Computer College, Northwestern Polytechnical University, Xi'an 710072)

【Abstract】 Today scientific applications on distributed computing environment have huge data transfer which continues to increase drastically every year. This implies a major necessity to move huge amounts of data from original data site to target site on the whole computation cycle, which brings with it the problem of efficient and reliable data placement. The current approach to solve this problem of data placement is either doing it manually, or employing simple scripts which do not have any automation or fault tolerance capabilities. The goal is to make data placement activities robust and efficient. It will be queued, scheduled, monitored, managed, and even check-pointed. The data placement activities should be treated differently from computational jobs, since they may have different semantics and different characteristics. The method for data placement is tested.

【Key words】 data placement; reliable transfer; distributed computing

1 概述

随着科学应用对数据需求的迅速增加，计算过程中数据放置活动也必须被仔细地调度和管理。对数据密集型应用来说，数据的访问特性是分布式计算过程中的一个潜在的瓶颈。

分布式计算给研究者带来巨大资源的同时，也带来了挑战[1]。为了有效利用分布式资源，研究者必须解决有关数据放置的挑战。(1)分布式是一个异构的环境，有很多不同的存储系统、数据转移中间件和协议共存；(2)分布式会带来失败的网络连接，传输中实施的多样化以及冲突的用户、服务器和存储系统；(3)不同的工作可能有不同的策略和优先级；(4)一个应用访问的网络和存储资源可能是有限的，因此必须有效地使用它们。以前，分布式计算中的数据放置活动由手工或简单的脚本执行[2]。数据放置活动在以计算为主的分布式计算中被看作是次要的。最近，越来越多的人将数据放置工作与计算工作同等对待起来，提出了许多可靠的数据传输协议：一种是失败恢复技术[3]，即转移一个大的文件失败了，只转移文件中没有转移的那部分；另一种是提高数据传输的服务质量[4]。然而，前者仅仅对于数据传输过程进行了分析，对于数据的其他过程没有涉及到；而后者主要从效率角度考虑，目的只是提高传输的质量。这些方法在处理单个数据的传输时，能够很好地工作，但是，对于分布式计算过程的数据放置的整个过程，无法理解数据放置调度器中的数据转移语义，特别对于传输过程中出现的过载现象以及传输过程的动态变化特性(如某种协议突然失效等)处理不足。

鉴于以上问题，本文提出了一种将数据放置工作和计算工作区分开的方法。数据放置工作被提交到一个能调度和管理数据放置工作的调度器上，计算工作被提交到一个能调度和管理计算工作的调度器上。这使得数据能够被排队、调度、监视、管理以及失败恢复，通过在多种传输协议中的转换以及传输过程的优化，达到可靠、高效的数据转送。

2 相关数据放置过程定义

很多分布式中的应用需要从远程站点移动输入数据到执行站点，然后从执行站点移动输出数据到原来的或另一个远程站点。图1给出一个作业在远程节点的执行模型。作业在远程节点执行时，一般包含如下的数据放置过程：

