数字有机体系统的大数据处理

1. 概述

按照教科书式的解释，大数据（Big Data）具有四个特性，即体量大、价值密度低、数据类型多样和处理速度要求高。这四个特性的综合影响对数据的输入、存储、传输和处理带来以下难题：

1）数据不再是集中产生和输入，而是由分散的主机并行的产生和输入系统。这就要求系统具有分布式并行输入输出能力。

2）不仅要海量存储，还需要充足的并行读取带宽。现在的 SAN 系统已经可以提供上 P 级的存储容量，但是要提供满足 P 级数据并行处理的读取带宽则是困难的事情。

3）数据和处理融合的难题。如果数据处理和数据存储分离，则处理时必然需要大量传输数据，显然现有的网络无法提供大数据处理需要的充足带宽。因此，必须让数据存储和处理紧密的结合在一起。

4）数据的多样性要求系统具有灵活的数据处理方式。单纯的关系数据库、NoSQL 数据库、并行计算环境或者文件大数据分析环境无法满足数据多样性的需求。

5）秒级的数据处理速度。过长的数据处理延迟使得大数据分析的价值降低。

显然这些难题已经超出了传统系统的处理能力，增对这些问题，数字有机体系统按照数据组织的两种方式，即文件和数据库，分别给出解决方案。基于文件的大数据解决方案针对非结构化数据，针对数据库的解决方案针对结构化数据。而且两者是可以集成使用的，可以共同应对各种应用的大数据处理问题。

2. 基于文件的非结构化数据处理

2.1. 功能和特性

该方案由数字有机体文件系统和数字有机体远程过程调度用（DOSRPC）实现。数字有机体文件系统具有以下功能和特性：

1）系统由大量分散分布的，用高速网络互连的节点构成。每个节点既是计算设备也是存储设备。

2）整合分布在系统中的各节点上的存储设备，形成统一的存储池，满足大数据处理海量存储的需求。

3）为各计算设备提供传统文件系统的共享文件服务，简化共享数据访问的复杂度。

4）提供高扩展性，系统可根据规模扩展，存储容量不存在瓶颈问题。

5）提供分布式并行输入输出，系统的每个节点都可作为文件系统入口读写文件，存取数据，不存在输入输出瓶颈。

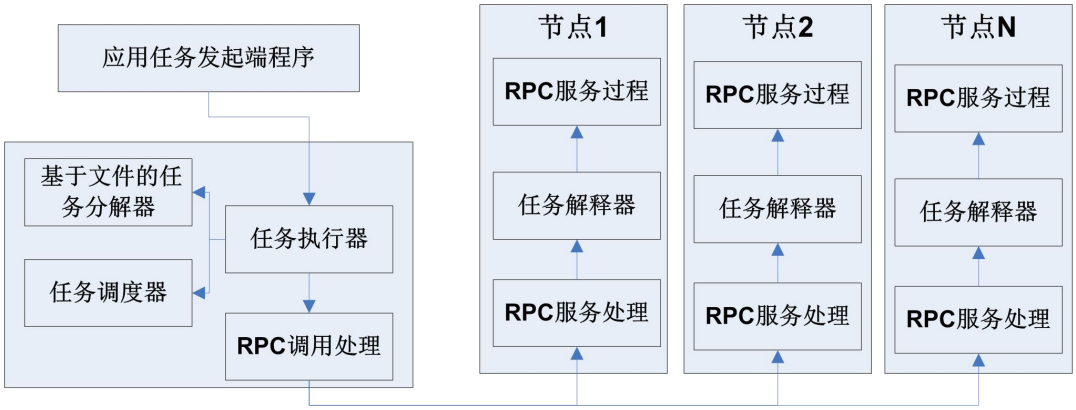
6）具有文件分块存储功能，支持超大文件的存储，单个文件大小不受单一存储设备容量的限制。

