Yarnplus:基于Yarn的异构任务资源共享框架

王文峰, 胡春明，沃天宇, 杨任宇*
计算机学院 北京航空航天大学
{wangwf,hucm,woty,yangry}@act.buaa.edu.cn

Introduction

随着网络服务数量与规模的增大，越来越多的服务在云计算环境中运行。这些服务的提供需要大规模数据处理支持，同时需要对产生的数据进行分析处理从而更改进服务的质量。MapReduce已成为云环境中一种重要的大规模数据处理编程模型，包括Facebook和Yahoo！在内的许多公司与学术机构都采用了MapReduce的开源实现Apache Hadoop。这样，数据处理与分析的需求使得互联网公司、研究机构及其他企业部署了用来处理大规模MapReduce任务的Hadoop集群。同时随着网络带宽不断增长，通过网络访问非本地化的计算服务（包括应用、存储和信息服务等）的条件越来越成熟，其中由于虚拟化技术取得的突破使得IaaS (Infrastructure as a Service)发展尤其迅速，学术界与工业界都涌现出虚拟化云计算平台提供虚拟机服务。

不同需求与研究使得工业界及学术界对不同任务和服务有不同的计算集群，然而为了满足峰值资源的需求，各个集群都是采用最大资源量来部署，这样使得集群中资源利用率较低，而且不同业务集群的资源需求不同，使得资源的空闲浪费尤其严重。Facebook 2000节点的Hadoop集群的分析表明集群资源利用率不高，其中内存平均利用率不到30%。

集群资源利用率不高，主要是由于不同任务独占集群引起的空闲资源浪费，通过合理调度使得不同任务共享集群可以减少浪费提高资源利用率。然而现有的单层调度器将资源分派与作业调度由同一进程负责的特性不适用于异构任务的调度。双层调度器由元调度器负责将资源分派给任务相关的子调度器，而子调度器负责作业的调度与监控，这样不仅支持了多任务的调度，而且增强了系统扩展性。然而双层资源调度平台Mesos在资源分派中采用的Resource Offer机制容易对资源需求高的大任务产生饥饿，同时资源撤销机制又不能保证长任务的执行性能，这样使得Mesos无法适用于单任务资源需求大、任务生命周期长的任务。本文从云环境中最典型的两类任务：Hadoop MapReduce job与虚拟机VM任务出发，设计针对这两种任务可行的资源共享解决方案，并对不同方案从可行性到性能进行实验验证。根据结果，提出了一种基于Yarn（the next generation hadoop mapreduce framework）的VM与Hadoop MapReduce Job间高效可靠地资源共享框架YarnPlus，该框架能够根据全局资源使用情况和任务资源请求在异构任务间动态分派资源，避免了不同任务对相同资源竞争而产生的性能下降。从而在保证任务性能的前提下，实现了异构任务间的资源共享，减少了集群所需硬件资源，节省了成本。

Methodology

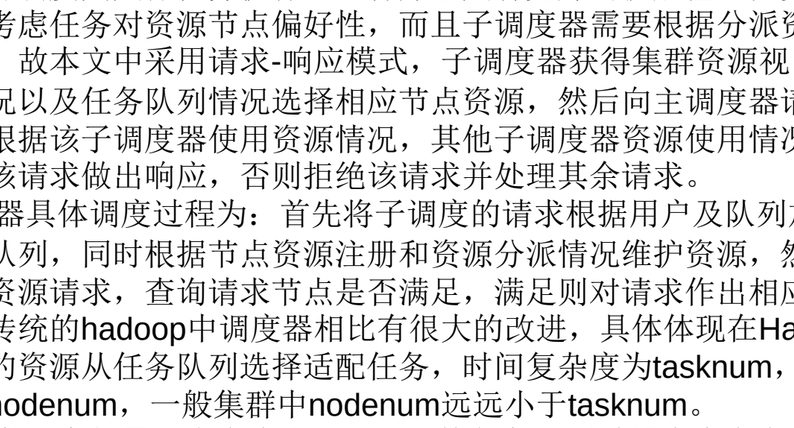
为了完成异构任务资源共享，首先进行了任务资源消耗特性的分析。Hadoop MapReduce Job与虚拟机VM是两类不同的任务。MapReduce Job是一类关心任务吞吐量的离线批处理任务，而VM是一类更在乎性能并且对延迟敏感的服务类任务。而对于资源消耗上，MapReduce是一种典型的离线批处理任务，消耗大量的CPU资源进行计算，同时消耗I/O资源读取所需处理的数据，在数据处理过程中消耗一等的内存资源。Facebook Hadoop集群内存资源利用率明显低于CPU资源利用率。而VM任务是一种在线服务类任务，由于每个VM需加载其运行所需整个操作系统，因此对内存资源消耗较高，而且内存资源的变化直接影响VM服务性能。从任务偏好性及敏感性出发，对两类任务统一以其资源消耗对其抽象，从而利用资源消耗维度动态调度两类异构任务。

