

# DBAplus Newsletter 第一期

## 目录

DBAplus Newsletter .....	1
推出 DBAplus Newsletter 的想法 .....	3
DB-Engines 数据库排行榜解读 .....	4
SQL、NoSQL 及 NewSQL 的发展历程 .....	5
RDBMS 家族 .....	6
Oracle .....	6
12c Release 2 文档发布 .....	6
12c Release 2 特性解读 .....	7
MySQL .....	10
MySQL 5.7 新特性解读 .....	10
MySQL 分支 Percona .....	13
Percona 5.7.16-10 发布 .....	13
MySQL 分支 MariaDB .....	14
MariaDB 10.1.19 发布 .....	14
SQL Server .....	15
SQL Server 2016 SP1 补丁包发布 .....	15
SQL Server On Linux 版本发布 .....	16
PostgreSQL .....	16
PostgreSQL 9.6 发布 .....	16
PostgreSQL 9.6.1 patch 发布 .....	18
DB2 .....	18
DB2 LUW V11.1 发布 .....	18
Greenplum .....	20
GPCC3.0 发布 .....	20
GPText 2.0 发布 .....	20
NoSQL 家族 .....	21
MongoDB .....	21
MongoDB 3.4 版本发布 .....	21
Redis .....	22
Redis 3.2 版本发布 .....	24
HBase .....	22
Apache HBase v1.2.4 发布 .....	23
Geode .....	25
Apache Geode 1.0 版本正式发布 .....	25
NewSQL 家族 .....	26
TiDB .....	26
TiDB 近期将发布 RC1 版本 .....	26
RethinkDB .....	27

RethinkDB 深情告别 .....	27
大数据生态圈 .....	27
Hadoop .....	27
Hadoop 3.0.0 Alpha 版本发布 .....	27
Druid .....	30
Druid 0.9.2 版本发布 .....	30
Kudu .....	31
Apache Kudu 1.1.0 正式发布 .....	31
HAWQ .....	32
HAWQ 2.1.0.0 企业版正式发布 .....	32
Docker 发展概况 .....	33
官方版本发布情况 .....	33
国内情况 .....	33
落地情况 .....	34
国产数据库概览 .....	34
达梦 DM .....	34
达梦 V7.1.5.117 发布 .....	34
GBase .....	35
GBase 8a MPP Cluster 发布最新版本 V8511 .....	35
GBase 8t 发布最新版本 V8.7 .....	36
星瑞格数据库 .....	36
12.10 FC7 版本发布 .....	36
感谢名单 .....	38

## 推出 DBAplus Newsletter 的想法

### 做 Newsletter 的目的

DBAplus Newsletter 旨在为广大技术爱好者提供数据库行业的最新技术发展趋势，为技术发展提供一个统一的发声平台。为此，我们策划了 RDBMS、NoSQL、NewSQL、大数据、虚拟化、国产数据库等几个版块的内容。

### 做 Newsletter 的底线

我们承诺这个栏目不以商业宣传为目的，不接受任何商业广告宣传，严格审查信息源的可信度和准确性，力争为大家提供一个纯净的技术学习环境，也欢迎大家监督指正。

### 对 Newsletter 的态度

因为一个人的能力毕竟有限，所以我们集合了大量一线专家及厂商支持。我们愿意并希望能和更多技术组织紧密合作，为广大技术爱好者提供最新的技术资讯；我们提供一个开放、友好的平台，支持并鼓励广大技术爱好者积极参与到这个栏目中来。

这是我们第一期 Newsletter 版本的发布，还有很多数据库技术的遗漏，还有很多细节需要补充完善，但我们相信推出了这一版，有了一些经验，以后的工作就会顺利很多。

### Newsletter 发布周期

至于 Newsletter 发布的周期，目前计划是每两个月左右做一次跟进，如果有相关的信息会汇总发布出来。

### Newsletter 简单声明

Newsletter 中所列举的数据库排名不分先后，如果没有上榜很可能是截止发布前还没有获取到相关信息，有兴趣的技术爱好者可以帮忙提供给我们，我们负责整理发布。在此声明，避免过度解读。

希望广大技术爱好者能在其中学有所得，也希望社群能紧贴技术发展趋势，我们互相成就。

欢迎提供 Newsletter 信息，发送至邮箱：[newsletter@dbaplus.cn](mailto:newsletter@dbaplus.cn)

欢迎技术文章投稿，发送至邮箱：[editor@dbaplus.cn](mailto:editor@dbaplus.cn)

## DB-Engines 数据库排行榜解读

以下取自 **2016 年 12 月** 的数据，具体信息可以参考 <http://db-engines.com/en/ranking/>

DB-Engines 排名主要根据五个因素来进行：Google 以及 Bing 搜索引擎的关键字搜索数量、Google Trends 的搜索数量、Indeed 网站中的职位搜索量、LinkedIn 中提到关键字的个人资料数以及 Stackoverflow 上相关的问题和关注者数量，数据仅供参考。

312 systems in ranking, December 2016

Rank			DBMS	Database Model	Score		
Dec 2016	Nov 2016	Dec 2015			Dec 2016	Nov 2016	Dec 2015
1.	1.	1.	Oracle +	Relational DBMS	1404.40	-8.60	-93.15
2.	2.	2.	MySQL +	Relational DBMS	1374.41	+0.85	+75.87
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1226.66	+12.86	+103.50
4.	4.	5.	PostgreSQL	Relational DBMS	330.02	+4.20	+49.92
5.	5.	4.	MongoDB +	Document store	328.68	+3.21	+27.29
6.	6.	6.	DB2	Relational DBMS	184.34	+2.89	-11.78
7.	7.	8.	Cassandra +	Wide column store	134.28	+0.31	+3.44
8.	8.	7.	Microsoft Access	Relational DBMS	124.70	-1.27	-15.51
9.	9.	10.	Redis	Key-value store	119.89	+4.35	+19.36
10.	10.	9.	SQLite	Relational DBMS	110.83	-1.17	+9.98
11.	11.	13.	Elasticsearch +	Search engine	103.27	+0.70	+26.71
12.	12.	14.	Teradata	Relational DBMS	73.37	-1.79	-2.34
13.	13.	11.	SAP Adaptive Server	Relational DBMS	70.42	+0.26	-11.05
14.	14.	12.	Solr	Search engine	69.00	+0.64	-10.15
15.	15.	16.	HBase	Wide column store	58.63	-0.11	+4.38
16.	16.	18.	Splunk	Search engine	54.92	+0.19	+11.06
17.	17.	17.	FileMaker	Relational DBMS	54.12	+0.20	+4.00
18.	18.	19.	SAP HANA +	Relational DBMS	51.77	+2.50	+12.91
19.	19.	15.	Hive	Relational DBMS	49.40	+0.28	-5.87
20.	20.	23.	MariaDB	Relational DBMS	44.09	+1.42	+16.35

从最新的一期榜单来看，数据库产品前三甲非常稳固，与后面差距依然很大。传统商业数据库的代表 Oracle，依旧独占鳌头，但其受到其他数据库逐步蚕食。

有很多人预计在明年 Q1，MySQL 将超过 Oracle 占领首位。MySQL 作为开源数据库的代表，持续在增长；但随着市场稳定已逐步趋于饱和，增长趋缓。

SQL Server 作为另类代表，增长喜人。其不久前宣布的未来将支持 Linux 版本的消息，应该起到了不少提振作用，我们也很期待它在未来的表现。

前十名中以 PostgreSQL、MongoDB、Cassandra、Redis 为代表的开源数据库，各自有其独家绝技，长期来看仍然持续上涨。

其中值得重点关注的 PostgreSQL、MongoDB。PostgreSQL 曾是关系型数据库的学院派代表，近些年更是演化为“全面手”，社区活跃，发展十分迅猛。

MongoDB 作为 NoSQL 阵营里最为接近关系型数据库的代表，发展也很不错。未来不排除这两者出现可挑战前三甲的可能。

前十名中其余产品如 **DB2**、**Access** 等老牌产品，不可避免的出现下滑，未来堪忧。

十到二十名范围内，则是百花齐放，排名变化频繁。值得关注的是 **Elasticsearch**，发展不错，已接近第十名，很有希望下去冲入十名内。

从产品类别角度来看，关系型数据库仍然占绝对主导（前二十名中占了十三位）。其他领域各有优秀代表进入其中，文档型数据库的 **MongoDB**、KV 型数据库的 **Redis**、搜索引擎的 **Elasticsearch**、列式存储的 **HBase** 等。

## SQL、NoSQL 及 NewSQL 的发展历程

上个世纪 70 年代提出了关系型数据库的概念，后来在关系型数据库上设计了一套相应的结构化查询语言称之为 **SQL**，在之后的发展中又引入了事务也就是 **ACID** 的概念，**SQL** 类或者说传统的关系型数据库最典型的一些产品有 **DB2**、**Oracle**、**SQLServer** 以及开源的 **MySQL**、**PostgreSQL** 等，这一代数据库的产品特点可以理解为强 **ACID**、且支持 **SQL** 接口。但这一代数据往往是单机版本，即使有的支持集群其扩展能力也很有限。

随着互联网的发展，很多应用的数据量也越来越大，传统的数据库在数据量越来越大的情况下性能等各方面往往也会出现明显的下降，所以后来才有了 **NoSQL** 的概念，这个概念其实很早就有，但是在 2009 年左右开始变得比较火，**NoSQL** 的概念主要是指非关系型的、不提供 **ACID** 的一种数据库设计理念，这个理念基础之上，很容易提供分布式解决方案，从而比较容易的解决原来传统的 **SQL** 产品面临的一些问题。**NoSQL** 类产品比较典型的有像 **Redis**、**MemcacheDB**、**HBase**、**Cassandra**、**Bigtable**、**LevelDB** 等，这类产品一般是根据 **Key** 来操作相应的 **Value**，有些产品的 **Value** 会比较简单，有些会稍微复杂一些。

**NoSQL** 类产品发展非常迅速，以至于曾经有一段时间大家觉得可能 **SQL** 类产品会很快退出历史舞台，随着 **NoSQL** 广泛使用以后，大家也发现 **NoSQL** 类的产品虽然操作简单，但在使用的时候应用层面要考虑的事情比较多，因为 **SQL** 改 **NoSQL** 很多东西都需要在应用层做拆分及汇总，开发人员心智负担比较重，主要体现在开发的代价会大一点，所以在这同一时期甚至更早一些的时候，大家也有在研究分布式的 **SQL** 解决方案。

分布式的 **SQL** 解决方案简单的说就是像使用传统的 **SQL** 产品一样来使用分布式的数据库产品，最典型的就是一些分库分表的解决方案。这些方案有的是在应用层面，按照一定的规则将数据进行拆分；也有的是通过引入兼容相关 **SQL** 传输协议的中间层透明的完成这种拆分工作，比如典型的一些开源产品有 **Cobar**、**Vitess** 等。

