

# 1、独立组件个数(按进程)

---

默认情况下是1个；如果需要使用副本机制，需要依赖zookeeper；如果需要监控功能，还得依赖第三方监控系统。

# 2、单机部署

---

很好的支持单机运行，并且单机情况下查询入库性能不错（通过其提供的示例数据进行体验）。

# 3、窗口函数

---

Clickhouse没有显示的支持窗口函数，根据网上的资料，可以通过 `arrayEnumerate`，`arrayEnumerateDense`，`arrayEnumerateUniq` 函数间接的实现简单的窗口函数功能。但是用这种方式写查询语句会比较繁琐。参考：

<https://blog.csdn.net/vkingnew/article/details/106781788>

# 4、数据自动平衡

---

（1）分布式表入库时，分布式表会根据sharding\_key把数据划分到不同的shard中，这个算是写入时的数据平衡机制；

（2）如果增加新shard，已经入库的数据不会自动均衡到新shard中，必须通过人工命令对数据进行移动。

# 5、离线处理

---

MergeTree系列的表引擎中包含几个有特殊功能的引擎：

- **ReplacingMergeTree**

引擎内部在merge时会对具有相同Sorting Key的行进行去重，至于多个重复的行保留哪个是由ReplacingMergeTree的参数决定的，参数指定一个表示版本的列，多个重复的行保留此列中最大值的那个；如果没有指定，则保存最后遇到的那行。

- **SummingMergeTree**

相当于对MergeTree中的Sorting Key进行分组，对数值型字段进行sum聚合。其包含一个可选的参数用于指定需要对哪些字段进行sum聚合，如果没有指定，则聚合除sorting key之外的所有数值类型字段。

需要注意的是，这种表中的数据只有再merge时才会执行聚合，所以聚合的结果不一定是最终值，所以查询时还是得用group by语句。

聚合规则说明：

(1) 如果参数执行columns，则这些column被sum；否则，除sorting key之外的数值型字段被sum；

(2) 如果sum之后，所有字段都为0，则这行会被删除；

(3) 如果某列既没有在sorting key中，也不能被sum，则从该列的多个值中随机选择一个。

- **AggregatingMergeTree**

相当于对Sorting Key进行分组，然后对 AggregateFunction 和 SimpleAggregateFunction 类型的字段进行聚合。同样，这个步骤也是在Merge时进行的。

因为这种引擎中包含 AggregateFunction 类型的字段，所以写入数据及查询时，需要注意 AggregateFunction 的用法。

- **CollapsingMergeTree**

简单点说，这个引擎通过一个 sign 字段来实现追加方式的数据更新，这个字段只有两个值：1 和-1。比如要记录某个对象的状态，如果第一次写入，则写入状态数据并把 sign 字段赋值为1；如果需要更新状态，则需要添加两行，第一行的内容与上一个状态信息相同，但sign为-1，表示对上一个状态的删除，第二行为新状态数据，sign为1，表示新状态的写入。合并时此引擎会自动删除无用的行（1和-1配对的行）。

这种方式导致每次写入数据前需要记住前一个数据的信息，而且查询结果的准确性严重依赖数据写入的一致性，只有正确的写入-1和1标志的数据，才能保证最终结果的正确性。

需要注意的是，Merge是行的合并只考虑sorting key和sign字段，所以写入标志为-1的数据时使用之前的状态只是为了计算的方便。Merge后，相同sorting key的行最多存在两个，一个sign为1，一个sign为-1。

合并规则为：

- (1) 如果1和-1的行数相同，并且最后一行为1，则保存第一个-1行和最后一个1行；
- (2) 如果1行多于-1行，则保留最后一个1行；
- (3) 如果-1行多于1行，则保留第一个-1行；
- (4) 其他情况下，不保留任何行。

- **VersionedCollapsingMergeTree**

相当于CollapsingMergeTree的升级版，通过增加一个version字段，解除了CollapsingMergeTree中数据写入时对顺序的严格要求。

对于合并规则，sorting key和version字段相同但sign字段不同的行会被删除。

## 6、冷热数据分离

---

冷热数据分离仅限于单机上分离，即不同类型的数据可以放置到不同的磁盘上，ClickHouse提供了存储策略支持数据在磁盘间自动转移。

具体参考《随机.md》中的磁盘配置及存储策略部分。

## 7、查询资源隔离

---

没有相关特性

## 8、存储效率

表名	原始大小	存储大小	空间比例
hits_v1	7784351125	1270083754	16.32%
visits_v1	2657415178	560532035	21.09%

