成功案例

高性能大数据平台助力京东提高业务和服务效率

英特尔®至强®处理器E5-2600
电子商务
大数据分析

伴随互联网尤其是移动互联网的高速发展，电子商务已经成为人们生活中不可或缺的一部分，人们的购物理念和消费模式正在发生颠覆性的转变。基于天然的数据优势，电子商务平台利用大数据计算技术不断实施数据的累积、分析和处理，消费者的购买意愿被预测，相关的商品推荐或服务被及时推送，电商平台访问量和购买率不断提升。不但如此，这些数据源源不断地被提炼出新的价值，并和电子商务业务环节实现无缝对接，形成新的电子商务商业模式。数据已经成为电商最重要的战略资源和制定商业决策的基石。

庞大的数据分析和处理离不开大数据计算技术的发展和革新，作为中国B2C电子商务的巨头，京东集团（以下简称京东），始终以富有竞争力的价格、丰富的产品和卓越的服务，为中国亿万用户提供愉悦的在线购物体验。伴随电子商务的快速增长，京东不断拓展业务领域以满足持续增长的用户和越来越多元化的需求，不断急速膨胀的数据量为京东的大数据计算平台带来了新的挑战。利用更先进的技术，实施大数据计算平台的迭代更新，迁移至新兴计算框架，提高数据计算分析性能，提高业务效率，从而不断创造新的价值，推动商务模式创新，已经迫在眉睫。

面临挑战

- **提高业务效率：** 京东需要依赖大数据平台，通过数据分析结果为商务决策提供支持。业务的不断扩张、用户量和数据量的不断增长使得现有计算框架不堪重负，大数据平台的计算分析性能开始成为影响京东业务效率的重要因素之一。京东急需构建具备更高性能的数据仓库服务，保障用户体验，增强市场竞争优势。

解决方案

- **迁移至Spark*计算框架：** 和英特尔大数据技术中心合作，实施基于英特尔®架构的京东大数据平台计算框架从MapReduce迁移至Spark。

- **基于英特尔®至强®处理器的SparkSQL性能优化：** Large-Joins是电子商务业务场景中比较典型的案例，英特尔大数据技术中心和京东一起，通过内存使用、计算效率及执行计划等一系列优化手段，实现比旧的平台架构（基于MapReduce*框架）平均三倍的性能提升。

商业价值

- **提高业务效率：** 比旧的平台架构（基于MapReduce*框架）平均三倍性能的提升显著提高了业务效率，京东不但可以更快速的依据库存制定合理的促销价格和策略，而且实现了更灵敏的用户浏览行为跟踪和购买率统计。

迁移至Spark计算框架，基于英特尔®架构的京东大数据平台实现3倍效率提升，增强的扩展性助力京东持续保持B2C电子商务市场竞争优势。

“在英特尔大数据技术中心的帮助下，京东实现了Spark-SQL在电子商务中典型的场景，例如Large-Joins等，针对基于英特尔®至强®处理器E5-2600的多种优化，包括内存使用、计算效率和执行计划等等。最终，京东顺利完成了基于英特尔®架构的京东大数据平台计算框架从MapReduce到Spark的无缝迁移，平均三倍性能的提升帮助京东实现了提升业务效率，改善用户体验，提高购买率转化的预期目标。”

刘彦伟
技术经理
京东大数据事业部

- **保持市场竞争优势：**京东大数据计算平台增强的可扩展性使得京东可根据计算规模的需求，便捷部署新的服务器以实现计算性能的线性扩展。

- **稳定可靠的服务：**通过技术优化手段降低数据分析对服务器内存的需求，显著降低了因内存垃圾回收而导致的作业执行超时或失败现象，京东实现了高稳定性和高可靠性的大数据计算平台，确保服务质量。

早在2012年，京东就部署了大数据计算平台。经过多年的数据积累，京东大数据平台数据量业已超过100PB，并保持日均200TB的增长态势，爆发增长的数据使得以MapReduce作为计算引擎的京东大数据计算平台的弊端越来越明显，数据运算分析的性能已成为京东提升业务效率的瓶颈。

爆发式增长的数据量挑战

京东大数据事业部技术经理刘彦伟说：“用户在电子商务平台从浏览到购买，再到货物配送的整个过程，会产生许多宝贵的数据，对这些数据实施快速的运算分析是我们制定商业决策的重要支撑之一。”

然而，分散的、来自不同数据源的数据需要经过多次数据的抽取、转换和装载（ETL）过程才能到达数据仓库，成为联机分析处理/商业智能(OLAP/BI)的数据。在京东大数据平台现有基于MapReduce的计算框架下，一次ETL任务往往需要多次HDFS读写操作才能完成，额外的数据读写负载，导致数据无法被及时利用，产生报表生成慢等各种效率低下的问题。