这种方案有些人也会将其归属到 **NewSQL** 的范畴，其实分库分表的分布式 **SQL** 解决方案已经可以满足绝大多数的业务需求，但是有一个不足点是从整体来看这种方式拆分以后的 **SQL** 解决方案在一个事务上下文中如果涉及到多个分片时候往往不能保证严格的事务的 **ACID** 属性。

后来，**Google** 在 2012 年、2013 年分别发表了 **Spanner** 和 **F1** 的论文，告诉业界自己实现了一套既满足 **SQL**，又满足事务的 **ACID** 属性；同时又将 **NoSQL** 的

分布式特性集合到一起的一种实现方式，这也是大家比较认同的一种全新架构的 NewSQL。当前比较典型的一些 NewSQL 的产品，比如像 Google 的 F1/Spanner、阿里的 OceanBase 以及开源的 cockroachdb、Tidb 等。NewSQL 还需要更多生产环境的检验。

以上就是大体的一个 SQL、NoSQL 及 NewSQL 的发展历程。

## RDBMS 家族

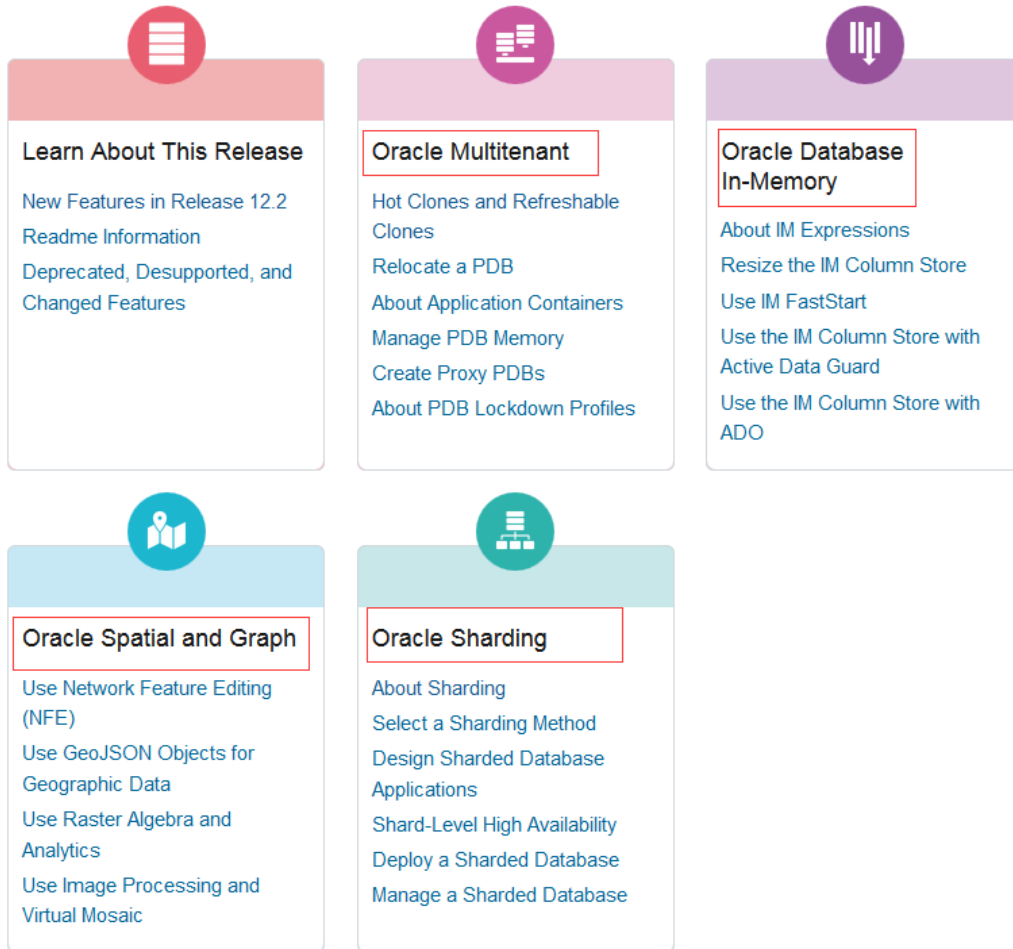
### Oracle

#### 12c Release 2 文档发布

Oracle Database 12c Release 2 文档在 2016 年 11 月中旬公开发布，12c Release 2 离公开下载的日子又近了一步，如果想了解新版本、新特性，可以从文档开始了。

<http://docs.oracle.com/database/122/index.html>

相比于 12c Release 1, Release 2 在 Multitenant, In-Memory, Spatial and Graph, Sharding 几个方面有了加强和改进。



## 12c Release 2 特性解读

Oracle 12c R2 从云端首先支持。下面是整理的一些新特性，发布时间以官方发布时为准。

### Oracle Core Database

Oracle 将针对 Linux 提出一个新的安装版本 Oracle Core Database，这意味着你安装 Oracle Core Database 只需要一条命令 `yum install oracle-core-database`。

与之前 Oracle 发布过的 Oracle Express 版本 Oracle XE 看起来有些相似，但是有巨大的差别，Oracle XE 是免费的，没有官方支持也没有补丁。而 Oracle Core database 是需要 license 许可证，同时官方也提供支持和补丁。

Oracle Core Database 版本将区别于 Oracle 企业版的较高价格，主要是为了让用户能从 sharding 中充分获得横向扩展能力，Core Database 可以大幅降低其 license 成本费用，达到 Oracle EE 企业版的几分之一乃至几十分之一的价格。

## Oracle Sharding

简单来说，Oracle 的 Sharding 技术就是通过分区（Partitioning）技术的扩展来实现的。以前一个表的分区可以存在于不同的表空间，现在可以存在于不同的数据库。不同分区存在于不同数据库，这就将数据隔离了开来，Sharding 就此实现。

在 Sharding 的架构里，存在一个“Shard Directories”目录库来管理 Sharding 的分布，当应用通过 Sharding Key 来访问数据时，连接池就会给出访问路径，快速指向需要访问的 Shard。如果应用不指定分区键访问，则需要通过协调库-Coordinator DB 来协助判定。

## 关联组件 — 全局数据服务（GDS）

在 Oracle 12.1 版本中增加的一个新的产品组件 GDS - Global Data Services，通过 GDS 可以构建一个访问“连接池”，为后端的数据库访问提供代理和路由服务，前面提到的 Shard Directories，正是在 GDS 中配置的。在 12.2 中，Sharding 中的重要作用就日益凸现出来。

## 多租户增强

12.2 中多租户支持更多的 PDB 共存，从上个版本中的 252 增加到 4096 个；在便利性上，支持 Hot Clone, 支持 Refresh，支持在线的 Tenent 转移。PDB 的 Hot Clone 可以让数据库在业务负载运行时进行 Clone 拷贝，并且实时同步变化数据，从而使得数据不断追平，进而实现在线切换，这极大的改善了上云的迁移过程。对用户来说是简化，并且在 OEM 的管理之下，所有工作可以近乎自动的完成。

## IMO 组件更新

In-Memory Option 在 12.2 上也获得了增强，这一特性在 ADG 上的增强使得读写进一步分离，由于 ADG 的只读属性，备库上的内存数据又可以和主库不同，比如备库在内存中可以存储更广泛的数据，实现实时计算。而在性能和易用性上改进也值得称道，In-Memory 在 12.2 中支持根据热图自动向内存进行数据转移，也可以动态的清除冷数据以释放内存空间，简化用户管理。

## Oracle DataGuard 增强

DBCA 备库创建 - 在备库主机安装软件启动监听，则可以通过 DBCA 来创建备库，指向主库来获取文件，创建备库更加简单。

## 口令文件维护增强

在主库上口令文件的修改变更，将自动复制到备库上去，不再需要手工复制文件了。

## AWR 支持远程快照



AWR 支持捕获远程数据库的信息，包括 ADG。在之前的 ADG 中，备库只能通过 Standby Statspack, in-memory ASH, tkprof 来进行性能分析和诊断，现在可以支持 AWR 了。

### 连接保持

在 ADG 中实现了 Failover、SwitchOver 中的连接会话保持，减少了重新连接的开销，极大的改善了用户体验。

### 自动块修复增强

ADG 自动块修复自 11gR2 引入，现在已经非常成熟，修复的类型大大增加。在新版本中，各种类型的坏块支持增量修复。即使是原来 11g 不支持的 datafile header 块，新版本中也可以修复了。

### DataGuard 中并行日志应用

在 12cR2 之前，DG 的备库只能由一个实例通过 MRP 进程进行应用，现在可以在多实例并行进行。多实例应用，可以在所有 Mounted 或者 Open 的实例上并行进行，对比一下单实例应用和多实例应用的架构对比，在常规模式下，多实例的备库，可以有多个 Remote File Server (RFS)进程进行 Redo Thread 的日志接收，但是仅有一个实例进行 Managed Recovery Process (MRP)应用恢复。当然，在单个实例上，仍然可以启动多个 MRP 进程，进行并行的恢复。在 Oracle 12.2 的版本中，多实例并行 MRP 恢复被支持，这一改变将极大的提升 DataGuard 的效率和可用性。

### BIG SCN

在 12cR2 中将引入 BIG SCN 的特性，SCN 的格式将从 6 个字节变成 8 个字节存储，这意味着更高的 SCN 上限 以及能满足更快的 SCN 增速。

### Cross-Endian Dictionary Project

跨 endian 的数据字典项目，使得在 AIX 等 Big Endian 平台上生成的 Oracle 数据字典可以在 Little Endian 上读写，反之亦然。 Cross-Endian Dictionary 让 PDB plug-in 可拔插数据库和表空间传输(XTTS)更方便。另一个方面 Cross-Endian Dictionary 跨 Endian 数据字典让跨 Endian 的 ADG active data guard 变成可能， 这意味着 AIX 上的 primary 库，可以使用 X86 的 pc server 作为 standby

### Other

在 V\$SQL 中加入 redo size 字段，即可以显示每一条 SQL 语句产生的 redo 量。

表和列的名字上限从 30 字符提高到 128 字符。

表在整个移动过程中，仍然是可以访问的。此外，还可以自动更新索引有效。

## MySQL

### MySQL 5.7 新特性解读

关于 5.7 的新特性会从安全性和功能性两方面来解读。

#### 安全性

1. 5.7 版本的用户表 `mysql.user` 要求 `plugin` 字段非空，且默认值是 `mysql_native_password`，并且不再支持 `mysql_old_password`。
2. 增加密码过期功能，DBA 可以设置任何用户的密码过期时间，到期之后必须更改密码，从而增加密码账号的安全性。具体保留时间由 `default_password_lifetime` 控制，如果为 0，则说明永久不过期，生产环境则建议关闭该参数。
3. DBA 可以通过对用户加锁/解锁进一步控制其访问 db，具体例子如：  
`alter user yang@'%' account lock;`
4. MySQL 5.7 版本提供了更为简单 SSL 安全访问配置，并且默认连接就采用 SSL 的加密方式。保障 db 物理文件被拖走之后也
5. 使用更安全的初始化 db 的方式，并且废弃 `mysql_install_db` 的安装方式，使用 `initialize` 代替(`mysql_install_db <5.7.6<= mysqld --initialize`)，使用 `initialize` 参数初始化数据库有如下特性
  - a 只创建一个 root 账号，并且生成一个临时的标记为过期密码。
  - b 不创建其他账号。
  - c 不创建 test 数据库。