资源的分派方式主要有请求分派模式和中央统一分派模式两种。中央统一分派由中央调度器根据获取的资源列表对资源进行统一分派，由各子调度器根据分派所得资源调度其具体任务执行。这种方法具有更好的扩展性，及资源分派更快，然而未考虑任务对资源节点偏好性，而且子调度器需要根据分派资源进行二次调度决策。故本文中采用请求-响应模式，子调度器获得集群资源视图副本，根据资源情况以及任务队列情况选择相应节点资源，然后向主调度器请求该资源，主调度器根据该子调度器使用资源情况，其他子调度器资源使用情况以及资源请求情况对该请求做出响应，否则拒绝该请求并处理其余请求。

主调度器具体调度过程为：首先将子调度的请求根据用户及队列加入相应的资源请求队列，同时根据节点资源注册和资源分派情况维护资源，然后不断中队列中选取资源请求，查询请求节点是否满足，满足则对请求作出相应的响应。这种方法与传统的hadoop中调度器相比有很大的改进，具体体现在Hadoop中根据心跳汇报的资源从任务队列选择适配任务，时间复杂度为tasknum，而现在时间复杂度为nodenum，一般集群中nodenum远远小于tasknum。

异构任务共享相同节点中资源，由于异构任务运行过程中会存在在一定的资源干扰和性能竞争，因此要采取相应的隔离方法来减轻这种影响，同时隔离本身不应该增大任务运行的负载。主要采用Cgroups轻量级资源隔离机制对CPU资源的使用进行CPU资源下限控制，主要是基于CPU只影响任务执行的进度，不会造成任务的直接失败。而对内存利用OS线程监控技术，动态监控内存使用变化，当发现使用异常时杀死相应的异常任务。

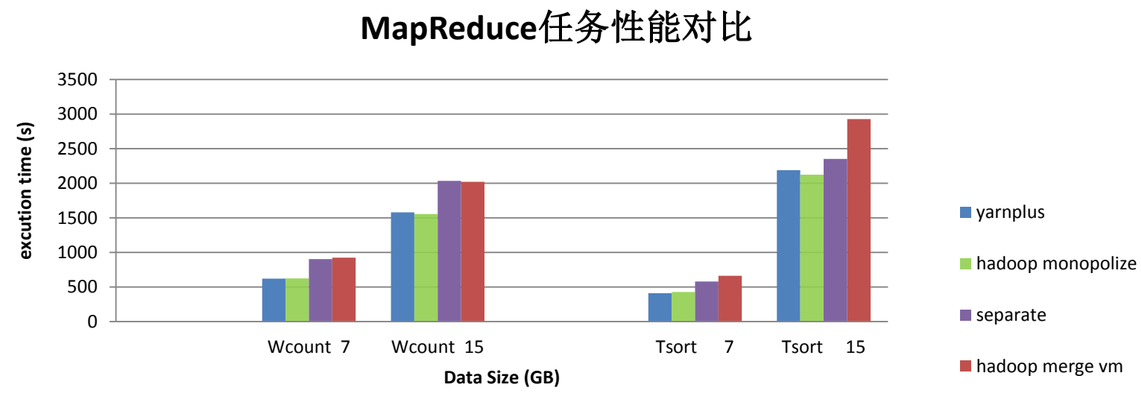
总之任务的处理流程为：用户提交任务给主调度器，主调度器将任务分发给相应的应用调度器，应用调度器根据分解任务并向主调度器申请资源，最后在相应的容器中运行任务，并定期汇报任务运行情况，具体流程如下图。