“此外，现有计算框架亦缺乏交互式的OLAP，无法很好地支持即席查询(ad-hoc)等，这些问题对京东的业务影响越来越明显。”刘彦伟补充到：“京东急需寻找一种解决方案，能够更快速的将数据变为驱动业务效率提升的工具，保持京东在B2C电子商务领域的市场竞争优势。”

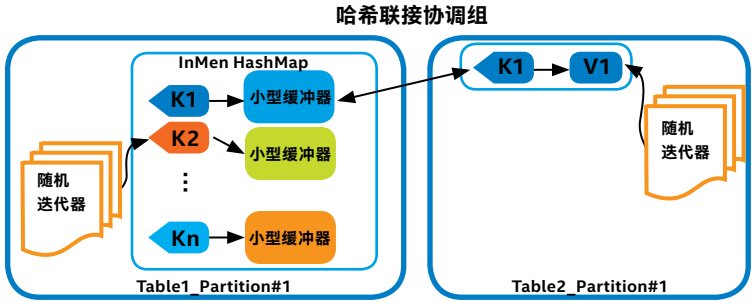
持续增长的数据分析性能需求

新技术的不断革新，大数据分析已经从Volume、Variety、Value演进到对于Velocity的追求，对京东而言，如何快速地解读大数据，不仅可以提升用户体验，更可以实实在在地快速转化为最终购买率，带来丰厚的效益。

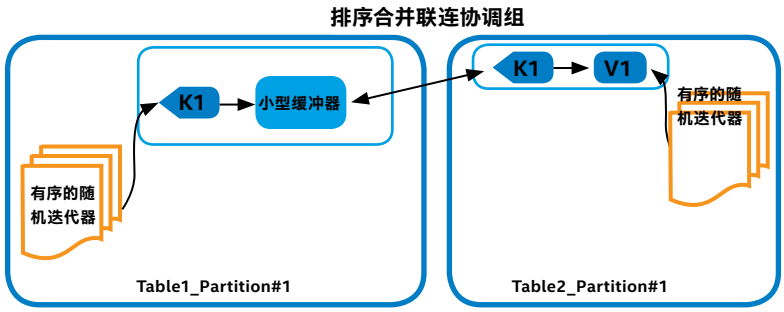
刘彦伟说：“目前，京东年活跃用户已高达1.18亿，不断扩大的业务规模和更具创新的商务模式将吸引到更多的用户，可以说，电子商务平台对数据分析的性能是永无止境的，我们需要一个扩展性更优异的解决方案满足未来不断增长的数据分析需求。”

京东决定将大数据平台的计算框架从MapReduce向Spark迁移。一方面，在Spark计算框架下，各种操作会尽可能串接在一个阶段（stage）中，直到遇到数据交换（shuffle）或者最终运算结果输出，因此Spark计算框架大大减少了对HDFS的访问。另一方面，对于一些反复使用的数据，Spark计算框架可以将计算结果缓存到内存或者磁盘上，提高后续计算的访问效率。

刘彦伟说：“显然，这将帮助我们实现更快速的将数据和业务结合的目标，实际评估过程也得到了充分的验证，现有许多查询业务不需要太多的调试，就可以看到较大



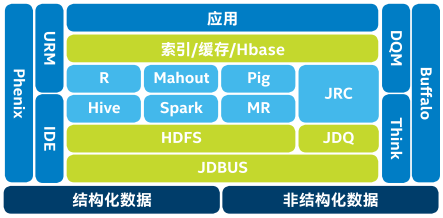
HashBasedJoin 原理示意图



SortMergeJoin 原理示意图

的性能改善。譬如：我们经常需要依据产品库存制定促销价格和策略，Spark-SQL使得我们只需要不到原来一半的时间就可得到运算结果，换而言之，我们可以实现更机动的促销方案，优化库存，实现利润最大化。”

然而，京东在测试大数据表Joins操作时，却发现了严重的GC 瓶颈或者Spark的执行器（Executor）由于内存溢出而失联，导致任务超时而失败。最终，在英特尔大数据技术中心的帮助下，京东完成了大数据平台计算框架从MapReduce到Spark的无缝迁移，顺利开始提供Spark-SQL服务。



京东大数据计算平台数据仓库架构图

利用SMJ优化内存使用

英特尔大数据技术中心和京东一起分析了京东的实际业务场景后，发现京东的large joins数据表格规模大，查询作业经SparkSQL的Catalyst优化器优化后，会引入多个ShuffledHashJoin操作——即被join的两张表都会发生数据迁移（shuffle）并重新按照join关键字分区后进行HashJoin，其中一侧的数据表分区会还会被全部存储在内存的JavaHashMap中，如果数据表格的某一块数据分区超过了分配的内存空间，就会带来巨大内存压力，造成频繁的内存垃圾回收（GC）现象，有时候甚至通过反复调试spark.sql.shuffle.partitions参数亦无济于事。