#### 功能性

1. 从 MySQL 5.7.8 开始，MySQL 支持原生的 JSON 格式，即有独立的 json 类型，用于存放 json 格式的数据。JSON 格式的数据并不是以 string 格式存储于数据库而是以内部的 binary 格式，以便于快速的定位到 json 格式中值。

在插入和更新操作时 MySQL 会对 JSON 类型做校验，已检查数据是否符合 json 格式，如果不符合则报错。同时 5.7.8 版本提供了多种 JSON 相关的函数，可以方便的访问目标数据而不用遍历全部数据。

  - a 创建: `JSON_ARRAY()`, `JSON_MERGE()`, `JSON_OBJECT()`
  - b 修改: `JSON_APPEND()`, `JSON_ARRAY_APPEND()`, `JSON_ARRAY_INSERT()`, `JSON_INSERT()`, `JSON_QUOTE()`, `JSON_REMOVE()`, `JSON_REPLACE()`, `JSON_SET()`, and `JSON_UNQUOTE()`
  - c 查询: `JSON_CONTAINS()`, `JSON_CONTAINS_PATH()`, `JSON_EXTRACT()`, `JSON_KEYS()`, `JSON_SEARCH()`.
  - d 属性: `JSON_DEPTH()`, `JSON_LENGTH()`, `JSON_TYPE()` `JSON_VALID()`.
2. MySQL 5.7 版本新增了 `sys` 数据库，该库通过视图的形式把 `information_schema` 和 `performance_schema` 结合起来，查询出更加令人容易理解的数据，帮助 DBA 快速获取数据库系统的各种纬度(如谁占用资源最多，实例的

内存分布, 某个 ip 访问占用的 iops 和 io 延迟时间)的元数据信息, 帮助 DBA 和开发快速定位性能瓶颈。详细的信息请参考《官方文档》

<http://dev.mysql.com/doc/refman/5.7/en/sys-schema.html>

3.innodb 表支通过 ALTER TABLE 语句以 in place 方式修改 varchar 的大小且无需 table-copy。但存在限制:表示 varchar 长度的字节数不能变化(如果变更前使用 1 个字节表示长度, 变更后也必须使用 1 个字节表示), 即只支持 0~255 内的或者 255 以上的范围变更(增大), 如果字段的长度从 254 增到 256 时就不能使用 in-place 算法, 必须使用 copy 算法, 否则报错。

4.online DDL 语句重建普通表和分区表

OPTIMIZE TABLE、ALTER TABLE ... FORCE、ALTER TABLE ...

ENGINE=INNODB 等操作支持使用 inplace 算法。减少了重建时间和对应用的影响。

5.支持新的 DATA\_GEOMETRY 空间类型的数据

InnoDB 现在支持 MySQL-supported 空间数据类型。也即, 之前的空间数据是以 binary

BLOB 数据存储的, 现在空间数据类型被映射到了一个 InnoDB 内部数据类型 DATA\_GEOMETRY。

6.innodb buffer dump 功能增强

5.7.5 新增加 innodb\_buffer\_pool\_dump\_pct 参数, 来控制每个 innodb buffer 中转储活跃使用的 innodb buffer pages 的比例。之前的版本默认值是 100%, 当触发转储的时候会全量 dump innodb buffer pool 中的 pages。如果启用新的参数比如 40, 每个 innodb buffer pool instance 中有 100 个, 每次转储每个 innodb buffer 实例中的 40 个 pages。

注意:当 innodb 发现 innodb 后台 io 资源紧张时, 会主动降低该参数设置的比例。

7.支持多线程刷脏页

MySQL 5.6.2 版本中, MySQL 将刷脏页的线程从 master 线程独立出来, 5.7.4 版本之后, MySQL 系统支持多线程刷脏页, 进程的数量由 innodb\_page\_cleaners 参数控制, 该参数不能动态修改, 最小值为 1, 最大值支持 64, 5.7.7 以及之前默认值是 1, 5.7.8 版本之后修改默认参数为 4。当启用多线程刷脏也, 系统将刷新 innodb buffer instance 脏页分配到各个空闲的刷脏页的线程上, 如果设置的 innodb\_page\_cleaners>innodb\_buffer\_pool\_instances,系统会自动重置为 innodb\_buffer\_pool\_instances 大小。

8.动态调整 innodb buffer pool size,

从 5.7.5 版本, MySQL 支持在不重启系统的情况下动态调整 `innodb_buffer_pool_size`。resize 的过程是以 `chunk`(每个 `chunk` 的大小默认为 128M) 的为单位迁移 `pages` 到新的内存空间, 迁移进度可以通过 `Innodb_buffer_pool_resize_status` 查看。记住整个 `resize` 的大小是以 `chunk` 为单位的。`innodb_buffer_pool_chunk_size` 的大小, 计算公式是 `innodb_buffer_pool_size / innodb_buffer_pool_instances`, 新调整的值必须是 `innodb_buffer_pool_chunk_size * innodb_buffer_pool_instances` 的整数倍。如果不是整数倍, 则系统则会调整值为两者乘积的整数倍且大于指定调整的值。

### 9. 升级 `innochecksum`

`innochecksum`--离线的 InnoDB 文件校验工具, 新增新的选择项或扩展的功能, 如可指定特定的校验算法、可以只重写校验值而不进行验证、可指定允许的校验和不匹配量、显示各类页的个数、导出页类型信息、输出至日志、从标准输入读取数据等。从 5.7.2 起可支持校验超过 2G 的文件。详细的用法参考《[innochecksum 官方文档](#)》

### 10. 支持原生的分区表

在 MySQL 5.7.6 之前的版本中, 创建分区表时 MySQL 为每个分区创建一个 `ha_partition handler`, 自 MySQL 5.7. 6 之后, MySQL 支持原生的分区表并且只会为分区表创建一个 `partition-aware handler`, 这样的分区表功能节约分区表使用的内存。对于老版本创建的分区表在升级到新的版本之后怎么处理呢? 莫慌, 5.7.9 之后, MySQL 提供了如下升级方式解决这个问题:

```
ALTER TABLE ... UPGRADE PARTITIONING.
```

### 11. 支持 `truncate undo logs`

MySQL 5.7.5 版本开始支持 `truncate undo` 表空间中的 `undo log`。启用该特性必须设置 `innodb_undo_log_truncate=[ON|1]`。大致原理是系统必须设置至少两个 `undo` 表空间(初始化的时候设置 `innodb_undo_tablespaces=2`) 用于清理 `undo logs` 的切换。该特性的好处是 解决 5.6 之前虽然将 `ibdata` 和 `undo` 表空间隔离, 但是 `undo log` 文件一直增大的问题, 减轻系统的空间使用压力。详细信息参考《[官方文档](#)》

### 12. 新增内置的 Full Text 插件, 支持中文, 韩文, 日文全文索引。

之前的版本 只能依赖单词之间空格进行分词, 对于依赖于语义分词而非空格分词的其他语言种类, 5.7 版本的引入支持解析中文, 韩文, 日文的全文索引--`ngram full-text parser` 解决了该问题。

### 13. 支持在线调整 Replication Filter, 可以 online 调整

`REPLICATE_IGNORE_DB`, `REPLICATE_WILD_IGNORE_TABLE`, `REPLICATE_WILD_DO_DB`, `REPLICATE_WILD_DO_TABLE`, 无需重启 slave。

14.支持多源复制，通过 **channel** 支持一个从库复制多个主库。

15.支持事务级别的并行复制，区别于 5.6 版本基于 **schema** 级别的并行复制，5.7 版本基于 **group commit(slave-parallel-type=LOGICAL\_CLOCK)**做到真正的并行复制，降低了 **slave** 延迟的风险。

16.半同步增强

a 增加 **ack thread** 解决 5.6 版本异步复制主库 **dump thread** 必须等到 **slave** 返回消息之后才能接受新的事务请求，提高事务处理能力。

b 增加 **after sync** 模式，事务是在提交之前发送给 **Slave**（默认，**after\_sync**），当 **Slave** 没有接收成功，并且 **Master** 宕机了，不会导致主从不一致，因为此时主还没有提交，所以主从都没有数据。**MySQL5.7** 也支持和 **MySQL5.5\5.6** 一样的机制：事务提交之后再发给 **Slave**（**after\_commit**）

17.GTID 增强

a 支持在线修改 **gtid** 在线开启 **GTID** 的步骤：不是直接设置 **gtid\_mode** 为 **on**，需要先设置成 **OFF\_PERMISSIVE**，再设置成 **ON\_PERMISSIVE**，再把 **enforce\_gtid\_consistency** 设置成 **ON**，最后再将 **gtid\_mode** 设置成 **on**，如上面所示。

b 存储 **GTID** 信息到表中，**slave** 不需要再开启 **log\_bin** 和 **log\_slave\_updates**。表存在在 **mysql.gtid\_executed**，**MySQL5.6** 上 **GTID** 只能存储在 **binlog** 中，所以必须开启 **Binlog** 才能使用 **GTID** 功能。

## MySQL 分支 Percona

### Percona 5.7.16-10 发布

**Percona Server** 为 **MySQL** 数据库服务器进行了改进，在功能和性能上较 **MySQL** 有显著的提升,**Percona** 在近期推出了 **5.7.16**，该版本基于 **MySQL 5.7.16**。同时 **Percona** 提供了近 300 页的文档说明。

主要的变更如下：

过时的插件：

- [Metrics for scalability measurement](#) 这个特性已被废弃。因为会有一些 **crash** 的情况发生，用户如果安装了这个插件但是还没有使用到会被建议卸载。

Bug 修复：

- 当存储程序会调用如下的管理命令，如 OPTIMIZE TABLE, ANALYZE TABLE, ALTER TABLE, CREATE/DROP INDEX 时，参数 log\_slow\_sp\_statements 的值会被 log\_slow\_admin\_statements 锁覆盖，Bug fixed [#719368](#).
- 如果参数 innodb\_force\_recovery 设置为 6，并行 double write 文件存在的情况下，数据库在崩溃后无法启动，Bug fixed [#1629879](#).
- 线程池中的线程如果溢出导致无法创建线程，这些信息现在会及时反馈到日志中。Bug fixed [#1636500](#)
- 在 TokuDB 中 INFORMATION\_SCHEMA.TABLE\_STATISTICS 和 INFORMATION\_SCHEMA.INDEX\_STATISTICS 这两个表数据不够准确。Bug fixed [#1629448](#).