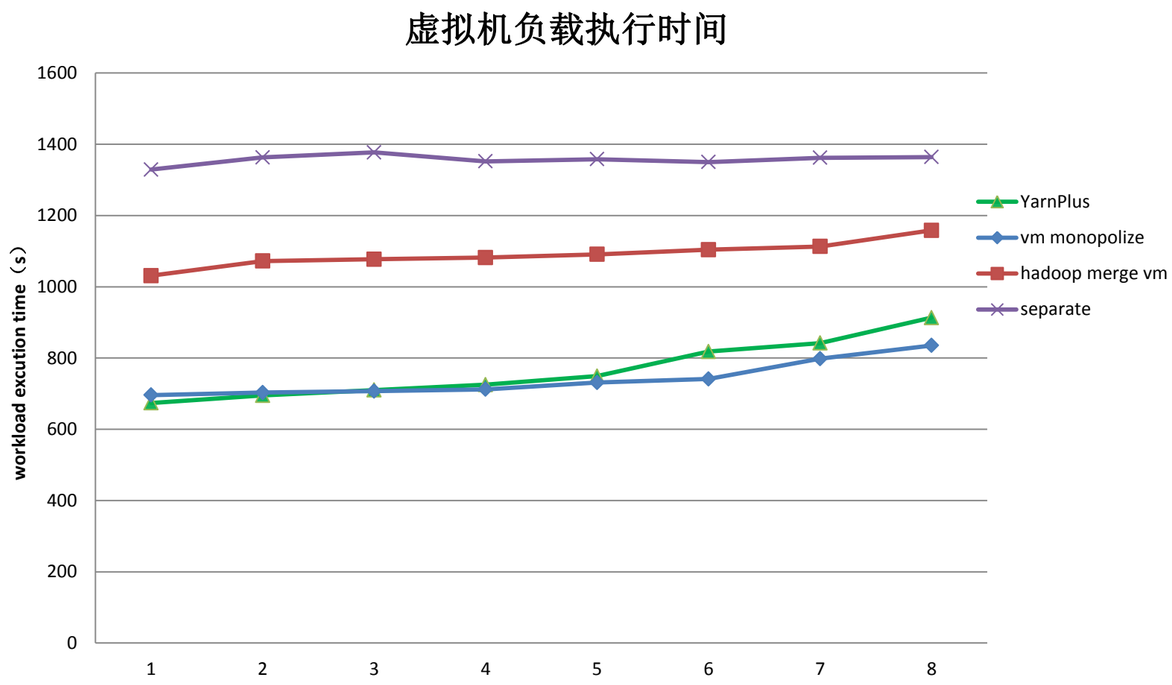


Results

异构任务共享集群可以使得集群中资源细粒度划分与共享，然而不同任务的运行特性以及出现的资源竞争可能导致任务性能下降，为此我们选择采用Cgroups方法来对任务的资源进行细粒度的隔离，同时通过统一调度来让任务的分派更加合理。在完成相应的功能模块后，通过运行典型MapReduce负载与VM负载来进行性能间的比较。性能的分析主要是针对与MapReduce任务与VM负载运行时间通过对比来衡量。具体对比环境与结果详见下表与下图。

运行环境	解释
Hadoop merger VM	集群 A 启动虚拟机并在物理机运行 Hadoop 执行 MapReduce job
Hadoop on VM	集群 A 启动低配虚拟机和高配虚拟机，高配虚拟机运行 Hadoop 并执行 MapReduce job
YarnPlus	集群 B 利用 YarnPlus 启动虚拟机，同时利用 YarnPlus 运行 MapReduce job
VM Monopolizw	集群由 VM 任务独占，只处理 VM 一种任务
MapReduce Monopolize	集群由 MapReduce 独占，整个集群只处理 MapReduce 任务





通过结果我们可以发现利用YarnPlus来实现资源共享后，对于MapReduce或者VM两类任务任务性能与将集群分割以及两类任务直接混合共享集群资源这两种情况相比，任务性能有很大的提升，这主要是因为任务可以使用整个集群资源并经过合理的调度分派，但性能与单一任务独占真个集群相比，性能略有下降，主要是由于整个集群负载增大（同时运行两类任务）的原因。

Conclusions

利用Yarn对其扩展来支持VM与MapReduce job间的资源共享，可以让Yarn的调度器同时获得集群中两类任务的资源需求和集群真实负载状况，这样避免了不同任务队同一资源竞争而产生性能下降，不仅能保证VM服务的性能，而且相比简单混合系统缩短了MapReduce Job的执行时间，高效地实现了异构任务间的细粒度资源共享，从而避免了多个集群的部署，将集群所耗机器数目减少，从而提升了资源利用率，节省了成本。

目前，扩展后的Yarn系统YarnPlus是可用的，但对多任务之间的资源分派仍然采用Hadoop中默认的调度算法，需要更加详细的分析这两类任务运行的Trace log来设计更加适用于这两种任务混合调度的算法。另外，可向用户提供一个友好的可视化界面，使用户在网页即可方便的提交任务并监控任务执行进度。以上两点是后续的主要研究内容。

References

[1] J. Polo, D. Carrera, Y. Becerra, V. Beltran, and J. T. andEduard Ayguad. Performance Management of Mccelerated Mapreduce Morkloads in heterogeneous clusters. ICPP 2010.
[2] Matei Zaharia, Andy Konwinski, Anthony D. Joseph, and Randy Katz Ion Stoica, “Improving MapReduce Performance in Heterogeneous Environments”OSDI 2008
[3] B. Hindman, A. Konwinski, M. Zaharia, A. Ghodsi, A.D. Joseph, R. Katz, S. Shenker, I. Stoica. Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center. NSDI 2011.
[4] A.Ghodsi and et al. Dominant Resource Fairness:Fair Allocation of Multiple Resource Types. NSDI 2011