7）具有多副本存储功能，副本数量和位置可控；

8）自动处理节点故障，包括自动增减副本，调整副本位置，以及屏蔽故障节点等。

9）无需共享存储设备。

DOSRPC 基于数字有机体系统研发，利用了数字有机体文件系统的特性。DOSRPC 是一套基于文件的大数据处理的开发环境。其原理可以用下图表示：



应用程序按照 RPC 编程模型分为两个部分，即任务发起端程序和任务执行端程序。任务执行端程序注册 RPC 服务过程，供任务发起端程序调用。在 RPC 的基础上，增加任务执行器，由其调用其他模块完成任务分解和任务调度，并分布并行地在许多节点上调用服务过程。现在实现按照文件以及文件分块情况分解任务的功能，即如果文件不分块则将整个文件作为一个任务，如果文件分块存储则每个文件分块就是一个任务。任务调度主要根据节点的 CPU 性能和磁盘性能均衡地分配任务。当某个节点死亡，其上需要执行的子任务如果在别的节点上还有存储，则在别的节点上重新启动该子任务的执行。每个任务执行的结果以函数调用返回值的方式传递给回调函数。从而应用任务发起端程序可以处理个人子任务执行的结果。如果子任务执行的结果很大，还可以写入数字有机体文件系统，只返回文件名，然后通过数字有机体文件系统的共享功能读取子任务处理的结果。

DOSRPC 自身具有以下功能和特性：

1）自动按照文件以及文件分块的情况分解任务；

2）自动按照文件或者文件分块的位置，以及系统状况分配任务；

3）利用远程过程调用机制，将计算任务转移到存储节点上执行，避免大数据传输问题，提升数据处理效率；

4）分布并行地在系统节点上执行这些任务，缩短任务执行时间；

5）处理任务执行节点的故障，自动选择其他可执行节点重新执行失败的子任务；

6）支持多 RPC 联合执行（即一个 RPC 处理的结果以文件的形式传递给下一个 RPC，多个 RPC 间是流水线关系），以并行流水线的形式提升处理效率；

7）提供检查点机制，通过保存的检查点文件可以恢复 DOSRPC 调用的执行，应对调用发起节点故障的情况；

8）以成熟的远程过程调用（RPC）编程模型为接口，易于掌握；

9）具有程序生成器和任务执行监视器等，易于使用；

10）支持多个 DOSRPC 调用并行执行。

2.2. 实例测试结果

为了检验该方案的性能，本次进行了两个不同案例的测试。第一个测试案例是在大量文

件中寻找某个单词，并统计该单词出现的次数。在舆情分析系统中，该功能常用在敏感词分析上。第二个案例是大文件中数据抽取和排序。其功能是从许多大文件中抽取出数字，然后对抽取出的数据进行排序。

测试的网络为 100Mbps 交换网络。测试的主机中，大部分是如下配置的 PC 计算机，还有两台如下配置的普通服务器。

机型	CPU	内存	硬盘	数量（台）
PC 计算机	AMD Phenom II X6 1055T	3.8GB DDR2 内存	单个 7200 转 SATA-2 硬盘	4
普通服务器	Intel(R) Xeon(R) CPU E56202.40GHz	8GB 服务器内存	4 个 1500 转 SAS 硬盘，做 RAID5 阵列	2

测试在自动生成的 100GB 数据上进行。下表是单词查找统计案例的对比测试结果。

测试项目	测试内容	节点数量（台）	花费时间（秒）
PC 计算机单机测试	在单台 PC 计算机上完成单词查找统计	1	3029
普通服务器单机测试	在单台普通服务器上完成单词查找统计	1	1771
4+2 测试	在四台 PC 计算机和 2 台普通服务器上测试	4+2	313

从测试结果可以得出：即使是相对于普通服务器来说，4+2 测试的查询统计速度都达到 5 倍，如果是相对 PC 计算机来说则达到 9 倍。这表明利用 DOSRPC 可以有效的提升数据处理速度。

下表是数据抽取排序的测试结果。

测试项目	测试内容	节点数量（台）	花费时间（秒）
PC 计算机单机测试	在单台 PC 计算机上完成单词查找统计	1	2756
普通服务器单机测试	在单台普通服务器上完成单词查找统计	1	1351
10+2 测试	在 10 台 PC 计算机和 2 台普通服务器上测试	10+2	235