其他 Bug 修复: [#1633061](#), [#1633430](#), and [#1635184](#).

## MySQL 分支 MariaDB

### MariaDB 10.1.19 发布

MariaDB 是 MySQL 源代码的一个分支，主要由开源社区在维护，采用 GPL 授权许可。2016 年 11 月 7 日发布了 MariaDB 10.1.19 版本，MariaDB 10.1 是目前稳定版本，这在 10.0 版本的基础上有几个全新的特性，还有一些是基于 MySQL 5.6, 5.7 改进和重新实现。

主要变更如下：

- 1.XtraDB 更新至 5.6.33-79.0
- 2.TokuDB 更新至 5.6.33-79.0
- 3.添加了 Ubuntu 16.10 yakkety 软件包，可以使用配置工具添加 MariaDB 的 Ubuntu 软件库到你的系统中。
- 4.修复了以下安全漏洞：  
cve-2016-7440  
cve-2016-5584

更多的变更信息可以参考官网链接：

<https://mariadb.com/kb/en/mariadb/mariadb-10119-changelog/>

## SQL Server

### SQL Server 2016 SP1 补丁包发布

SQL Server 的所有版本在安装了 SP1 补丁包之后将具有以前企业版才有的所有与开发相关的特性,这些特性包括

- 行级加密
- 动态数据掩码
- 更改跟踪
- 数据库快照
- 列存储
- 表分区
- 表压缩
- In Memory OLTP
- 总是加密
- PolyBase
- 数据库审计
- multiple filestream 容器

SP1 补丁包还包含下面一些主要新功能

#### 数据库克隆

DBCC CLONEDATABASE 命令是 DBCC 命令家族的新成员, 这个命令允许你只克隆数据库的表结构、统计信息, query store 元数据, 而不会克隆/拷贝数据, 利用这个克隆可以帮助 DBA 和微软技术支持团队更好的排查数据库问题

#### CREATE OR ALTER 新语法

这个新语法可以使用在存储过程、触发器, 用户定义函数, 视图, 这个功能也是跟某些开源数据库学习所做的改进

#### 新的 USE HINT 查询提示

格式 OPTION(USE HINT('<option>')), 主要用来替代之前只有用 trace flags 才能开启的功能, 并且不需要 sysadmin 权限

#### showplan XML 执行计划

xml 执行计划里增加了内存授予警告, 能够显示单个查询允许的最大内存, 开启的 trace flags 信息

查询所有 cpu 时间, 查询消耗时间, top waits 和参数的数据类型, 对性能调优有很大的帮助

#### 新增的动态管理函数 sys.dm\_db\_incremental\_stats\_properties

sys.dm\_db\_incremental\_stats\_properties 能够显示分区表里每个分区的增量统计信息的情况

#### 新增 EstimatedIRowsRead 属性

在 showplan XML 里新增加 EstimatedIRowsRead 属性, 可以更好排查查询计划是否使用了谓词条件下推

### 对 Bulk insert 大容量插入数据场景开启 TF715 会自动添加 TABLOCK 提示

Trace flag 715 只在没有非聚集索引的堆表生效，当开启 TF715 之后，大容量拷贝数据到一个表的同时 bulk load 操作会申请 bulk update (BU) 锁。在保持 Bulk update (BU) 锁的情况下，SQL Server 会开启多线程并行大容量加载数据到表里，BU 锁可以阻止其他大容量加载线程访问被插入表。SQL Server 2016 以前需要手动指定 TABLOCK 提示才能使用多线程并行大容量加载，现在 SQL Server 2016 已经默认帮用户添加上 TABLOCK 提示，使得 SQL Server 2016 更加人性化。

### 优化 Hekaton 存储引擎的错误日志输出

当使用了内存优化表之后，SQL Server 错误日志会增加非常多的 Hekaton 存储引擎相关的错误日志，甚至淹没整个 SQL Server 的 errorlog，根据社区反馈从 SQL Server 2016 SP1 开始优化并减少 Hekaton 存储引擎的错误日志输出

## SQL Server On Linux 版本发布

目前只是技术预览版 CTP1，并且名字也有所改变，叫做 SQL Server vNext CTP1，多了一个 vNext

安装之后的内部版本号是 14.0，而现在已经发布的 SQL Server 2016 的内部版本号是 13.0

意味着这是下一个版本的 SQL Server，而不是沿用 SQL Server 2016 这一个版本

微软为了在 Linux 上使用 SQL Server，创建了从 Drawbridge 演变而来的 SQL 平台抽象层 (SQLPAL)。

正因为有了这个 SQL 平台抽象层 (SQLPAL) 更加加快了 SQL Server 的移植速度，官方宣布在 2017 年年中 SQL Server vNext 会正式 GA。

目前来看，SQL Server vNext 采用了组件化安装，核心数据库引擎安装包才 138MB，安装和使用都非常方便

对 SQL Server vNext 感兴趣的同学可以到官方网站查看安装教程，安装试玩，地址：<https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-overview>

或者到微软 Azure 公有云，微软 Azure 公有云上已经提供了相应的 SQL Server vNext 虚拟机镜像，地址：<https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-azure-virtual-machine>

## PostgreSQL

### PostgreSQL 9.6 发布

PostgreSQL 9.6 于 2016-09-29 正式发布，包含了许多里程碑式的新功能。



### 1. 多核并行执行

可以利用多核实现 JOIN，全表扫描，聚合操作的加速。达到线性性能提升的效果。

企业的很多报表需求，完全可以在业务低峰时将所有的硬件资源利用起来，快速完成报表的需求。

### 2. 强一致任意多副本同步复制

允许用户选择事务的可靠性级别，配置任意多副本强同步。

例如 1 主 2 备，用户可以要求事务结束的 REDO 发送到至少任意一个备库，才返回给客户端事务结束。

### 3. 增加 remote\_apply 同步级别

对于对读写分离的数据一致性要求非常严格的场景，用户可以将事务的同步级别设置为 remote\_apply，那么事务在结束前，会确保它的 REDO 已经在任意数量（用户配置）的备库 apply 后，才返回客户端。

### 4. 全文检索支持 phrase

全文检索支持文本相似度，支持文本的位置信息，支持按位置匹配，例如'速度与激情'可以与'速度 <1> 激情'匹配。实现精准的匹配规则。

### 5. 全文检索拼写字典增强

支持拼写纠正，例如 stories, story, big, bigger, biggest, doing, do, done, hack, hacking, hacked 都会转换为原来的单词。

### 6. postgres\_fdw 支持 JOIN , SORT , UPDATE , DELETE 下推至远程数据库

通过 postgres\_fdw，可以支持数据库的 sharding，计算和 filter 条件都可以下推到远程数据库执行。

10.0 的版本已经增加了聚合的下推。

### 7. 高并发性能增强

可以充分发挥硬件的性能，满足高并发时的高 TPS 需求，单机(72HT)实现 180 万 tps。

### 8. VM 增加 freeze bit 识别

增加 VM 的 freeze 识别，从而大幅降低静态历史数据的 FREEZE 的扫描。

### 9. 支持等待事件

新增了 69 个等待事件的监控，用户可以根据等待事件统计信息，找出数据库的瓶颈。

### 10. 排序增强

排序的内核性能增强，充分利用 CPU CACHE。

#### 11. snapshot time out

支持快照过旧。

#### 12. IO 调度优化

支持检查点排序，降低离散 IO。

支持 checkpoint, bgwriter, wal writer, backend writer 进程的平滑 IO 调度策略。

#### 13. 相同算子的聚合子函数复用

通过复用同算子的聚合子函数，在多个聚合中，性能提升非常明显。

#### 14. 支持 bloom 任意列索引

支持对整行的所有列建立一个 bloom 索引，支持任意列的组合查询，满足前端无法固定的多变的查询条件需求。

更多材料可以参考

<https://github.com/digoal/blog/blob/master/201610>

<https://www.postgresql.org/docs/9.6/static/release-9-6.html>

<https://yq.aliyun.com/articles/51131>

## PostgreSQL 9.6.1 patch 发布

2016-10-27 发布了 9.6 的第一个 patch，以 fix 为主。

## DB2

### DB2 LUW V11.1 发布

2016 年 6 月 15 日，IBM 正式发布 DB2 LUW 新版本 11.1，提供了满足各种业务需求的新功能部件和增强功能，从而使数据库更有效率、更简化且更可靠。全面的企业安全性、简化的安装和部署、更高的易用性和适用性、顺利的升级过程、对超大型数据库的增强功能以及对 BLU 加速的显著改进是此技术提供的主要益处。

#### 1.对分区数据库环境的按列组织的表支持

下列特定增强功能与分区环境中按列组织的表有关：

- 列矢量处理引擎的 MPP 感知查询计划和优化
- 分区之间纵列数据交换的优化向量格式

- 允许数据跨网络保持压缩状态的公共表字典
- 为实现多核心并行性而设计的优化通信基础结构
- 在 **DPF** 多分区环境中支持列存储表，在多服务器集群环境中分析型负载的扩展性得以大幅提升；获取的益处包括：最少的建模设计和查询调优、行业领先的数据压缩、分析型查询性能大幅提升等
- 列存储表上支持 **NOT LOGGED INITIALLY** 语句，针对列表的 **INSERT, UPDATE, DELETE** 语句运行速度更快；
- **INSERT / INGEST / IMPORT** 过程中自动字典生成（**Automatic Dictionary Creation, ADC**）序列化得到增强，数据字典得以更早生成，大数据量表初始化操作（使用 **INSERT ... SELECT...** 方式）的压缩率（**PCTENCODED**）更高、查询性能更好、高并发作业的吞吐量更佳；
- 高并发作业的性能和扩展性增强，频繁访问公共数据页面的高并发作业性能得以增强，小表查找和频繁索引页面访问（如嵌套表关联时）的门闩（**DB2 Latch**）明显降低，交易延时和锁等时间明显降低，这些改进可以明显提升高并发作业的扩展性。
- 行列安全控制（**RCAC**）支持列存储表，将传统行存储表上的安全控制机制推广到列存储表，控制用户访问具有权限的数据行、数据列或数据单元，是对数据库授权机制的补充；

## 2.企业加密密钥管理

- 原生数据库加密支持集中式密钥管理，支持集中式密钥管理器来存放原生加密主密钥，支持任何使用 **KMIP 1.1** 或以上版本的任何密钥管理产品。还通过已与 **DB2 V11.1** 集成的企业密钥管理来支持硬件安全性模块（**HSM**），以向用户提供各种 **HSM** 选项