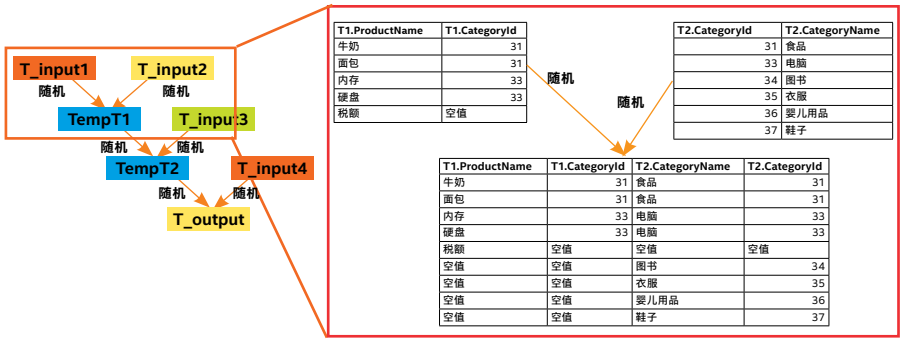
刘彦伟说：“由于Spark计算引擎基于内存计算，内存溢出而导致的作业执行超时或者失败现象将对京东大数据平台稳定性产生极大的影响。”

为此，京东采用了数据库中常用的SMJ（SortMergeJoin）。和HashJoin相比，SMJ利用外部排序算法对内存使用要求不高的特点，显著改进进程对内存的占用，减少GC带来的额外代价。测试表明，采用SMJ后，Spark计算引擎的优势得以充分显现，性能和原来MapReduce的Hive有了约3倍的提升。

“性能的提升为业务效率带来了明显的提升，数以亿计记录的大数据表连接操作，从以小时计算的运算周期缩减到以分钟计算。也正是因为如此，我们不但可以比用户更加了解用户的购买需求，提供更精准的商品推荐， 提高用户满意度和购买率，而且实现更智慧的商品采购和销售策略，避免脱销或过高的库存成本。”刘彦伟说。

经验

- 基于英特尔®架构的大数据平台可以满足电子商务对数据分析的性能需求，英特尔大数据技术中心提供丰富的技术支持和优化方案，进一步实现性能的优化。
- 将大数据平台的计算框架从MapReduce迁移至Spark，获得三倍的数据分析性能提升。
- 英特尔®至强®处理器 E5-2600支持DDR4内存，更多的内核、支持更大的内存容量和更高的带宽为采用Spark计算框架的大数据计算平台带来更多优势，是电子商务大数据计算平台值得信赖的选择。
- 利用基于英特尔®架构、采用Spark计算框架的大数据计算平台，实现增强的横向扩展性，持续满足电子商务大数据平台对运算性能需求。
- 利用英特尔®固态盘和英特尔®万兆位以太网服务器适配器实现Spark在shuffle环节7~8倍的速度提升，提升电子商务业务效率。



多路连接(Join)产生严重空值数据倾斜示意图

引入Reduce端的Sort Based Shuffle实现计算效率的进一步提升

此外，京东技术人员注意到：SMJ的使用导致增加了一次shuffle过程后的数据排序操作，因而其计算效率不如HashJoin。详细研究后发现，原来基于横向扩展性考虑，Spark内核中的shuffle在1.1版本中有过一次升级，在Map端引入了Sort Based Shuffle(即对键值的Hash值进行归并外排)，利用归并排序减少并行打开的文件个数，从而减轻磁盘IO负载，然而其Reduce端依然采用的是Hash Based Shuffle 方式。于是，当用户需要对shuffle输出数据排序时，就会产生额外的排序操作与计算。

刘彦伟说："我们必须进一步提高排序的计算性能以满足京东大数据平台横向扩展性的需求，在英特尔大数据技术中心的帮助下，京东决定引入Reduce端的Sort Base Shuffle以避免额外的计算操作。"首先，对原先默认的Map 端Sort Based Shuffle数据进行分区内部排序。当Reduce端做汇聚时，利用已知排序结果进行快速归并，高效获得有序的Shuffle输出。

结果表明，在引入Sort Based Shuffle Read之后，京东大数据计算平台SparkSQL SMJ的性能得到了20%的进一步提升，十分接近HashJoin效果。刘彦伟说："至此，京东大数据计算平台large joins的横向扩展性得以解决，同时执行时间更短。面向未来不断增加的数据量和运算量，我们可以便捷的为计算集群增加新的服务器满足业务需求。"