从测试结果可以得出：10+2 测试的花费时间仅是单台普通服务器花费时间的近六分之一。在 10 台 PC 计算机中，有 6 台是性能更差的 PC 和虚拟机，因此贡献的处理能力不多。而相对于单台 PC 计算机来说，则花费时间近 1/12。

除了性能测试外，也测试了服务节点故障的情况，系统能够自动处理节点故障，使 DOSRPC 调用正常完成。如果是调用发起节点故障，则只能利用检查点恢复执行。也测试了并行进行单词查询统计和数据抽取排序，除因争用资源而增大处理时间外，程序都能正常结束。

结论：基于文件的大数据处理能够有效缩短处理时间，能处理服务节点故障，支持多任务并行执行。

3. 基于数据库的结构化数据处理

3.1. 功能和特性

数字有机体工作库提供结构化数据的大数据处理支持。它在数据库管理系统内部内置分布式并行查询引擎，不仅支持单表查询的分布式并行执行，还支持多表连接查询的分布式并行执行。数字有机体工作库在大数据方面的功能和特性分别如下：

1）支持数据表水平分片；

2）数据表分片可分布在大量节点上，支持逻辑表的海量数据存储；

3）无需共享的存储设备；

4）数据表分片本身支持多副本复制，副本数量和位置可控；

5）系统自动根据查询语句和表分片情况分解执行任务，将任务并行地在分片存储节点上执行；

6）不仅支持单表查询的分布式并行执行，也支持多表连接查询的分布式并行执行；

7）系统自动处理和屏蔽故障节点，使查询在有节点故障时仍然可以继续执行；

8）系统的每个节点都是访问入口，都可进行所有数据操作，支持分布式并行输入输出，没有入口瓶颈；

9）每个数据表分片都可独立直接的访问，从而提升数据写入性能；

10）支持多个用户并行执行查询；

11）具有良好的扩展性，可通过重新分片扩展分片数量，通过增加服务器可以分散副本的存储，以便支持更多用户的并行操作。

3.2. 实例测试结果

在单表查询性能测试方面，进行了单表 2100 万条记录的对比查询测试。在多表联合查询方面，利用标准的数据库性能测试工具 TPCH 进行了 10GB 数据的测试。下面分别说明两个测试的结果。

3.2.1. 单表查询性能测试

被测试的数据表为 tx_verify_gaj_info，共有 2100 万条记录，单纯数据量为 5GB，在数据库中的存储量为 9GB。测试环境为 100Mbps 网络，测试主机为 6 台 PC 计算机和 2 台普通服务器，其配置如下：

机型	CPU	内存	硬盘	数量（台）
PC 计算机	AMD Phenom II X6 1055T	3.8GB DDR2 内存	单个 7200 转 SATA-2 硬盘	6
普通服务器	Intel(R) Xeon(R) CPU E56202.40GHz	8GB 服务器内存	4 个 1500 转 SAS 硬盘，做 RAID5 阵列	2

测试使用的语句分别如下：

项目	说明	语句
1	在非索引上的条件的记录数统计	select count(*) from tx_verify_gaj_info where server_no=”server_no”;
2	在索引上的条件的记录数统计	select count(*) from tx_verify_gaj_info where register_no=”no”;
3	在非索引上的条件的 sum 统计	select sum(current_node_code) from tx_verify_gaj_info where name=”name”;
4	在索引上的条件的 sum 统计	select sum(current_node_code) from tx_verify_gaj_info where apply_date=”date”;
5	在非索引上的记录查询	select * from tx_verify_gaj_info where name=”name”;
6	在索引上的记录查询	select * from tx_verify_gaj_info where register_no=”no”;
7	在非索引上的条件的 max 统计	select max(current_node_code) from tx_verify_gaj_info where name=”name”;
8	在索引上的条件的 max 统计	select max(current_node_code) from tx_verify_gaj_info where register_no=”no”;
9	统计表的记录数	select count(*) from tx_verify_gaj_info;
10	索引上字符串的 like 查询	select * from tx_verify_gaj_info where apply_date like “2010-09%” limit 10;