## 3.DB2 pureScale 特性增强

- 支持 **HADR** 的同步和准同步模式；
- 统一工作负载管理，支持定义成员子集的备用成员；
- **GDPC** 双活增强，支持 **ROCE** 和 **TCPIP**，支持 **AIX, REDHAT** 和 **SUSE** 操作系统，每个 **CF** 和 **MEMBER** 支持多卡多口以提升带宽和增强可用性；每个站点支持双交换机模式；
- **GPFS** 复制增强，通过 **db2cluster** 命令进行 **GPFS** 复制配置，简化 **GDPC** 双活配置过程；
- **pureScale** 健康检查，**db2cluster -verify** 命令用于 **pureScale** 产品安装后的健康检查，包括：**TSA** 和 **GPFS** 集群的配置检查，**member** 和 **CF** 见通讯检查，文件系统复制检查，文件系统中每个磁盘状态检查等。

## Greenplum

**Greenplum** 数据库是由 **Pivotal** 公司基于 **Postgres** 开发的分布式数据库。架构采用了 **MPP** 无共享架构，集群中的每个节点都有自己的资源，节点之间的信息交互是通过节点互连网络实现的。**Greenplum** 数据库支持高并发，线性扩展，高可用性，向客户提供高性能的超级数据引擎，并将强大的并行计算能力融入到大规模数据仓库分析领域中。

2016 年 10 月，**Greenplum 4.3.10.0** 引入了 **S3** 可写表，开启了 **Greenplum** 对亚马逊 **S3** 的全面读写支持。用户可以非常方便地在亚马逊云平台上使用到 **Greenplum** 大数据服务，并且可以快速读写存储在 **S3** 上的数据。近日，**Pivotal** 又刚刚宣布了 **Greenplum** 数据库对微软 **Azure** 云平台的支持，用户可以在微软 **Azure** 云平台上方便地使用到 **Greenplum** 提供的各项专业高效的大数据服务。

### GPCC3.0 发布

2016年11月，新版GPCC3.0的发布，给用户提供了一个更好的管理 **Greenplum** 数据库的工具。管理员可以通过GPCC查询系统的性能指标，查看当前在系统中运行的查询，监控资源使用情况，并且方便问题诊断和报告。**GPCC3.0**改进的部分包括但不限于：

- **GPCC3.0**彻底抛弃了过时的Flash技术，使用了对浏览器更加友好和安全的HTML5方案。
- 采用**Pivotal UI** 风格，与**Pivotal**相关产品保持了外观、操作等的一致性。
- 采用响应式布局设计，增强了与图表间的交互式操作。
- 采用Go语言设计开发了RESTful API的服务端，实现了平台无关性，优化了查询性能，降低了GPCC对**Greenplum**系统的资源占用。
- 采用测试驱动开发，并使用持续集成保证产品的质量，实现产品的持续交付。

### GPText 2.0 发布

2016 年 12 月，**GPText 2.0** 发布，集成了 **SolrCloud 6.1**，增强了高可用特性，提供了启动、配置、扩展、备份、恢复等管理工具。**GPText** 是 **Greenplum** 的一个扩展，它深度集成 **SolrCloud** 企业搜索功能和 **Madlib** 数据分析库，结合 **Greenplum** 的大规模并行处理机制为用户提供快速、便捷的文本搜索和分析服务，用户只需要通过熟悉的 **SQL** 命令调用相关的 **UDF** 函数就可以完成文本数据的索引、查询和分析。

最近的发布持续改进查询优化和查询执行，特别是基于 **ORCA** 的新 **Pivotal Query Optimzer**，对于多个连接操作的语句、带有子查询的语句和包含 **IN** 的语句的优化可以减少查询执行时间，对于查询预处理的改进可以减少复杂查询的优化时

间。也包含了大量工具如 gpfdist, gpcheckcat、gpcrondump 等性能和功能的增强。

## NoSQL 家族

### MongoDB

#### MongoDB 3.4 版本发布

MongoDB 近日发布最新版本 3.4! 以下是 3.4 的新功能一览。

更多内容可以参考 MongoDB 中文社区

<http://www.mongoing.com/archives/3586>

#### 分片集群 (Sharded Cluster)

##### Sharding Zones

分片集群里引入了 Zone 的概念, 主要取代现在的 tag-aware sharding 机制, 能将某些数据分配到指定的一个或多个 shard 上。

##### Faster Balancing

- \* 使用 wiredtiger 引擎时, moveChunk 的 secondaryThrottle 选项默认设置为 false, 即不用等待迁移的数据复制到 secondary 节点

- \* 支持并行的 chunk 迁移, 对于包含 N 个 shard 的 sharding 集群, MongoDB 最多可以同时跑 N/2 个迁移任务。

#### 复制集 (Replica Set)

##### 支持 Linearizable Read Concern

"linearizable" Read Concern 级别保证, 一定能读到 WriteConcern 为 majority, 并且确认时间在读请求开始之前的数据, 该级别仅在查询结果只有单个文档的情况下有效。

##### 提升全量同步

- \* 在拷贝数据的时候, 同时建立所有的索引 (以前版本只有 \_id 索引是在同步数据时建立的)

- \* 拷贝数据的阶段, secondary 不断拉取新的 oplog, 确保 secondary 的 local 数据库有足够的空间来存储这些临时数据。

##### Decimal Type

MongoDB 3.4 新增对[decimal128 format]的支持, 最多支持 34 位小数位。

#### Aggregation

MongoDB 在 3.4 版本增加了大量的 aggregation 操作符，特别是 Facet Search 和 Graph Lookup

### Collation and Case-Insensitive Indexes

MongoDB 3.4 开始支持

[collation](https://docs.mongodb.com/master/reference/collation/), 在之前的版本里，文档里存储的字符串，不论是中文还是英文，不论大小写，一律按字节来对比，引入 collation 后，支持对字符串的内容进行解读，可以按使用的 locale 进行对比，也支持对比时忽略大小写。

### 视图 (Views)

MongoDB 3.4 里增加了对[只读视图](https://docs.mongodb.com/master/core/views/#reference-views)的支持，视图将集合里满足某个查询条件的数据虚拟成一个特殊的集合，用户可以在特殊的集合上做进一步的查询操作。

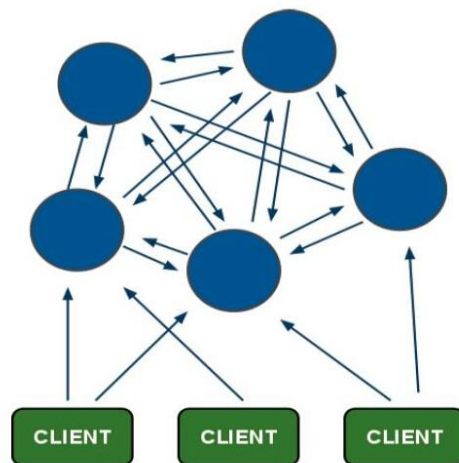
### 工具 (MongoDB Tools)

MongoDB 3.4 引入 mongoreplay 工具，可用于监控并记录 mongod 上执行的命令并 replay 到另一个 mongod 实例上，该工具可用于代替 mongosniff。

对新版本功能感兴趣的同学可以下载 MongoDB-3.4 来试玩下。

## Redis

提到 Redis 不得不提 Redis 的 3.0 版本，该版本于 2015-4-1 GA，它的 GA 代表着 Redis Cluster 已经足够成熟，对传统“ZK+twemproxy+Redis+Redis-sentinel”这种复杂缓存架构带来较大的冲击。Redis cluster 的架构如下图：



Redis Cluster 特性如下：

- (1) 高可用：集群自动恢复机制(强调一点就是每个主节点必须有从节点)
- (2) 高性能：集群不需要 proxy 代理，没有单 redis 节点中复杂的 merge 操作
- (3) 集群的扩展性，官方文档给出的是它能线性扩展到上千个节点。
- (4) 丰富的集群命令：cluster addslots/delslots, cluster meet, cluster nodes, cluster info, cluster replicate、cluster setslot 等

Redis Cluster 高可用、高性能、易扩展等特性，使得 Redis Cluster 已经越来越多的应用在各大互联网的生产环境。

相比于之前的 Redis2.8 版本，3.0 版本还有以下的新增或者优化：

- (1) 提升了 AOF 重写的速度；
- (2) 升级 LRU 算法让 keys 更好的回收；
- (3) 迁移连接缓存从而能更快的迁移 keys，并且 migrate 操作提供了 copy/replace 选项；
- (4) 灵活的内存设定：使用 config set 设定内存时接受不同的内存值表达，例如 config set maxmemory 1gb
- (5) client pause timeout 命令暂停客户端的命令
- (6) incr 操作的性能提升等

## Redis 3.2 版本发布

Redis 最新版本为 3.2.6(2016 年 12 月 6 日)，Redis 3.2 这个大版本相对于 3.0 版本，最重要的是增加了 GEO 地理信息存储，底层是采用有序集合(sorted set)的方式存储，可以使用丰富的 GEO 命令实现“摇一摇”、“附近的人”等基于地理位置信息的功能。

除了上面的 GEO，该版本还在下面几个方面进行了优化：

- (1) Redis cluster:migrate 性能提升 10 倍；Redis cluster slots rebalancing
- (2) 在 API 接口实现上发生了重要的变化
- (3) 新增 BITFIELD 命令
- (4) 内存优化：同样的数据存储比之前消耗更少的内存。
- (5) Lua 脚本优化：复制功能以及远程 Lua 脚本调试器编写简单任务

## HBase

HBase - Hadoop Database，是一个高可靠性、高性能、面向列、可伸缩的分布式存储系统，利用 HBase 技术可在廉价 PC Server 上搭建起大规模结构化存储集群。

HBase 是 Google Bigtable 的开源实现，类似 Google Bigtable 利用 GFS 作为其文件存储系统，HBase 利用 Hadoop HDFS 作为其文件存储系统；Google 运行

MapReduce 来处理 Bigtable 中的海量数据，HBase 同样利用 Hadoop MapReduce 来处理 HBase 中的海量数据；Google Bigtable 利用 Chubby 作为协同服务，HBase 利用 Zookeeper 作为对应。

## Apache HBase v1.2.4 发布

Apache HBase v1.2.4 于 2016 年 11 月 7 日发布，HBase 1.2.4 是 HBase 1.2.x 系列中的第四个维护版，在该版本中修复了自 1.2.3 发行以来的 35 个 BUG，主要更新：

### Sub-task

[HBASE-14734] - BindException when setting up MiniKdc

[HBASE-15984] - Given failure to parse a given WAL that was closed cleanly, replay the WAL.