表结构:

hits\_v1

```
ATTACH TABLE hits_v1
(
  `WatchID` UInt64,
  `JavaEnable` UInt8,
  `Title` String,
  `GoodEvent` Int16,
  `EventTime` DateTime,
  `EventDate` Date,
  `CounterID` UInt32,
  `ClientIP` UInt32,
  `ClientIP6` FixedString(16),
  `RegionID` UInt32,
  `UserID` UInt64,
  `CounterClass` Int8,
  `OS` UInt8,
  `UserAgent` UInt8,
  `URL` String,
  `Referer` String,
  `URLDomain` String,
  `RefererDomain` String,
  `Refresh` UInt8,
  `IsRobot` UInt8,
  `RefererCategories` Array(UInt16),
  `URLCategories` Array(UInt16),
  `URLRegions` Array(UInt32),
  `RefererRegions` Array(UInt32),
  `ResolutionWidth` UInt16,
  `ResolutionHeight` UInt16,
  `ResolutionDepth` UInt8,
  `FlashMajor` UInt8,
  `FlashMinor` UInt8,
  `FlashMinor2` String,
  `NetMajor` UInt8,
  `NetMinor` UInt8,
  `UserAgentMajor` UInt16,
  `UserAgentMinor` FixedString(2),
  `CookieEnable` UInt8,
  `JavascriptEnable` UInt8,
  `IsMobile` UInt8,
  `MobilePhone` UInt8,
  `MobilePhoneModel` String,
  `Params` String,
  `IPNetworkID` UInt32,
  `TrafficSourceID` Int8,
```

```
`SearchEngineID` UInt16,  
`SearchPhrase` String,  
`AdvEngineID` UInt8,  
`IsArtificial` UInt8,  
`WindowClientWidth` UInt16,  
`WindowClientHeight` UInt16,  
`ClientTimeZone` Int16,  
`ClientEventTime` DateTime,  
`SilverlightVersion1` UInt8,  
`SilverlightVersion2` UInt8,  
`SilverlightVersion3` UInt32,  
`SilverlightVersion4` UInt16,  
`PageCharset` String,  
`CodeVersion` UInt32,  
`IsLink` UInt8,  
`IsDownload` UInt8,  
`IsNotBounce` UInt8,  
`FUniqID` UInt64,  
`HID` UInt32,  
`IsOldCounter` UInt8,  
`IsEvent` UInt8,  
`IsParameter` UInt8,  
`DontCountHits` UInt8,  
`WithHash` UInt8,  
`HitColor` FixedString(1),  
`UTCEventTime` DateTime,  
`Age` UInt8,  
`Sex` UInt8,  
`Income` UInt8,  
`Interests` UInt16,  
`Robotness` UInt8,  
`GeneralInterests` Array(UInt16),  
`RemoteIP` UInt32,  
`RemoteIP6` FixedString(16),  
`WindowName` Int32,  
`OpenerName` Int32,  
`HistoryLength` Int16,  
`BrowserLanguage` FixedString(2),  
`BrowserCountry` FixedString(2),  
`SocialNetwork` String,  
`SocialAction` String,  
`HTTPError` UInt16,  
`SendTiming` Int32,  
`DNSTiming` Int32,  
`ConnectTiming` Int32,  
`ResponseStartTiming` Int32,  
`ResponseEndTiming` Int32,  
`FetchTiming` Int32,  
`RedirectTiming` Int32,  
`DOMInteractiveTiming` Int32,  
`DOMContentLoadedTiming` Int32,  
`DOMCompleteTiming` Int32,  
`LoadEventStartTiming` Int32,  
`LoadEventEndTiming` Int32,  
`NSToDOMContentLoadedTiming` Int32,  
`FirstPaintTiming` Int32,  
`RedirectCount` Int8,  
`SocialSourceNetworkID` UInt8,
```

```

        `SocialSourcePage` String,
        `ParamPrice` UInt64,
        `ParamOrderID` String,
        `ParamCurrency` FixedString(3),
        `ParamCurrencyID` UInt16,
        `GoalsReached` Array(UInt32),
        `OpenstatServiceName` String,
        `OpenstatCampaignID` String,
        `OpenstatAdID` String,
        `OpenstatSourceID` String,
        `UTMSource` String,
        `UTMMedium` String,
        `UTMCampaign` String,
        `UTMContent` String,
        `UTMTerm` String,
        `FromTag` String,
        `HasGCLID` UInt8,
        `RefererHash` UInt64,
        `URLHash` UInt64,
        `CLID` UInt32,
        `YCLID` UInt64,
        `ShareService` String,
        `ShareURL` String,
        `ShareTitle` String,
        `ParsedParams.Key1` Array(String),
        `ParsedParams.Key2` Array(String),
        `ParsedParams.Key3` Array(String),
        `ParsedParams.Key4` Array(String),
        `ParsedParams.Key5` Array(String),
        `ParsedParams.ValueDouble` Array(Float64),
        `IslandID` FixedString(16),
        `RequestNum` UInt32,
        `RequestTry` UInt8
    )
ENGINE = MergeTree()
PARTITION BY toYYYYMM(EventDate)
ORDER BY (CounterID, EventDate, intHash32(UserID))
SAMPLE BY intHash32(UserID)
SETTINGS index_granularity = 8192