测试结果显示如下：（数据为查询响应时间，单位都为秒）

项目	低性能服务器无分片	4 台低性能服务器 4 分片	较高性能服务器无分片	2 台较高性能服务器 4 分片	2 台较高性能服务器 8 分片	2 台较高性能服务器 16 分片	2 台较高性能服务器 32 分片	8 台混合服务器 16 分片	8 台混合服务器 32 分片
1	108.00	6.30	34.41	5.24	3.21	2.98	4.07	1.72	1.73
2	0.03	0.06	0.01	0.01	0.02	0.01	0.09	0.01	0.01
3	110.00	6.52	38.26	5.48	3.26	2.92	5.14	1.80	1.77
4	0.03.00	0.01	0.01	0.01	0.01	0.01	0.03	0.01	0.01
5	127.00	9.14	65.09	9.90	5.21	3.30	6.44	2.73	3.02
6	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
7	110.00	6.58	43.14	5.60	3.26	2.93	3.66	1.70	1.87
8	0.03	0.01	0.01	0.01	0.01	0.01	0.02	0.04	0.18
9	96.20	10.30	29.57	9.06	9.23	9.44	9.47	8.46	10.78
10	0.01~11	0.01~26	0.01~4.51	9.95	5.13	3.27	6.42	2.67	3.12

从测试结果可知：

1）有索引时，因单个操作本身就很快，因此性能没有提升，反而因增加额外开销而减少。但是，如果数据量继续增加，则即使有索引，查询耗时也会增长，从而额外开销就不明显，而处理加速则更加明显。

2）从测试结果看，如果传递的结果集很小，则额外开销仅在 0.01 秒左右。随着结果集增大，传输结果集的额外开销也就增加。不过，通常查询的结果集都不是很大。

3）如果查询不能使用索引，则必须进行全表扫描处理。这时，扫描的速度受检查字段总长度、记录数和条件复杂度等影响。很显然，全表扫描是很耗时的，因此并行处理必然提升查询性能。这就是这些语句明显提升的原因。其中第 5 和第 10 条语句的查询条件耗时更多，因此其响应延迟也稍大些。

4）第 9 条语句在分片很仍然没有并行执行，采用的是串行的处理每个分片然后累计结果的方式，因此其处理时间都比其他语句长。不过，因为分片后每个分片表的数据量更少，因此处理的速度更快，即使串行处理也比不分片时更快。

结论：在同等测试条件下，8 台同上配置的普通服务器足以将查询响应延迟缩短到 1 秒以下。

另外，在 8 台混合服务器的环境下，分别测试了对单个分片的和整个表的不同并发任务数的查询性能。查询语句为：select count(*) from tx_verify_gaj_info where name like “ds%”; 数据库服务器禁止了查询结果缓存，以便测试到真实查询的速度。

下表是测试结果。时间单位为秒。

并行任务数	1	2	3	4	5	6	7	8
单分片查询速度	0.76	0.85	0.88	0.93	1.3	1.5	1.67	1.85
整个库查询速度	1.7	2.32	3.3	4.3	5.7	6.9	7.7	8.7

从测试结果可知：并行执行多任务将增加查询的响应时间，因此应相应的增加服务器数量或者提升服务器配置。

3.2.2. 多表连接查询的性能测试

本测试采用的是数据库性能测试工具 TPCH，版本为 2.14.0。测试的数据量为 5GB，在数据库中的存储量为 14GB。其中大表 orders 有 750 万条记录，lineitem 有 3000 万条记录。对比测试不分片和 16 分片的情况。不分片时在普通服务器上执行。分片时，采用四台 PC 计算机和两台普通服务器构建系统。

语句序号	语句简述	单机执行时间	数字有机体工作库执行时间
1	单大数据表统计	155.85	235.99
2	多普通表连接	49.78	54.87
3	普通和大表连接统计	65.73	7.67
4	两个大表嵌套子查询	53.65	179.49
5	含两个大表的多表连接统计	62.14	39.49
6	大表统计	87.67	23.91
7	对含大数据表的多表连接的结果做统计	61.85	56.68
8	对含大数据表的多表连接的结果做条件统计	85.87	23.04