[HBASE-16101] - Procedure v2 - Perf Tool for WAL

[HBASE-16522] - Procedure v2 - Cache system user and avoid IOException

[HBASE-16970] - Clarify misleading Scan.java comment about caching

### Improvement

[HBASE-15315] - Remove always set super user call as high priority

[HBASE-16033] - Add more details in logging of responseTooSlow/TooLarge

[HBASE-16667] - Building with JDK 8: ignoring option MaxPermSize=256m

[HBASE-16972] - Log more details for Scan#next request when responseTooSlow

[HBASE-17004] - Refactor IntegrationTestManyRegions to use @ClassRule for timing out

[HBASE-17006] - Add names to threads for better debugability of thread dumps

[HBASE-17165] - Add retry to LoadIncrementalHFiles tool

### Task

[HBASE-16340] - ensure no Xerces jars included

[HBASE-16518] - Remove old .arcconfig file

[HBASE-16748] - Release 1.2.4

[HBASE-16749] - HBase root pom.xml contains repo from people.apache.org/~garyh

### 完整更新日志：

[https://issues.apache.org/jira/secure/ReleaseNote.jspa?version=12338116&styleName=&projectId=12310753&Create=Create&atl\\_token=A5KQ-2QAV-T4JA-FDED%7C4402d8283e6ee19870bd680ad801c940e7e64bc7%7Cclout](https://issues.apache.org/jira/secure/ReleaseNote.jspa?version=12338116&styleName=&projectId=12310753&Create=Create&atl_token=A5KQ-2QAV-T4JA-FDED%7C4402d8283e6ee19870bd680ad801c940e7e64bc7%7Cclout)

下载地址：<http://apache.fayea.com/hbase/1.2.4/hbase-1.2.4-bin.tar.gz>



## Geode

### Apache Geode 1.0 版本正式发布

在2016年11月21日 Apache 基金会宣布 GemFire的开源版本 Apache Geode 已从 Apache 孵化器毕业, 正式成为 Apache 的顶级项目, Apache Geode 1.0 正式发布.

Geode增强了以前 GemFire 分布式系统的高可用和弹性收缩功能, 提升了它的管理和监控功能。

#### 高可用功能增强

在分布式系统中, 由于网络故障导致, 成员节点的失联比较难于处理. 在Geode 中, 增加了

使用多播发现系统成员.

在出现大面积失联的情况下, 集群中的节点可以自动重连协调器.

同时, Geode加强了Client API 自动重连集群的功能, 同时在断开连接的时间段, 客户端向服务器发送的数据保持不丢失.

另外, Geode添加新的 API 可以对 Gateway 网关接收和发送数据进行负载均衡.

#### 管理监控功能增强

Geode 新增加了易于使用的管理监控工具, 可以轻松地对 Geode 的性能进行监控.包括对统计当前的状态和历史性能, 提供命令行直接对集群中的节点进行管理.

Geode有几个重要的更新:

使用 Apache Log4j 2 更新内部日志系统

增强了 SSL 安全配置和 LDAP 认证配置

增强了PDX 序列化配置, 包括使用别名重命名 PDX 类

增强了 Geode查询功能和索引创建, 包括索引提示和创建多索引的功能

增强了Geode对内存数据的批处理功能

增加了REST API Endpoints 和 SSL 配置功能

官方网站地址: <http://geode.apache.org/>

## NewSQL 家族

### TiDB

- TiDB 是基于 Google Spanner & F1 实现的分布式 NewSQL 数据库，目标定位支持 100% 的 OLTP + 80% 的 OLAP，除了底层的 RocksDB 存储引擎之外，分布式 SQL 解析层、分布式 KV 存储引擎（TiKV）完全自主设计和研发。
- TiDB 是开源且网络接口和语法 MySQL 兼容的，可以简单理解为一个可以无限水平扩展的 MySQL，提供分布式事务、跨节点 JOIN、保证跨数据中心的数据的强一致性（ACID 跨行事务支持）、故障自恢复的高可用、提供更快的查询和写入吞吐；对业务没有任何侵入性，简化开发，利于维护和平滑迁移。

### TiDB 近期将发布 RC1 版本

TiDB 目前 版本 Beta 4，近期即将发布 RC1 版本，总体目标是让 TiDB 在性能、稳定性、安全性、文档方面达到可商用级别。

RC1 的几个主要改进：

1. SQL 优化器改进，进一步下推更多算子，提升 MPP 性能
2. 基于 myloader / mydumper 的数据导入提速
3. 提升稳定性，完善了过载保护机制
4. 与 Kubernetes 整合，实现在 K8S 上的数据库编排部署，以及一键部署工具
5. 实现 TiDB 与 MySQL 互为主备的实时数据同步方案

更进一步文档请阅读: <https://github.com/pingcap/docs-cn>

## RethinkDB

### RethinkDB 深情告别

RethinkDB 的开发工作始于 2009 年 7 月份，用 C++、JavaScript 和 Bash 编写。它采用 AGPL 开源许可证来发布。最后一个稳定版本是 2016 年 5 月 2 日发布的版本 2.3.2。

近期 RethinkDB 对其为实时 Web 开发的开源、可扩展的 JavaScript 对象标注（JSON）数据库进行了一场深情告别。

该公司关闭后，RethinkDB 团队随之搬家，预计工程团队会加入 Stripe Inc.，而 Stripe 是完全成熟的软件平台和工具包，面向在线支付。

## 大数据生态圈

### Hadoop

#### Hadoop 3.0.0 Alpha 版本发布

由于 Hadoop 2.0 是基于 JDK 1.7 开发的，而 JDK 1.7 在 2015 年 4 月已停止更新，所以 Hadoop 社区于 2016 年 9 月 3 日发布了全新的基于 JDK 1.8 的 Hadoop 3.0.0 alpha 版本

目前 **hadoop 3.0.0** 仅提供 **alpha** 测试版本，整个系统的稳定性没有保障，所以不推荐在正式的开发系统中使用该版本。

相比于 hadoop2.7.0，3.0.0 版本更新的主要变更的官方链接：

<http://hadoop.apache.org/docs/r3.0.0-alpha1/index.html>

全部变更的官方连接：

<http://hadoop.apache.org/docs/r3.0.0-alpha1/hadoop-project-dist/hadoop-common/release/3.0.0-alpha1/RELEASENOTES.3.0.0-alpha1.html>

我们针对主要的功能点版本进行了翻译，如下：

#### 1、Java 的最低版本要求从 Java 7 改成了 Java 8

3.0 以后所有的 Hadoop JARs 都是基于 Java 8 进行编译的，所有还在使用 Java 7 或者是更低版本的用户，应该升级到 Java 8。

#### 2、HDFS 支持纠删码（Erasure Coding, EC）

Erasure coding 纠删码技术简称 EC，是一种数据保护技术。最早用于通信行业中数据传输中的数据恢复，是一种编码容错技术。他通过在原始数据中加入新的校验数据，使得各个部分的数据产生关联性。在一定范围的数据出错情况下，通过纠删码技术都可以进行恢复。

与副本技术相比，纠删码是一种能够显著的节约空间的用于存储持久化数据的方法。进行数据存储的时候，标准编码（比如 **Reed-Solomon (10,4)**）有着 1.4 倍的空间开销，而如果使用 **HDFS** 的副本模式已经数据存储的时候，有着 3 倍的空间开销。由于纠删码在进行重建和远程读的时候会有额外的开销，所以它一般情况下，被用于存储冷的，不经常被访问的数据。用户在使用纠删码这一功能的时候应该要考虑 CPU 以及网络的额外开销。

关于 **HDFS** 的纠删码的更多细节，可以访问官方文档或者 Jira:

<http://hadoop.apache.org/docs/r3.0.0-alpha1/hadoop-project-dist/hadoop-hdfs/HDFSErasureCoding.html>

<https://issues.apache.org/jira/browse/HDFS-7285>

国内也有相关的技术博客文章:

<https://www.iteblog.com/archives/1684>

### 3、YARN 时间轴服务 V.2:

在 **hadoop2.4** 版本之前对任务执行的监控只开发了针对 **MR** 的 **Job History Server**，它可以提供给用户用户查询已经运行完成的作业的信息，但是后来，随着在 **YARN** 上面集成的越来越多的计算框架，比如 **spark**、**Tez**，也有必要为基于这些计算引擎的技术开发相应的作业任务监控工具，所以 **hadoop** 的开发人员就考虑开发一款更加通用的 **Job History Server**，即 **YARN Timeline Server**

**3.0** 版本引入了 **YARN Timeline Service: v.2**，主要用与解决两大挑战:

1) 提高时间轴服务的稳定性和可拓展性。

2) 增强流式和聚合的可用性

更多关于 **YARN Timeline Service: v.2** 可以访问官方文档

<http://hadoop.apache.org/docs/r3.0.0-alpha1/hadoop-yarn/hadoop-yarn-site/TimelineServiceV2.html>

### 4、Shell 脚本重写

The Hadoop shell scripts have been rewritten to fix many long-standing bugs and include some new features. While an eye has been kept towards compatibility, some changes may break existing installations.

为了修复一些长时间存在的 **BUG** 并添加一些新的功能，**Hadoop** 的 **shell** 脚本在 **3.0** 版本被重新进行了改写。其中绝大部分的脚本能够保证一定的兼容性，但是有些变化还是会现有的一些配置出现问题。

相关不兼容的变更我们可以查看:

<https://issues.apache.org/jira/browse/HADOOP-9902>

如果你想了解 **3.0** 版本中脚本的更多细节我们可以访问 **Unix Shell** 用户指导及 **API** 手册:

<http://hadoop.apache.org/docs/r3.0.0-alpha1/hadoop-project-dist/hadoop-common/UnixShellGuide.html>

<http://hadoop.apache.org/docs/r3.0.0-alpha1/hadoop-project-dist/hadoop-common/UnixShellAPI.html>

### 5、MapReduce 任务级别的本地优化

MapReduce 添加了 C/C++输出到 collector 的本地实现(包括了 Spill,Sort,和 Ifile 等), 通过作业级别的参数调整就能够切换到该实现上对于 shuffle 密集型的作业来说, 可以提高 30%以上的性能。

更多细节可以查看:

<https://issues.apache.org/jira/browse/MAPREDUCE-2841>

## 6、支持两个以上的 namenode

早期版本的 HDFS namenode 的高可用提供了一个 active 状态的 namenode 和一个 standby 状态的 namenode, 他们之间通过 3 个 JournalNode 应用 edit 的方式进行同步。在这种架构下, 允许两个节点中的任意一个节点失效。有的时候, 我们的系统可能需要更要的容错性, 这个时候我们就需要 3.0 的多 namenode 的新特性了。它可以允许用户配置多个 standby namenode, 如果说, 配置说三个 namenode, 5 个 JournalNode, 这样这个系统就允许坏两个 namenode 了, 进一步提高了系统的冗余性。