```

visits\_v1

```

ATTACH TABLE visits_v1
(
    `CounterID` UInt32,
    `StartDate` Date,
    `Sign` Int8,
    `IsNew` UInt8,
    `VisitID` UInt64,
    `UserID` UInt64,
    `StartTime` DateTime,
    `Duration` UInt32,
    `UTCStartTime` DateTime,
    `PageViews` Int32,
    `Hits` Int32,

```

```
`IsBounce` UInt8,
`Referer` String,
`StartURL` String,
`RefererDomain` String,
`StartURLDomain` String,
`EndURL` String,
`LinkURL` String,
`IsDownload` UInt8,
`TrafficSourceID` Int8,
`SearchEngineID` UInt16,
`SearchPhrase` String,
`AdvEngineID` UInt8,
`PlaceID` Int32,
`RefererCategories` Array(UInt16),
`URLCategories` Array(UInt16),
`URLRegions` Array(UInt32),
`RefererRegions` Array(UInt32),
`IsYandex` UInt8,
`GoalReachesDepth` Int32,
`GoalReachesURL` Int32,
`GoalReachesAny` Int32,
`SocialSourceNetworkID` UInt8,
`SocialSourcePage` String,
`MobilePhoneModel` String,
`ClientEventTime` DateTime,
`RegionID` UInt32,
`ClientIP` UInt32,
`ClientIP6` FixedString(16),
`RemoteIP` UInt32,
`RemoteIP6` FixedString(16),
`IPNetworkID` UInt32,
`SilverlightVersion3` UInt32,
`CodeVersion` UInt32,
`ResolutionWidth` UInt16,
`ResolutionHeight` UInt16,
`UserAgentMajor` UInt16,
`UserAgentMinor` UInt16,
`WindowClientWidth` UInt16,
`WindowClientHeight` UInt16,
`SilverlightVersion2` UInt8,
`SilverlightVersion4` UInt16,
`FlashVersion3` UInt16,
`FlashVersion4` UInt16,
`ClientTimeZone` Int16,
`OS` UInt8,
`UserAgent` UInt8,
`ResolutionDepth` UInt8,
`FlashMajor` UInt8,
`FlashMinor` UInt8,
`NetMajor` UInt8,
`NetMinor` UInt8,
`MobilePhone` UInt8,
`SilverlightVersion1` UInt8,
`Age` UInt8,
`Sex` UInt8,
`Income` UInt8,
`JavaEnable` UInt8,
`CookieEnable` UInt8,
```

```
`JavascriptEnable` UInt8,
`IsMobile` UInt8,
`BrowserLanguage` UInt16,
`BrowserCountry` UInt16,
`Interests` UInt16,
`Robotness` UInt8,
`GeneralInterests` Array(UInt16),
`Params` Array(String),
`Goals.ID` Array(UInt32),
`Goals.Serial` Array(UInt32),
`Goals.EventTime` Array(DateTime),
`Goals.Price` Array(Int64),
`Goals.OrderID` Array(String),
`Goals.CurrencyID` Array(UInt32),
`WatchIDs` Array(UInt64),
`ParamSumPrice` Int64,
`ParamCurrency` FixedString(3),
`ParamCurrencyID` UInt16,
`ClickLogID` UInt64,
`ClickEventID` Int32,
`ClickGoodEvent` Int32,
`ClickEventTime` DateTime,
`ClickPriorityID` Int32,
`ClickPhraseID` Int32,
`ClickPageID` Int32,
`ClickPlaceID` Int32,
`ClickTypeID` Int32,
`ClickResourceID` Int32,
`ClickCost` UInt32,
`ClickClientIP` UInt32,
`ClickDomainID` UInt32,
`ClickURL` String,
`ClickAttempt` UInt8,
`ClickOrderID` UInt32,
`ClickBannerID` UInt32,
`ClickMarketCategoryID` UInt32,
`ClickMarketPP` UInt32,
`ClickMarketCategoryName` String,
`ClickMarketPPName` String,
`ClickAWAPSCampaignName` String,
`ClickPageName` String,
`ClickTargetType` UInt16,
`ClickTargetPhraseID` UInt64,
`ClickContextType` UInt8,
`ClickSelectType` Int8,
`ClickOptions` String,
`ClickGroupBannerID` Int32,
`OpenstatServiceName` String,
`OpenstatCampaignID` String,
`OpenstatAdID` String,
`OpenstatSourceID` String,
`UTMSource` String,
`UTMMedium` String,
`UTMCampaign` String,
`UTMContent` String,
`UTMTerm` String,
`FromTag` String,
`HasGCLID` UInt8,
```

```

`FirstVisit` DateTime,
`PredLastVisit` Date,
`LastVisit` Date,
`TotalVisits` UInt32,
`TrafficSource.ID` Array(Int8),
`TrafficSource.SearchEngineID` Array(UInt16),
`TrafficSource.AdvEngineID` Array(UInt8),
`TrafficSource.PlaceID` Array(UInt16),
`TrafficSource.SocialSourceNetworkID` Array(UInt8),
`TrafficSource.Domain` Array(String),
`TrafficSource.SearchPhrase` Array(String),
`TrafficSource.SocialSourcePage` Array(String),
`Attendance` FixedString(16),
`CLID` UInt32,
`YCLID` UInt64,
`NormalizedRefererHash` UInt64,
`SearchPhraseHash` UInt64,
`RefererDomainHash` UInt64,
`NormalizedStartURLHash` UInt64,
`StartURLDomainHash` UInt64,
`NormalizedEndURLHash` UInt64,
`TopLevelDomain` UInt64,
`URLScheme` UInt64,
`OpenstatServiceNameHash` UInt64,
`OpenstatCampaignIDHash` UInt64,
`OpenstatAdIDHash` UInt64,
`OpenstatSourceIDHash` UInt64,
`UTMSourceHash` UInt64,
`UTMMediumHash` UInt64,
`UTMCampaignHash` UInt64,
`UTMContentHash` UInt64,
`UTMTermHash` UInt64,
`FromHash` UInt64,
`WebVisorEnabled` UInt8,
`WebVisorActivity` UInt32,
`ParsedParams.Key1` Array(String),
`ParsedParams.Key2` Array(String),
`ParsedParams.Key3` Array(String),
`ParsedParams.Key4` Array(String),
`ParsedParams.Key5` Array(String),
`ParsedParams.ValueDouble` Array(Float64),
`Market.Type` Array(UInt8),
`Market.GoalID` Array(UInt32),
`Market.OrderID` Array(String),
`Market.OrderPrice` Array(Int64),
`Market.PP` Array(UInt32),
`Market.DirectPlaceID` Array(UInt32),
`Market.DirectOrderID` Array(UInt32),
`Market.DirectBannerID` Array(UInt32),
`Market.GoodID` Array(String),
`Market.GoodName` Array(String),
`Market.GoodQuantity` Array(Int32),
`Market.GoodPrice` Array(Int64),
`IslandID` FixedString(16)
)
ENGINE = CollapsingMergeTree(Sign)
PARTITION BY toYYYYMM(StartDate)
ORDER BY (CounterID, StartDate, intHash32(UserID), VisitID)