优化执行策略缩短作业执行时间

此外，在京东的实际业务中，例如：对用户浏览行为的跟踪及购买转化率统计，需要引入一些多表同键值的多路 Full Outer Join，部分数据表格规模较大，且分区不均匀的特点使得在Spark计算框架下，Join的性能并没有显著提升，出现因大量空值数据倾斜而拖慢整个查询作业的执行时间，导致查询失败，甚至异常和作业失败等现象。

英特尔大数据技术中心通过对类似案例执行计划的分析，提出进一步优化的方案。即：忽略数据表空键值的随机分布，避免引入额外的数据shuffle操作，并在同一个分区内直接进行co-location的join操作。该方案不但节省了中间数据shuffle所带来的高额I/O和网络通信代价，而且避免了由于中间数据处理所产生的大量单键值为空的严重数据倾斜现象。最终，执行时间和Hive on MapReduce相比，缩短到原来的1/2.5。

得益于英特尔大数据技术中心的支持，采用Spark计算框架的京东基于英特尔至强处理器E5-2500服务器的大数据平台的性能和横向扩展性得以施展，显著的性能改善将帮助京东实现业务和服务效率的不断提升。面向未来，京东将继续和英特尔公司合作，持续利用技术优势保障用户体验。

其它值得分享的配置参数和优化手段

- spark.yarn.executor.memoryOverhead该参数默认值为384与0.07 * spark.executor.memory之间的最大值，该值越大，执行器(executor)被Yarn调度器因过多的off-heap内存使用而杀掉进程的概率越小。
- spark.sql.shuffle.partitions/spark.shuffle.sort.bypassMergeThreshold此两参数默认值为200，当服务器产生较高的IO访问压力时，可以考虑升高spark.sql.shuffle.partitions或者降低spark.shuffle.sort.bypassMergeThreshold。
- spark.dynamicAllocation.enabled/spark.shuffle.service.enabledynamicAllocation对于SparkSQL的资源利用率得到了很好的调配，但其必须配合shuffle.service一起使用，以避免多余executor被杀掉后，原执行器(Executor)所产生的shuffle数据不可访问。
- spark.speculation在查询中，如果经常出现因某些任务而拖慢整个作业的执行时间时，打开speculation之后，任务的执行会变得更加平均，显著缩短作业时间。
- 英特尔®固态盘和英特尔®万兆位以太网适配器对于涉及多个shuffle阶段的large joins性能具有非常明显的帮助，单shuffle环节可以得到7-8倍的速度提升。

找到适合于您公司的解决方案，请与您的英特尔代表联系或访问**英特尔商用频道**。

扫描关注英特尔商用频道，查看更多英特尔前沿技术及解决方案。

英特尔技术特性和优势取决于系统配置，并可能需要支持的硬件、软件或服务才能激活。没有计算机系统是绝对安全的。更多信息，请见Intel.com，或从原始设备制造商或零售商处获得更多信息。

性能测试中使用的软件和工作负荷可能仅在英特尔微处理器上进行了性能优化。诸如SYSmark和MobileMark等测试均系基于特定计算机系统、硬件、软件、操作系统及功能。上述任何要素的变动都有可能导致测试结果的变化。请参考其他信息及性能测试(包括结合其他产品使用时的运行性能)以对目标产品进行全面评估。

更多信息，详见http://www.intel.com/performance。优化声明：英特尔编译器针对英特尔微处理器的优化程度可能与针对非英特尔微处理器的优化程度不同。这些优化包括SSE2、SSE3 和SSSE3 指令集和其他优化。对于非英特尔微处理器上的任何优化是否存在、其功能或效力，英特尔不做任何保证。

本产品中取决于微处理器的优化是针对英特尔微处理器。不具体针对英特尔微架构的特定优化为英特尔微处理器保留。请参考适用的产品用户与参考指南，获取有关本声明中具体指令集的更多信息。所有性能测试的结果均来自于京东的测试报告。想了解更多关于性能和标准的信息，请联系京东(www.jd.com)。

英特尔不控制或审计本文提及的第三方基准测试数据或网址。请访问提及的网站，以确认提及的数据是否准确。

描述的成本降低情景均旨在在特定情况和配置中举例说明特定英特尔产品如何影响未来成本并提供成本节约。情况均不同。英特尔不保证任何成本或成本降低。

英特尔公司2016年版权所有。所有权保留。是英特尔在美国和/或其他国家 的商标。英特尔、英特尔标识和英特尔至强是英特尔在美国和/或其他国家的商标。

*其他名称和品牌可能属于其它公司的财产。

0616/SHA/PMG/XX/PDF 334407-001CN