定义 1(空间分配) 分配启动、监测点以及输出数据需要的空间(图1中a)。

定义 2(启动数据) 作业的可执行文件以及环境变量从原节点转移到目标节点，或者作业由于中断而重新执行时从原节点转送检测点数据的到目标节点的过程(图1中b)。

定义 3(检测点数据) 作业在执行节点执行时，用于将检测点文件转送到原节点的过程，即完成一次检测点保存(图1中c)。

定义 4(结果数据) 执行节点将执行的结果文件返回到原节点的过程(图1中d)。

定义 5(存储空间释放) 作业执行结束或者迁移到其他节

作者简介：汤小春(1969－)，男，副教授、博士，主研方向：软件开发环境，分布/并行处理技术；胡 杰，硕士研究生

收稿日期：2008-03-24 **E-mail：**tangxc@nwpu.odu.cn

点后，释放所占用的存储空间的过程。图 1 中 e 表示释放启动数据空间；f 表示释放检测点数据空间；g 表示释放结果数据空间。

把这些计算和数据传输的步骤看作真正的任务，用有向无环图(DAG)来代表它们，方向弧代表它们之间的依赖关系。

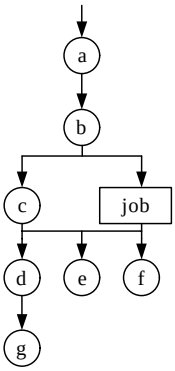

图 1　分布式计算环境下的数据放置模型

在图 1 中，椭圆表示数据作业，方框表示执行作业。将数据和执行作业分离后，它允许用户同时调度 CPU 和存储资源。将 DAG 流的数据放置工作提交到数据放置调度器。这样，计算作业在执行前，就可以确定计算所需的输入文件是否到达一个接近执行节点的存储设备。类似的，在计算结束后，输出文件可以被移动到一个远程存储系统。

3　数据放置模型

3.1　分布式环境下数据放置协议模型

对任何分布式计算来说，数据放置是其中的一个关键部分。分布式计算中的数据处理过程不再是单个节点的 I/O 子系统的任务，而是全局的数据放置过程。图 2 给出了一个分布式计算过程中的数据放置协议。

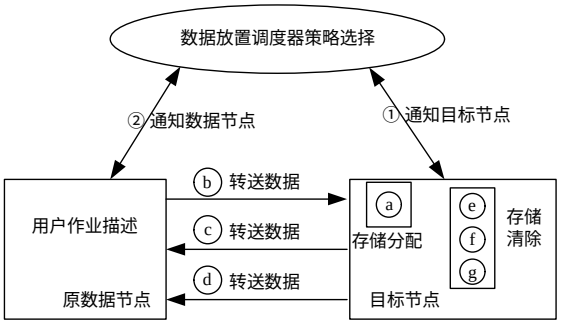

图 2　数据放置协议

图 2 中的①和②是数据存放节点与目标节点之间的消息交换。分布式系统通过资源的查找算法[5]将作业映射到合适的资源后，用户的作业在远程节点上执行前，数据的放置就必须作为首要任务开始考虑。其调度协议如下：

(1)作业提交后，数据放置调度器收到数据大小值(图 2 中①)。

(2)调度器通知目标节点按照数据大小值分配存储空间(图 2 中①)。

(3)目标节点检查存储空间大小(图 2 中 a)，并返回成功或不成功(图 2 中①)。

(4)不成功，请求调度器执行选择策略；成功，调度器通知原数据节点(图 2 中②)。

(5)从原节点向目标节点转送数据(图 2 中 b)。

(6)作业在目标节点执行过程中，向原数据节点发送检测点数据(图 2 中 c)。

(7)作业执行结束，调度器通知原数据节点接收结果数据(图中②)，并发送结果数据(图 2 中 d)。

(8)结果文件转送完毕，调度器经过判断后，认为目标节点的数据不再需要，就执行清除活动(图 2 中 e,f,g)。

数据放置调度器可以有效地控制数据的可靠性、异构支持以及适应性等。

3.2　异构存储资源下的数据放置

在不同的存储系统、数据传输协议或中间件中，增加数据放置调度支持是很直接的。数据放置调度器可以支持异构的存储系统、多种数据传输协议。用户可以立即使用它们而不需要做额外的工作。数据放置调度器可以与诸如 FTP, GridFTP, HTTP 和 DiskRouter 等的数据传输协议联合使用；也可以与数据存储系统 SRB, UniTree 和 NeST 以及数据管理中间件 SRM 进行实时的交互。

3.3　运行时的适应性

数据放置调度器可以动态并自动地在运行时间里决定使用哪个数据传输协议用于相应的传输。在传输之前，调度器做一个快速的检测，确定源和目的主机之间哪些协议可以使用。如果用户指定的源和目的 URL 允许的协议不能够实施传输，调度器将利用自己协议库中的协议来实施传输。用户也可以不指定任何协议，让调度器来决定使用哪个协议。数据放置调度器自己的协议库内容定义如下：

```
host_name = "tangxc.co-think.com";
supported_protocols = "http, gridftp, ftp";
```