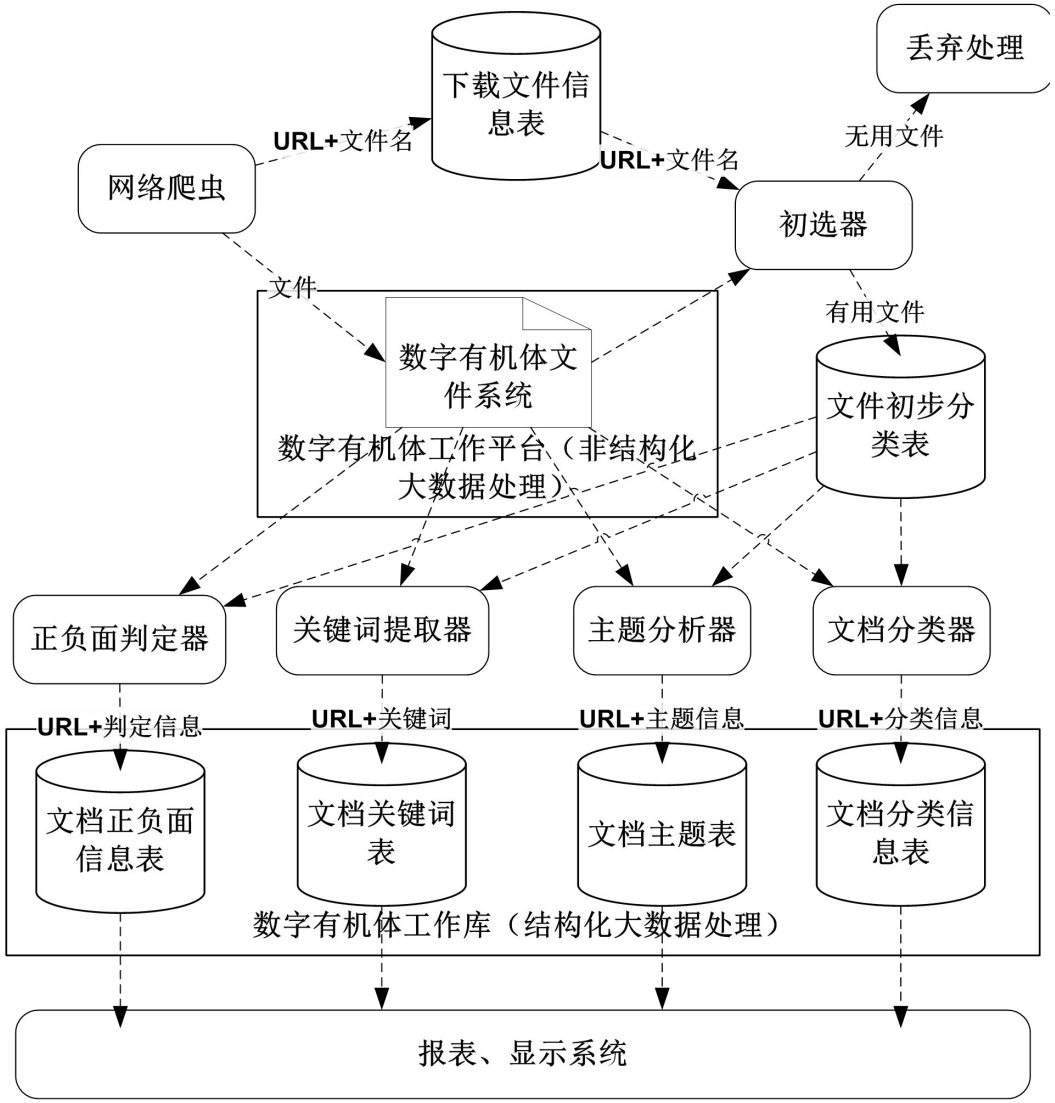
9	对含大数据表的多表连接的结果做分组统计	x	x
10	含大数据表的多表连接的复杂分组统计	96.97	55.48
11	普通表嵌套查询	53.31	43.28
12	两个诱导分区大表连接的条件统计	112.09	9.20
13	大表嵌套子查询的统计	44.80	14.61
14	普通和大表连接统计	71.31	12.06
15	大表快照和普通表的连接选择	136.90	6.61
16	普通表嵌套查询	71.27	10.37
17	大表和普通表连接加嵌套的大表子查询	59.23	54.95
18	大表和普通表连接加嵌套的大表子查询，分组统计	82.36	36.40
19	大表连接，带复杂选择条件	117.29	18.97
20	含大表的复杂嵌套子查询	96.87	x
21	含大表的复杂嵌套子查询	167.87	x
22	普通表复杂嵌套子查询	15.91	14.57
总时间开销(忽略因执行时间太长放弃的项)		1483.98	897.64

