



链滴

# Apache ORC 格式简介和使用工具读写

作者: [flowaters](#)

原文链接: <https://ld246.com/article/1529595963054>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 背景

在[列存储格式学习](#)一文中，我们了解了各种列存储格式的基本知识。

其中存储空间成本最优的格式为ORC格式。

本文针对性的来看一下ORC格式。

## 特点

ORC格式在2013年1月提出，目的是提高Hadoop Hive的计算性能和存储效率。

另外，做为一种存储格式，其是一种自描述的列存储格式，为大规模流式读取而专门设计优化。使用描述的列存储格式的优点是，可以根据不同的列类型选择不同的压缩算法。

主要特点为：

- [支持ACID](#)
- [内置索引](#): 包括每列的最小值、最大值和bloom过滤器，可以直接定位到对应的列。使用[Search Arguments](#)语法来使用索引。
- [支持复杂类型](#): 支持Hive的[所有类型](#)，尤其是复杂类型包括struct, list, map, union.

## 典型使用案例

官网上的用户案例时间比较久了，不过还是有可参考性。

- [2014年4月11日 Facebook使用ORC来存储了300PB的数据](#)
- [2014年 Yahoo使用ORC来存储数据](#)

## 使用示例

官方提供了一些[快速使用示例](#)

- [Hive](#)
- [Hadoop MapRed](#)
- [Hadoop MapReduce](#)
- [Core Java](#)
- [Core C++](#)

官方也同时提供了一些[工具](#)

- [Java工具](#)
- [C++工具](#)

本文只关注最后的[Java示例](#)

## mac下安装

这里以C++版本使用为例.

## 下载

wget <http://mirror.bit.edu.cn/apache/orc/orc-1.5.1/orc-1.5.1.tar.gz>

## 编译

```
% mkdir build
% cd build
% cmake ..
% make package test-out
```

结果

```
CPack: Create package
CPack: - package: /Users/note/orc/orc-1.5.1/build/ORC-1.5.1-Darwin.tar.gz generated.
Test project /Users/note/orc/orc-1.5.1/build
  Start 1: orc-test
1/3 Test #1: orc-test ..... Passed   1.62 sec
```

## orc-contents使用

以JSON格式显示ORC文件内容

语法

```
% orc-contents [--columns=1,2,...] <filename>
```

示例

```
$ orc-contents examples/TestOrcFile.test1.orc
{"boolean1": false, "byte1": 1, "short1": 1024, "int1": 65536, "long1": 9223372036854775807, "float1": 1, "double1": -15, "bytes1": [0, 1, 2, 3, 4], "string1": "hi", "middle": {"list": [{"int1": 1, "string1": "bye"}, {"int1": 2, "string1": "sigh"}]}, "list": [{"int1": 3, "string1": "good"}, {"int1": 4, "string1": "bad"}], "map": []}
{"boolean1": true, "byte1": 100, "short1": 2048, "int1": 65536, "long1": 9223372036854775807, "float1": 2, "double1": -5, "bytes1": [], "string1": "bye", "middle": {"list": [{"int1": 1, "string1": "bye"}, {"int1": 2, "string1": "sigh"}]}, "list": [{"int1": 100000000, "string1": "cat"}, {"int1": -100000, "string1": "in"}, {"int1": 1234, "string1": "hat"}], "map": [{"key": "chani", "value": {"int1": 5, "string1": "chani"}}, {"key": "mauddib", "value": {"int1": 1, "string1": "mauddib"}}]}
```

## orc-metadata使用

以JSON格式显示ORC文件的元数据信息。

**verbose**显示ORC文件布局信息

**raw**直接显示protocol buffers内容，而不是解释其格式

语法

```
% orc-metadata [-v] [--raw] <filename>
```

## 示例

```
$ orc-metadata examples/TestOrcFile.test1.orc
{"name": "examples/TestOrcFile.test1.orc",
 "type": "struct<boolean1:boolean,byte1:tinyint,short1:smallint,int1:int,long1:bigint,float1:float
double1:double,bytes1:binary,string1:string,middle:struct<list:array<struct<int1:int,string1:stri
g>>>,list:array<struct<int1:int,string1:string>>,map:map<string,struct<int1:int,string1:string
>>>",
 "rows": 2,
 "stripe count": 1,
 "format": "0.12", "writer version": "HIVE-8732",
 "compression": "zlib", "compression block": 10000,
 "file length": 1711,
 "content": 1015, "stripe stats": 250, "footer": 421, "postscript": 24,
 "row index stride": 10000,
 "user metadata": {
},
 "stripes": [
  { "stripe": 0, "rows": 2,
    "offset": 3, "length": 1012,
    "index": 570, "data": 243, "footer": 199
  }
]
}
```