```



```
SAMPLE BY intHash32(userID)
SETTINGS index_granularity = 8192
```

## 9、副本支持

参考《infos.md》中的ClickHouse的副本管理

## 10、排序字段类型支持

创建表时可以指定order by子句，表中存储的数据均以此子句中指定的列进行排序，可以指定多列，依次排序。不限字段个数及类型。

## 11、数据过期策略

MergeTree有个TTL属性：

```
TTL expr [DELETE|TO DISK 'aaa'|TO VOLUME 'bbb'], ...
```

其中expr用于判断表中的某行是否达到某个条件，其后可跟随一条规则，有三种规则类型可指定：

- DELETE  
用于删除满足expr的行。
- TO DISK 'aaa'  
当某个part中所有的行都满足expr时，把这个part移动到指定磁盘。
- TO VOLUME 'bbb'  
当某个part中所有的行都满足expr时，把这个part移动到指定的volume中。

如果没有指定规则，则默认为DELETE。

TTL在表创建后可以通过ALTER TABLE进行修改。

满足expr的数据只会在merge的时候删除，所以ClickHouse内部会启动一些针对TTL的合并，这些合并的执行频率由配置项 `merge_with_ttl_timeout` 控制。当数据已经过期，但还没有执行合并时，查询会连带过期数据一起处理，所以ttl合并的时间间隔太大会导致数据删除不及时，太小导致合并资源占用过大。另外，如果不想查询到过期数据，则可以在查询之前执行OPTIMIZE语句。

## 12、日志分析

日志分析大多数为结构化数据的分析，每个日志都会被转化为固定的字段进行存储及查询，但日志分析在进行数据过滤可能会设计掉全文本搜索功能，要完成全文本搜索，需要对字段内容建立倒排索引，然后提供查询倒排索引的机制。

ClickHouse可以有效的处理日志分析中的结构化数据存储及查询，但其不支持全文本的搜索功能。所以对日志分析场景的适用性需要判断其是否存在全文检索的需求。