如何进行配置多个 namenode 请查看如下文档:

<http://hadoop.apache.org/docs/r3.0.0-alpha1/hadoop-project-dist/hadoop-hdfs/HDFSHighAvailabilityWithQJM.html>

## 7、多个服务的默认端口的变化

在此之前, 很多 Hadoop 的默认端口都是属于 linux 的临时端口范围 (32768-61000) 这意味着我们的服务在启动的时候由于端口已经被其他的服务所占用导致启动的失败。现在这些可能会引起冲突的端口都已经做了变更, 不再是在临时端口范围内了, 这些端口的变化会影响 NameNode、SecondaryNameNode、DataNode 和 KMS。这些变化, 在 Jira 中都有说明, 详见:

<https://issues.apache.org/jira/browse/HDFS-9427>

<https://issues.apache.org/jira/browse/HADOOP-12811>

## 8、支持 Microsoft Azure Data Lake filesystem 连接器

现在 Hadoop 支持 Microsoft Azure Data Lake filesystem, 并可替代作为 Hadoop 默认的文件系统

## 9、Intra-DataNode 均衡器

一个 datanode 可以管理多个磁盘, 正常写入, 各个磁盘会被均匀写满。然后, 后期当重新添加或者替换磁盘的时候, 就会导致 DataNode 内部的数据存储严重倾斜。这种情况下, HDFS 的 balancer 是无法进行处理的。这种情况下, 通过调用 dfs diskbalancer CLI, 可以实现 Intra-DataNode 的动态 balance。

更多细节详见:

<http://hadoop.apache.org/docs/r3.0.0-alpha1/hadoop-project-dist/hadoop-hdfs/HDFSCommands.html>

## 10、重写守护进程及任务的堆管理

<https://issues.apache.org/jira/browse/HADOOP-10950>

介绍了配置守护进程 Heap 大小的新的方法，现在已经可以实现基于主机内存的动态调节，同时 HADOOP\_HEAPSIZE 此参数已经被弃用。

<https://issues.apache.org/jira/browse/MAPREDUCE-5785>

介绍如何去配置了 Map 以及 Reduce task 的 heap 大小。现在所需的堆大小不在需要通过任务配置或者 Java 选项来实现，已经指定的配置选项不受变更的影响。

## Druid

### Druid 0.9.2 版本发布

Druid 是一款开源的分布式分析型数据存储系统，它最大的卖点是实时 OLAP，导入即可查询，OLAP 查询亚秒级响应，数据量可以扩展到 PB 级，其高并发以及高可用等特性适合构建面向用户的数据分析产品。

2016 年 12 月 1 日，Druid 发布了 0.9.2 版本。在这一版本中包含了来自 30 多位 Contributor 贡献的 100 多项性能提升、稳定性提升以及 Bug 修复。主要的新功能包括新的 Group By 查询引擎，数据摄入阶段可以关闭 Roll-up，支持对 Long 类型字段的过滤，Long 类型字段的新编码方式，HyperUnique 和 DataSketches 的性能提升，基于 Caffeine 的查询缓存实现，可配置缓存策略的 Lookup 扩展，支持 ORC 格式数据的摄入以及增加了标准差和方差的 Aggregator(聚合函数)等。

#### 新的 Group By 查询引擎

Druid 为了提升性能以及优化内存管理，彻底重写了 Group By 查询引擎。在官方测试数据源上进行基准测试，有 2~5 倍的性能提升。新的查询引擎依然支持严格限制内存使用大小，同时增加了在内存耗尽时溢写到磁盘的选项，这样避免对结果集大小的限制以及潜在的内存溢出异常发生。新引擎模式默认是不开启的，你可以通过修改查询上下文中的相应参数开启，在 Druid 的未来发行版本中将默认采用新的查询引擎。其实现细节，详见

<http://druid.io/docs/0.9.2/querying/groupbyquery.html#implementation-details>。

#### 禁用 Roll-up 的选项

从一开始，Druid 在数据摄入阶段以及查询阶段，分区维度和度量值。Druid 是唯一仅有的数据库，支持在数据摄入阶段进行聚合，我们把这一行为叫做

“Roll-up”。Roll-up 可以显著地提升查询性能，但会失去查询原始数据的能力。禁用 Roll-up 以后，会将原始数据摄入，但不会影响查询阶段的聚合。Roll-up 默认依然是开启的，具体配置详见 <http://druid.io/docs/0.9.2/ingestion/index.html>

#### 对整数类型列的筛选能力

Druid 新增了对整数类型列的复杂筛选能力，整数类型列包括 long 类型的度量值以及特殊的 \_\_time(时间)列。以它为基础，将开创一系列新功能：

- 对时间列的筛选聚合，它对于实现对比两个时间段的查询非常有帮助，例如使用 **DataSketches** 实现用户留存。
- 针对整数类型列进行筛选，特别是禁用 **Roll-up** 以后。

### Long 类型字段的新编码方式

直到现在，Druid 中 Long 类型字段，包括 Long 类型度量值以及特殊的 \_\_time 列，都是采用 64bit 8 字节 longs 存储，然后使用 LZ4 按块压缩。Druid0.9.2 增加了新的编码方式，它会在很多场景下，减少文件大小以及提升性能。它有如下 Long 类型编码方式

- auto，会在扫描一遍数据以后，得到其基数以及最大值，选择 table 或者 Delta 编码方式，使得存储使用小于 64bit 存储每一行。
- longs，默认的策略，总是使用 64bit 存储每一行。

新增加了“none”的压缩策略，类似原来的“uncompressed”，摒弃了原来块存储的方式，提升了随机查找的性能。

默认仍然采用“longs”编码方式+“lz4”压缩方式。在官方测试中，产生有效收益的两个选项是 "auto" + "lz4"(通常文件大小会小于 longs+lz4)和 "auto" + "none"（通常会比 longs+lz4 块，但文件大小影响各不相同）。

### Sketch 性能提升

DataSketch 性能最多提升 80%，HyperUnique 性能提升 19%~30%

## Kudu

### Apache Kudu 1.1.0 正式发布

Kudu 是 Cloudera 开源的新型列式存储系统，是 Apache Hadoop 生态圈的新成员之一（incubating），专门为了对快速变化的数据进行快速的分析，是对 HDFS 与 HBase 的补充。

#### 新功能和改进：

1. Python 客户端已经与 C++和 Java 客户端具有同等特性。
2. 列表谓词。
3. Java 客户端现在具有客户端跟踪功能。
4. Kudu 现在发布用于使用 Scala 2.11 编译的 Spark 2.0 的 jar 文件。
5. Kudu 采用 Raft 算法实现预选举，在我们的测试中这个功能大大提高了稳定性。

更多内容可以参考链接：<http://kudu.apache.org/2016/11/15/weekly-update.html>

## HAWQ

Apache HAWQ 是基于 HDFS 存储上的 MPP 引擎，兼备 MPP 并行数据库的查询处理优势和 HDFS 存储的可扩展性优势。此外，HAWQ 还提供了 Madlib 数据挖掘工具，提供了对 R，Python，Perl 的脚本嵌入，使得用户可以方便地开发兼容第三方语言和 SQL 的软件。

### HAWQ 2.1.0.0 企业版正式发布

2016 年 12 月，基于 Apache HAWQ 2.0 的 Pivotal HDB 2.1.0.0 企业版正式发布，功能包括：

#### 一、资源管理及弹性查询执行

**1. 资源管理与 YARN 集成：**可动态从 YARN 申请资源并在负载下降时动态归还。支持查询级别的资源管理，可使用多级资源管理队列分配用户以及查询使用资源。自动对资源进行隔离及强制。

**2. 弹性查询执行：**基于 virtual segment 实现弹性查询执行。基于查询的代价动态分配 virtual segment，使其运行在集群部分节点。解决了经典 MPP 引擎缺点，提高扩展性和查询并行度。

**3. 动态集群伸缩：**彻底解耦了计算和存储，Segment 节点和 Master 节点。这意味着在集群节点大小变化的时候表不需要重新分布，从而 HAWQ 可以实现秒级扩容。

**4. 新版调度器：**新调度器可以动态组合不同节点上的查询执行进程来动态调度和管理查询的执行。

#### 二、存储相关

**1. 块级别存储：**优化了 AO 以及 Parquet 表存储格式，使得一个大文件可以被多个 Scanner 进程以块级别分割来读。使得 HAWQ 可以达到优化的查询并行度。

**2. 表数据单目录存储：**将一个表的所有数据文件存放在 HDFS 的一个目录中，方便了 HAWQ 与外部系统的数据交换，比如外部系统可以把数据文件放到该表对应 HDFS 目录中，实现数据加载。

**3. HDFS 元数据缓存：**HAWQ 需要 HDFS 文件块位置信息来实现派遣计算到数据的调度。HDFS 在 RPC 的处理速度上在高并发时往往响应较慢。元数据缓存提升了数据局部性匹配方法的效率。

**4. PXF 功能增强：**实现与 HCatalog 集成。基于 PXF 对 Hive 表的访问不需要先创建外部表；支持将 Hive 中 ORC 存储格式文件作为外部表进行查询，同时在外表处理中支持谓词下推和投影下推。

**5. HAWQ Register 功能支持：**HAWQ Register 能够直接将 HDFS 文件注册到 HAWQ 内部表。可被用于：直接注册 Hadoop 生态系统其他模块生成的 parquet 存储格式文件；HAWQ 集群的迁移。



### 三、其他功能改进及增强

**1. 优化器(gporca)升级:** 新版 gporca 查询优化器包括了许多新功能和大量的错误修正。

**2. 新的容错服务:** 基于心跳和动态探测的容错服务, 可动态检测到新加节点并且移除失效节点。

**3. 新管理工具接口:** 用户可以通过一个新的命令“hawq”来完成几乎所有的管理操作, 该命令整合了所有的其它子命令。

### 四、其他模块集成与支持

**1. HDP 2.5 支持:** HDP 2.4.x 的版本升级到 2.5.0, 包括 Apache Hadoop 从 2.7.1 到 2.7.3 升级。

**2. Ambari 2.2.0 插件支持:** Ambari 2.2.2 版本插件现在支持 HAWQ 2.0 的配置和部署。

**3. 自动的 Kerberos 配置:** HAWQ/PXF 插件现在支持自动的 Kerberos 安全配置。

## Docker 发展概况

### 官方版本发布情况

Docker 从 2013 年开源。2015 年是 Docker 开源项目突飞猛进的一年, 这段时间 Docker 官方先后发布了 V1.5、V1.6、V1.7、V1.8、V1.9 等 5 个大版本以及 7 个修订版本。

2016 年 Docker 发展同样迅速, 截止 2016 年 12 月 7 日 Docker 官方共发布了 V1.10、V1.11、V1.12、V1.13 等 4 个大版本以及 8 个修订版本。

重大变化主要是 V1.12 版本, 原 Swarm 项目改为 SwarmKit 并内置到了 Docker 引擎中。

详情: <https://github.com/docker/docker/releases>

### 国内情况

Docker 容器技术目前在国内还属于早期阶段。

- 第一: 技术太新, 技术人员需要一个接受的过程;
- 第二: 容器技术和虚拟化技术不一样, 和企业业务结合还需要进一步磨合;

阿里云和 Docker 官方合作, Docker 公司选择 aliyun 平台作为 Docker Hub 在中国运营的基础服务, 且阿里获得 Docker DataCenter 的运营权。。

由工信部、中国信息通信研究院联合成立的云计算开源产业联盟（OSCAR），在 12 月 2 日，发布了国内第一个“容器技术应用场景”规范，且该规范已立项为中国通信标准协会研究课题。

## 落地情况

国内的企业也有实际落地使用，互联网居多。代表性的列举几个：

互联网行业：京东、微博、知乎、豆瓣、PPTV；etc。

汽车行业：长安汽车、上海汽车；etc。

金融保险：永隆银行、中英人寿、广发证券；etc。

其他：...