用户的无执行协议的定义如下：

```
dap_type = "transfer";
src_url="any://www.nwpu.edu.cn/jc/input.dat";
dest_url="any://www.co-think.com/jc/input.dat";
```

3.4　故障恢复和有效利用资源

调度器对用户应用隐藏了网络、存储系统、中间件或软件等各种类型的故障。它有一个“重试”机制，在返回故障前，它可以重试任意给定次的数据放置工作。它还有一个“终止和重启”机制，允许用户为他的数据放置工作指定一个“最大允许运行时间”。当一个工作执行时间超过这个时间时，将会被调度器自动终止并重启。这个特性覆盖了一些导致传输一直挂起和从来不返回的系统中的错误。

调度器可以控制同时访问任意存储系统的请求的数目，并确保该存储系统和连接到该存储系统的网络都不过载，还可以实施空间分配和取消分配以及支持存储系统空间预留。

4　可靠的数据传输调度算法

在数据放置调度器中，将数据传输分为 3 个阶段：第 1 阶段是决策阶段，数据放置调度器使用自身包含的协议在目标节点与原数据节点之间进行测试，确定一个最佳的传输协议和最大的传输线程数量；第 2 阶段是数据的传输，对于传输过程中的失败进行重试和设置超时处理；第 3 阶段伴随着第 1、第 2 阶段进行，主要是进行适应性调整，防止出现过载或者死锁现象。具体算法如下：

S1　解析用户数据描述，取得原节点与目标节点。初始化失败重试次数和超时。

S2　在目标节点分配必需的存储空间，如果不够，等待。

START：

S3　循环选择调度器支持的协议库中的每个协议。

S3.1　测试该协议是否可以在原节点与目标节点之间可用；若不可用，转 S3。

S3.2　使用测试包测试数据传输时间，记下协议名与时间(X,T)；转 S3。

S3.3　采用 socket 通信，执行 S3.1。

S4　对测试的集合{(X,T)}进行再次测试，对每个协议得到一个最好的传输线程数量 P。

TRANS：

S5　从集合{(X,T)}中选择一个 T 最小的协议。

S5.1　传输数据，并记录每包响应时间 W。

S5.2　如果传输中出现失败，执行 S5.3。

S5.3　如果重试次数>0，重试次数减 1，转向 S5.1。

S5.4　超时到达仍传输失败，将该协议从集合{(X,T)}删除；转向 S5。

S5.5　如果集合{(X,T)}不为空，执行 S5；否则，错误退出。

TUNING：

S6　调度器定期发送测试包，并取得响应时间。

S6.1　若响应时间超过以前的 T 值，该协议的 P-1。

S6.2　若 P 小于 2，通知 TRANS。

S7　清除临时文件，退出。

5　试验评价

5.1　可靠数据转送算法测试

使用 2 个NEC的磁盘[5]阵列，一台磁盘阵列位于www.co-think.com，另一台位于www.nwpu.edu.cn，2 台Linux服务器分别对应各自的磁盘阵列。测试的任务是将科信公司(co-think)的磁盘阵列上的 10 万个文件转送到西工大的磁盘阵列上。测试过程是：科信公司的服务器读取磁盘阵列上的文件，然后使用不同的传输方式将文件转送到西工大的服务器上，再由该服务器保存文件，形成一个数据管道。测试数据如表 1 所示。

表 1　转送过程的可靠性

使用的传输协议	失败文件数	可靠性/(%)
FTP	198	98.02
HTTP	210	97.90
GridFTP	203	97.97
可靠数据转送算法	10	99.90

在测试中，重试的次数为 3，超时为 1 min。从表 1 可以看出，FTP, HTTP, GridFTP 的可靠性基本相当，而采用可靠数据转送算法后，可靠性可以提高 2 个百分点。