## orc-statistics工具

显示文件级别和strip级别的列统计信息

**withIndex**可以设置来包含每个组的列统计信息。

### 语法

```
% orc-statistics [--withIndex] <filename>
```

### 示例

```
$ orc-statistics examples/TestOrcFile.test1.orc
File examples/TestOrcFile.test1.orc has 24 columns
*** Column 0 ***
Column has 2 values and has null value: no

*** Column 1 ***
Data type: Boolean
Values: 2
Has null: no
(true: 1; false: 1)

*** Column 2 ***
Data type: Integer
Values: 2
Has null: no
Minimum: 1
Maximum: 100
```

Sum: 101

\*\*\* Column 3 \*\*\*  
Data type: Integer  
Values: 2  
Has null: no  
Minimum: 1024  
Maximum: 2048  
Sum: 3072

\*\*\* Column 4 \*\*\*  
Data type: Integer  
Values: 2  
Has null: no  
Minimum: 65536  
Maximum: 65536  
Sum: 131072

\*\*\* Column 5 \*\*\*  
Data type: Integer  
Values: 2  
Has null: no  
Minimum: 9223372036854775807  
Maximum: 9223372036854775807  
Sum: not defined

\*\*\* Column 6 \*\*\*  
Data type: Double  
Values: 2  
Has null: no  
Minimum: 1  
Maximum: 2  
Sum: 3

\*\*\* Column 7 \*\*\*  
Data type: Double  
Values: 2  
Has null: no  
Minimum: -15  
Maximum: -5  
Sum: -20

\*\*\* Column 8 \*\*\*  
Data type: Binary  
Values: 2  
Has null: no  
Total length: 5

\*\*\* Column 9 \*\*\*  
Data type: String  
Values: 2  
Has null: no  
Minimum: bye  
Maximum: hi

Total length: 5

\*\*\* Column 10 \*\*\*

Column has 2 values and has null value: no

\*\*\* Column 11 \*\*\*

Column has 2 values and has null value: no

\*\*\* Column 12 \*\*\*

Column has 4 values and has null value: no

\*\*\* Column 13 \*\*\*

Data type: Integer

Values: 4

Has null: no

Minimum: 1

Maximum: 2

Sum: 6

\*\*\* Column 14 \*\*\*

Data type: String

Values: 4

Has null: no

Minimum: bye

Maximum: sigh

Total length: 14

\*\*\* Column 15 \*\*\*

Column has 2 values and has null value: no

\*\*\* Column 16 \*\*\*

Column has 5 values and has null value: no

\*\*\* Column 17 \*\*\*

Data type: Integer

Values: 5

Has null: no

Minimum: -100000

Maximum: 100000000

Sum: 99901241

\*\*\* Column 18 \*\*\*

Data type: String

Values: 5

Has null: no

Minimum: bad

Maximum: in

Total length: 15

\*\*\* Column 19 \*\*\*

Column has 2 values and has null value: no

\*\*\* Column 20 \*\*\*

Data type: String

Values: 2  
Has null: no  
Minimum: chani  
Maximum: mauddib  
Total length: 12

\*\*\* Column 21 \*\*\*  
Column has 2 values and has null value: no

\*\*\* Column 22 \*\*\*  
Data type: Integer  
Values: 2  
Has null: no  
Minimum: 1  
Maximum: 5  
Sum: 6

\*\*\* Column 23 \*\*\*  
Data type: String  
Values: 2  
Has null: no  
Minimum: chani  
Maximum: mauddib  
Total length: 12

File examples/TestOrcFile.test1.orc has 1 stripes  
\*\*\* Stripe 0 \*\*\*

--- Column 0 ---  
Column has 2 values and has null value: no

--- Column 1 ---  
Data type: Boolean  
Values: 2  
Has null: no  
(true: 1; false: 1)