x 表示时间太长放弃测试。其中第 9 条因优化方式问题执行时间过长。第 20 和第 21 条则因分片方式不适宜执行关联的嵌套子查询而执行时间过长。

从表中结果可以看出：除了关联嵌套子查询因为在分片表上执行不适宜外，其他涉及大表的查询语句的执行时间都缩短了。典型的是诱导分区表的链接和普通表和大数据表的链接，例如 12、13、14、15 等。

4. 综合应用

在许多实际应用中，同时存在着非结构化数据和结构化数据的处理需求。单纯针对某类数据的解决方案并不能很好的满足需求。例如，Hadoop 系统对非结构化数据能够有效的处理，但是对海量结构化数据则难以处理。当其应用到舆情分析系统时，难以利用预先抽取结构化数据的方式加快分析速度。预先抽取的结构化数据只能保存在另外的关系数据库中。如果不预先抽取结构化数据，则每次都盲目的对原始的非结构化数据进行分析，不仅系统开销大，而且分析处理的响应时间长，无法满足应用需求。但是，当预先抽取的结构化数据的数据量很大时，普通的关系数据库又难以满足查询速度需求。因此，必须将结构化数据和非结构数据的大数据处理整合在一起。下面以舆情分析为例，说明如何应用数字有机体系统同时解决两类大数据的处理问题。



上图是数字有机体系统的大数据处理在舆情系统中应用的设计。在这个框架中，数字有机体工作平台作为文件存储、访问、共享和非结构化大数据的处理基础实施。数字有机体数据库作为各个表的存储、访问、共享和结构化大数据的处理基础实施。

网络爬虫程序可部署在大量服务器上，它们负责从网络上抓取网页、图片、视频、音频等数据；然后将数据以文件形式保存在数字有机体工作平台中；同时将数据的 URL 和文件名存储在“下载文件信息表”中。数字有机体工作平台为所有业务程序提供一致的文件系统映像，使得他们可以共享各台服务器上的文件。每台服务器都可以独立的访问这个一致的文件系统，因此各台服务器可以并行的工作。

初选器、正负面判定器、关键词提取器、主题分析器和文档分类器等都采用 DOSRPC 结构编写程序。各文档处理器的处理程序同时运行在系统的各台服务器上，用数字有机体远程过程调用处理要分析的各个文件。处理任务通过 DOSRPC 系统调度后，自动由保存文件的服务器执行，从而避免了大数据的传输；系统服务器可以并行的处理保存在文件中的非结构化数据，大大的缩短了处理延迟；系统不依赖单一的存储设备（如区域存储网络等），每台主机都可以并行的访问自有的独立的存储设备，系统的输入输出能力和处理能力都可以无限扩展，足以满足舆情分析对存储系统输入输出能力的要求。

各种预处理器抽取出的结构化数据都保存在数字有机体工作库中，即使每个数据表的数据量很大，也可以利用数字有机体工作库的分布式并行查询功能，快速的完成查询处理。

在舆情分析中，用户的许多分析需求都是类似的，可以通过预先分析原始文件的方式加快分析速度。这就可以充分利用数字有机体工作库大数据处理能力，预先进行足够充分的分析，而无需担忧数据库查询能力不足。当用户需要特殊的需求时，又可利用数字有机体工作平台的大数据处理能力，通过大量服务器的分布式并行处理，快速获得分析结果。因此，数字有机体系统可以同时满足应用对结构化大数据和非结构化大数据的处理需求。将两者完美的结合起来，才能应对各种大数据应用的多变需求。