当重试的次数设置为 6，超时设置为 3 min，对上述文件再次进行传输时，测试 10 次，数据转送的可靠性可以达到 99.999%。因此，该算法可以达到数据转送的高可靠性。

5.2　分布式批处理中的测试

在一个分布式批处理管理系统[5]中，采用可靠的数据放置方法以及socket直接从原节点与目前节点之间转送数据。设置作业提交规则为每秒钟提交 3 个作业，连续提交 24 h，作业执行任务是拷贝原文件为一个新的文件。

```
{
set JOBLOG="%JOBLOG_PATH%\%~n0.log"
iSMrc_replicate -file "%RCF%" -wait >> %JOBLOG% 2>>&1
if not %ERRORLEVEL%==0 goto ABEND
:NREND
echo %DATE% %TIME%   %~n0 Normal End >> %JOBLOG%
exit 0
:ABEND
echo %DATE% %TIME% %~n0 Abnormal End [ RC = %ERRORLEVEL% ] >> %JOBLOG%
exit %ERRORLEVEL%
}
```

统计以上连续执行中出现的错误作业的数量(复制过程错误不计入)。采用可靠数据放置方法其结果如图 3 所示，直接采用 socket 方法其结果如图 4 所示。

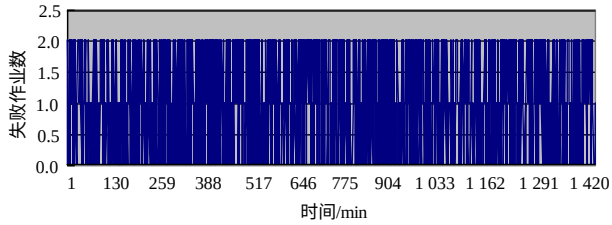

图 3　采用可靠数据放置方法批处理作业执行状态

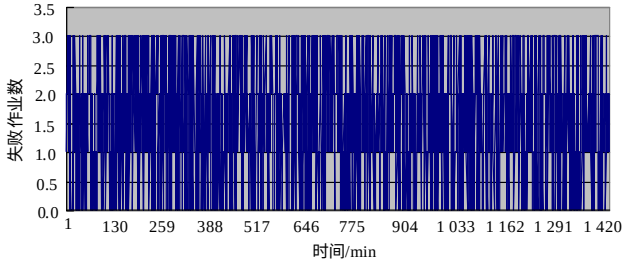

图 4　直接采用 socket 传输数据方法批处理作业执行状态

从图 3 和图 4 可以看出，采用可靠数据放置方法后，作业执行失败的可能性降低了 40%左右。因此，分布式计算中可靠数据放置方法提高了分布式计算系统的可靠性和健壮性。

6　结束语

在分布式计算中使用可靠数据放置方法后，缓解了由于数据处理异常而导致的作业执行效率下降的现象，系统的资源被充分合理地利用，该算法在一个大型的作业管理系统中得到应用，效果较好。

参考文献

[1] Allcock B, Bester J, et al. Efficient Data Transport and Replica Management for High Performance Data-intensive Computing[C]// Proc. of the 8th Symposium on Mass Storage Systems and Technologies. San Diego, CA, USA: [s. n.], 2001: 13.

[2] Kosar T, Livny M. Stork: Making Data Placement a First Class Citizen in the Grid[C]//Proc. of the 24th International Conference on Distributed Computing Systems. Tokyo, Japan: [s. n.], 2004: 251-258.

[3] Maddurri R, Allcock B. Reliable File Transfer Service[EB/OL]. (2003-05-25). http://www-unix.mcs.anl.gov/madduri/main.html.

[4] Li Wen-Syan, Batra V S. QoS-based Data Access and Placement for Federated Information Systems[C]//Proc. of the 31st International Conference on Very Large Data Bases. Trondheim, Norway: [s. n.], 2005: 1358-1362.

[5] NEC Corporation. WebSAM System User's Guide[EB/OL]. (2005-09-12). http://www.nec.co.jp.