--- Column 2 ---  
Data type: Integer  
Values: 2  
Has null: no  
Minimum: 1  
Maximum: 100  
Sum: 101

--- Column 3 ---  
Data type: Integer  
Values: 2  
Has null: no  
Minimum: 1024  
Maximum: 2048  
Sum: 3072

--- Column 4 ---

Data type: Integer  
Values: 2  
Has null: no  
Minimum: 65536  
Maximum: 65536  
Sum: 131072

--- Column 5 ---  
Data type: Integer  
Values: 2  
Has null: no  
Minimum: 9223372036854775807  
Maximum: 9223372036854775807  
Sum: not defined

--- Column 6 ---  
Data type: Double  
Values: 2  
Has null: no  
Minimum: 1  
Maximum: 2  
Sum: 3

--- Column 7 ---  
Data type: Double  
Values: 2  
Has null: no  
Minimum: -15  
Maximum: -5  
Sum: -20

--- Column 8 ---  
Data type: Binary  
Values: 2  
Has null: no  
Total length: 5

--- Column 9 ---  
Data type: String  
Values: 2  
Has null: no  
Minimum: bye  
Maximum: hi  
Total length: 5

--- Column 10 ---  
Column has 2 values and has null value: no

--- Column 11 ---  
Column has 2 values and has null value: no

--- Column 12 ---  
Column has 4 values and has null value: no



--- Column 13 ---  
Data type: Integer  
Values: 4  
Has null: no  
Minimum: 1  
Maximum: 2  
Sum: 6

--- Column 14 ---  
Data type: String  
Values: 4  
Has null: no  
Minimum: bye  
Maximum: sigh  
Total length: 14

--- Column 15 ---  
Column has 2 values and has null value: no

--- Column 16 ---  
Column has 5 values and has null value: no

--- Column 17 ---  
Data type: Integer  
Values: 5  
Has null: no  
Minimum: -100000  
Maximum: 100000000  
Sum: 99901241

--- Column 18 ---  
Data type: String  
Values: 5  
Has null: no  
Minimum: bad  
Maximum: in  
Total length: 15

--- Column 19 ---  
Column has 2 values and has null value: no

--- Column 20 ---  
Data type: String  
Values: 2  
Has null: no  
Minimum: chani  
Maximum: mauddib  
Total length: 12

--- Column 21 ---  
Column has 2 values and has null value: no

--- Column 22 ---  
Data type: Integer

Values: 2  
Has null: no  
Minimum: 1  
Maximum: 5  
Sum: 6

--- Column 23 ---  
Data type: String  
Values: 2  
Has null: no  
Minimum: chani  
Maximum: mauddib  
Total length: 12

## csv-import工具

将CSV文件内容导入ORC文件。目前此工具不支持复合类型。

**delimiter**来标识分隔符，默认为  
**stripe**为分片大小，默认为128MB  
**block**为压缩块大小，默认为64KB  
**batch**默认为1024行

语法

```
% csv-import [--delimiter=<character>] [--stripe=<size>]  
              [--block=<size>] [--batch=<size>]  
              <schema> <inputCSVFile> <outputORCFile>
```

示例

```
$ csv-import "struct<a:bigint,b:string,c:double>" examples/TestCSVFileImport.test10rows.csv  
tmp/test.orc  
[2018-06-21 23:42:26] Start importing Orc file...  
[2018-06-21 23:42:26] Finish importing Orc file.  
[2018-06-21 23:42:26] Total writer elapsed time: 0.001386s.  
[2018-06-21 23:42:26] Total writer CPU time: 0.001374s.
```

## orc-scan工具

扫描和显示ORC文件行数，用来检查文件是否损坏。batch默认值为1024。

语法

```
% orc-scan [--batch=<size>] <filename>
```

示例

```
$ orc-scan examples/TestOrcFile.test1.orc  
Rows: 2  
Batches: 1
```

## 参考

- [列存储格式学习](#)
- [Apache ORC: the smallest, fastest columnar storage for Hadoop workloads.](#)
- [Hive and Apache Tez: Benchmarked at Yahoo! Scale](#) : 其中有ORC File Layout和ORC参数配置幻灯片
- [Scaling the Facebook data warehouse to 300 PB: Facebook ORCFile的性能比Open source O CFile还要好](#)