## 国产数据库概览

### 达梦 DM

达梦数据库管理系统是达梦公司推出的具有完全自主知识产权的高性能数据库管理系统，简称 DM。目前产品的最新版本是达梦数据库管理系统 7.0 版本，简称 DM7。DM7 是达梦公司在总结 DM 系列产品研发与应用经验的基础之上，吸收主流数据库产品的优点，采用类 JAVA 的虚拟机技术设计的新一代数据库产品。DM7 基于成熟的关系数据模型和标准的接口，是一个跨越多种软硬件平台、具有大数据管理与分析能力、高效稳定的数据库管理系统。目前官网最新的可下载版本是 V7.1.5.117

### 达梦 V7.1.5.117 发布

近期主要改动包括：

#### 兼容性改进

1. 支持 DM7 到 DB2/SQLSERVER/SYBASE 的外部链接功能。
2. 支持 LISTAGG 等分析函数。
3. MERGE INTO 语法兼容 ORACLE，支持对派生表进行操作。

#### 工具/接口增强

1. 支持 PHP5.6 和 PDO5.6 接口。
2. OCI 接口新增一批函数/属性支持，如 OCINumberIsInt 等。
3. 开源 QT 环境的 QDM 驱动。
4. 支持对 PACKAGE 内部函数进行调试。
5. 支持 NODE.JS（独立安装）。

## 查询优化的改进

1. 支持对分区表进行分区裁剪优化功能。
2. 分区表支持间隔分区功能，优化间隔分区 DML 操作性能。
3. 优化器支持自适应优化功能，能够在执行期间发现计划达到预期限额时，自动切换为其他执行计划。
4. 新增索引监控功能，便于分析索引使用情况。

## 高级特性

### 1. DM DSC

DM DSC 是基于共享存储，允许多个 DM 数据库实例同时访问和操作同一数据库，具有高可用、高性能、负载均衡、故障透明切换等特性的高可用解决方案，类似 Oracle RAC。经过长达数年的测试和完善，近期 DM DSC 已经成功在几个试点项目上线，现正式对外推出。

### 2. DEM Web 管理工具

DEM 是 DM 基于 Web 的管理工具，可进行本地和远程联机管理；提供联机事件跟踪和会话、告警、性能监视，以协助数据库管理员进行系统管理和性能分析。

### 3. HUGE 分区表支持分区切换功能。

### 4. 支持 JSON 数据类型和相关操作。

### 5. 多级分区表支持增加分区和间隔分区功能。

### 6. MPP 环境复制表允许更新和删除操作。

## GBase

GBase 8a 是一款自主研发的列存储国产数据库。

## GBase 8a MPP Cluster 发布最新版本 V8511

### 以下是新功能一览

（更多内容可以参考官网 <http://www.gbase.cn>；）

- 集群架构与部署：完全并行的MPP + Shared Nothing 的分布式架构
- 海量数据分布压缩存储：采用hash分布、random 存储策略进行数据存储；
- 数据加载高效性：基于策略的数据加载模式,集群整体加载速度可达2TB/h；
- 高扩展、高可靠、高可用、高并发、易维护、低成本、行列混存、SQL标准化

## GBase 8t 发布最新版本 V8.7

以下是新功能一览

GBase8t 基于引进的 IBM 的成熟商用企业级数据库 Informix12.10 最新版授权源代码自主构造而发行。

更多内容可以参考官网及社区 <http://www.gbase.cn>;  
<http://www.informixchina.net/club/>;

- 高性能
  - 动态共享内存技术
  - 虚处理器技术
  - 磁盘技术
- 高可用性
  - HDR (High availability Data Replication, 高可用性数据复制)
  - RSS (Remote Standalone Secondary, 远程实时容灾)
  - SDS (Shared Disk Servers, 共享磁盘数据库集群)
  - ER (Enterprise Replication, 企业级复制)
- 高易用性
  - 监控运行状况
  - 管理高可用功能
  - 日志查看
  - 性能分析
  - 服务器管理

## 星瑞格数据库

Sinoregal Dynamic Server(Sinoregal DS) 是福建星瑞格软件公司基于 IBM Informix v12 授权源代码、并新增自主研发的图形化数据库监控优化、加密及审计功能，全新打造的国产数据库软件，适用于在线事务处理、集成的应用程序、大数据仓库分析工作负载、非结构化数据处理与分布式云计算架构。

### 12.10 FC7 版本发布

星瑞格数据库 12.10 FC7 新版本于 2016 年 12 月 1 日正式发布,新特性在数据存储、传感器数据、云功能、嵌入性和管理便捷性有了增强。

#### 数据仓库

新功能改善了服务器的实时性能和内存数据存储能力。升级后的管理界面让用户能够通过自动分区刷新和持续数据加载功能更轻易地对加速器进行日常配置管理。

## 传感器数据

存储和分析大量结构化数据的能力对于处理当前监测装置和智能仪表海量信息来说非常关键的。**Sinoregal DS** 独有的空间和时序数据类别能力让海量数据的存储变得更加简单,通过 **Sinoregal DS Warehouse Accelerator** (数据仓库加速器) 可以看清时序数据—将两种 **Sinoregal DS** 技术整合成一个数据搜索和分析引擎。时序数据管理能够定义存储容器的活跃和休眠时间段。虚拟表格界面和加载程序 API 也有多项强化功能, 确保能够比以前更加快速地存取时序数据。随着全新的 **ESRI 10.1** 数据库的使用, 空间数据支持也得到强化并具备全新的功能。

## 云功能

**Sinoregal DS Flexible Grid** 环境下的新功能, 包括通过推出区域的概念以及轮辐系统的简单配置, 让复杂复制环境内的共享数据配置和管理比起以前更加简单。强化了复制列的监测功能, 能够监测其异步属性, 并且在性能上也有所强化, 能够实现轻追加操作。连接管理器和故障内部服务器支持功能现在能够区分一个应用和数据库环境内的不同网络的优先级以确保维持高等级的服务水平, 在出现网络故障情况下确保共享盘辅助群集的运行。

## 嵌入性

除了自主功能, 其还能够自动压缩数据, 并且压缩功能现在支持更多的数据库目标, 从而降低了对存储空间的要求。对更多服务器参数的动态配置确保自动调整和自动修复功能, 包括通过输入输出界面实现参数的成组管理。分段表的滚动窗口定义强化了性能, 并可以自动控制实现简单管理。

## 便捷管理

开发出 **Open Admin Tool (OAT) for Sinoregal DS** 来对新功能进行监督和控制, 比如复制网格和 **Sinoregal DS Warehouse Accelerator** (数据仓库加速器) 数据集市。现在已经启用了—个 **DBAs** 的移动应用, 能够对所有数据库服务器系统的健康和性能进行监督。之前的 **Sinoregal DS Storage Manager** (存储管理器) 已经被全新的 **Primary Storage Manager** (总存储管理器) 替代, 确保服务器的备份流程更加简单快速。

## 应用开发支持

增加了多个全新的 **SQL** 语言成分简化开发应用, 特别是数据库的中性应用。除了上述 **OLAP** 功能之外, 还有新的 **UNION** 操作器, 用于同其它数据库服务器 (包括 **Sinoregal DS Parallel Server**) 和新的集合功能相兼容。**Sinoregal DS Genero®**的提升措施能够加快开发过程, 并为商业应用建模功能提供模板。**Sinoregal DS** 服务器也有许多内部功能强化用于改善交叉平台支持并实现性能提升。性能提升的案例包括下述情况: 通过使用哈希连接能够更好改善 **ANSI** 外连接请求, 并且服务器内部数据的压缩支持已经扩展到索引和简单大型标的, 从而降低输入/输出时间。

## 感谢名单

最后要感谢本期提供宝贵信息和建议的专家朋友，排名不分先后。

贡献者单位/职务	贡献者	贡献领域
宜信技术研发中心数据库架构师	韩锋	排行榜解读、Oracle 及通稿审核
MongoDB 中文社区发起人	唐建法	MongoDB
大街网数据库&缓存负责人	代晓磊	Redis
惠普金融 MySQL 专家	贺春昶	MariaDB 文章供稿
PingCAP CTO 兼联合创始人	黄东旭	TiDB、通稿审核
京东资深架构师	张成远	数据库发展历程供稿
阿里 O2O 仓储解决方案公司 DBA	林勇桦	SQL Server
Bloom Software CTO	杨旭钧	Geode
PGer	德哥@Digoal	PostgreSQL
中国移动苏州研发中心	姚昕 @bigclouder	HBase、Hadoop
杭州有赞科技 MySQL DBA	杨奇龙	MySQL 5.7 特性解读
DBAplus 社群	杨志洪	策划、通稿审核
IBM 资深数据库架构师	肖静静	DB2
DBAplus 社群	杨建荣	其他数据库、策划、通稿审核
星瑞格售前技术总监	杨建鑫	星瑞格
希云 cSphere-技术总监	张春源	Docker
南大通用华东区销售总监	李宁	GBase
武汉达梦数据库有限公司		达梦
优酷土豆技术专家	张海雷	Druid
DBAplus 社群	林林	策划、发布
熊猫直播高级 DBA	杨尚刚	通稿审核
新炬网络开源技术专家	高强	通稿审核
搜狐畅游数据仓库负责人	田红瀚	Kudu
哈尔滨银行数据库架构师	洪烨	DB2
Pivotal 高级产品经理	Brian Lu	Greenplum、HAWQ

欢迎提供 Newsletter 信息，发送至邮箱：[newsletter@dbaplus.cn](mailto:newsletter@dbaplus.cn)

欢迎技术文章投稿，发送至邮箱：[editor@dbaplus.cn](mailto:editor@dbaplus.cn)