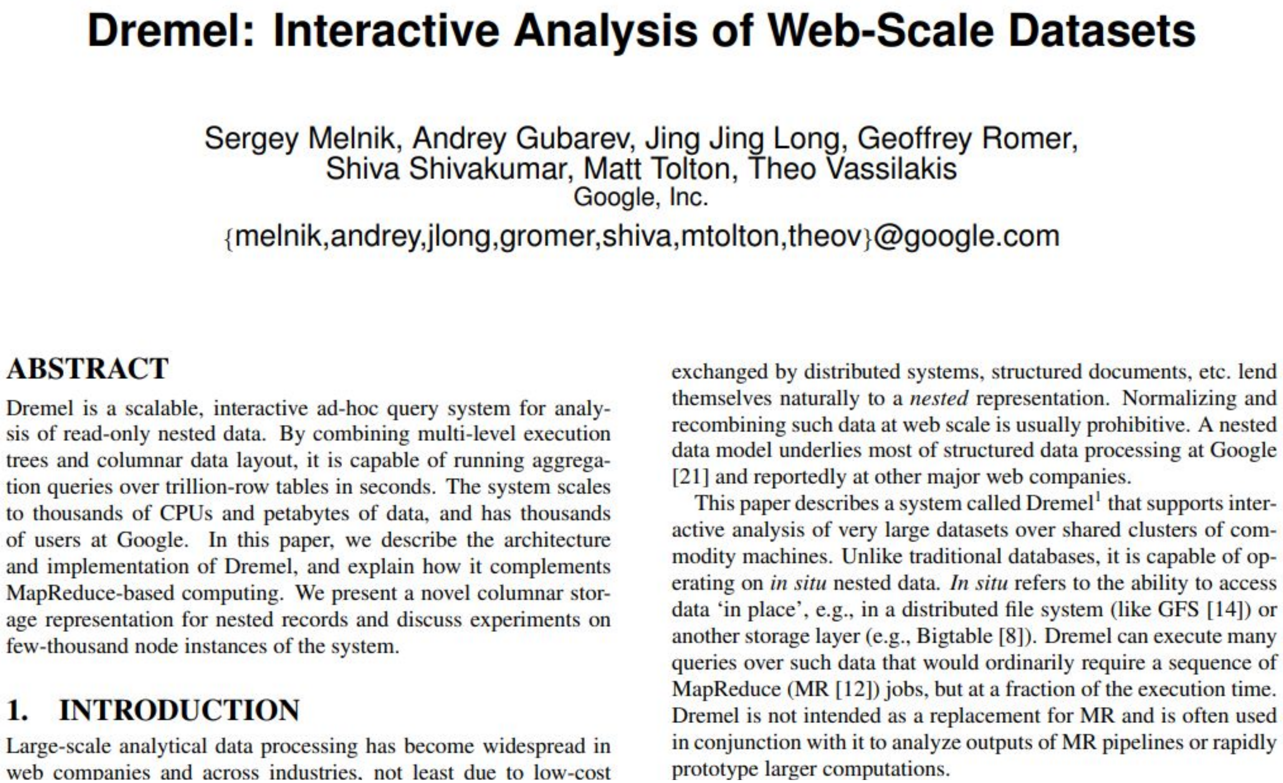
Формат данных Parquet и сравнение с ORC

Драль Алексей, study@bigdatateam.org
CEO at BigData Team, https://bigdatateam.org
https://www.facebook.com/bigdatateam

Google Dremel



Dremel: Interactive Analysis of Web-Scale Datasets

Sergey Melnik, Andrey Gubarev, Jing Jing Long, Geoffrey Romer,
Shiva Shivakumar, Matt Tolton, Theo Vassilakis
Google, Inc.
{melnik,andrey,jlong,gromer,shiva,mtolton,theov}@google.com

ABSTRACT

Dremel is a scalable, interactive ad-hoc query system for analysis of read-only nested data. By combining multi-level execution trees and columnar data layout, it is capable of running aggregation queries over trillion-row tables in seconds. The system scales to thousands of CPUs and petabytes of data, and has thousands of users at Google. In this paper, we describe the architecture and implementation of Dremel, and explain how it complements MapReduce-based computing. We present a novel columnar storage representation for nested records and discuss experiments on few-thousand node instances of the system.

1. INTRODUCTION

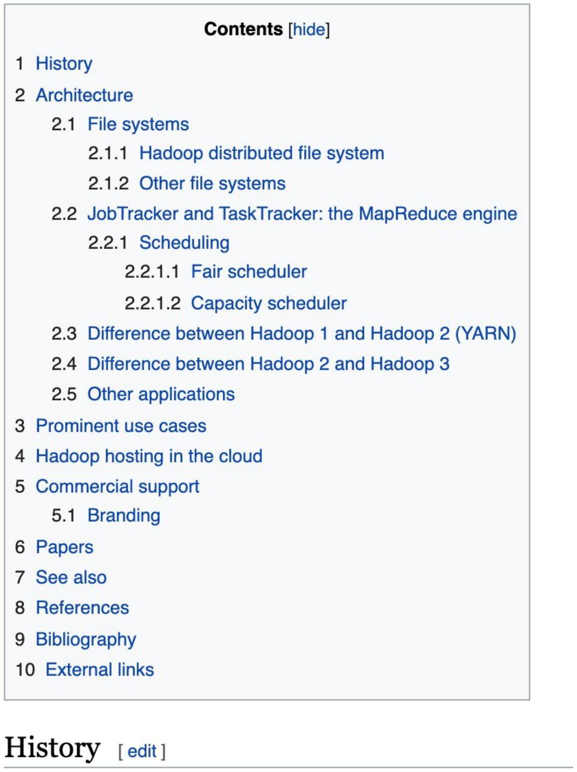
Large-scale analytical data processing has become widespread in web companies and across industries, not least due to low-cost

exchanged by distributed systems, structured documents, etc. lend themselves naturally to a *nested* representation. Normalizing and recombining such data at web scale is usually prohibitive. A nested data model underlies most of structured data processing at Google [21] and reportedly at other major web companies.

This paper describes a system called Dremel[1] that supports interactive analysis of very large datasets over shared clusters of commodity machines. Unlike traditional databases, it is capable of operating on *in situ* nested data. *In situ* refers to the ability to access data 'in place', e.g., in a distributed file system (like GFS [14]) or another storage layer (e.g., Bigtable [8]). Dremel can execute many queries over such data that would ordinarily require a sequence of MapReduce (MR [12]) jobs, but at a fraction of the execution time. Dremel is not intended as a replacement for MR and is often used in conjunction with it to analyze outputs of MR pipelines or rapidly prototype larger computations.

Dremel: Interactive Analysis of Web-Scale Datasets (2010)

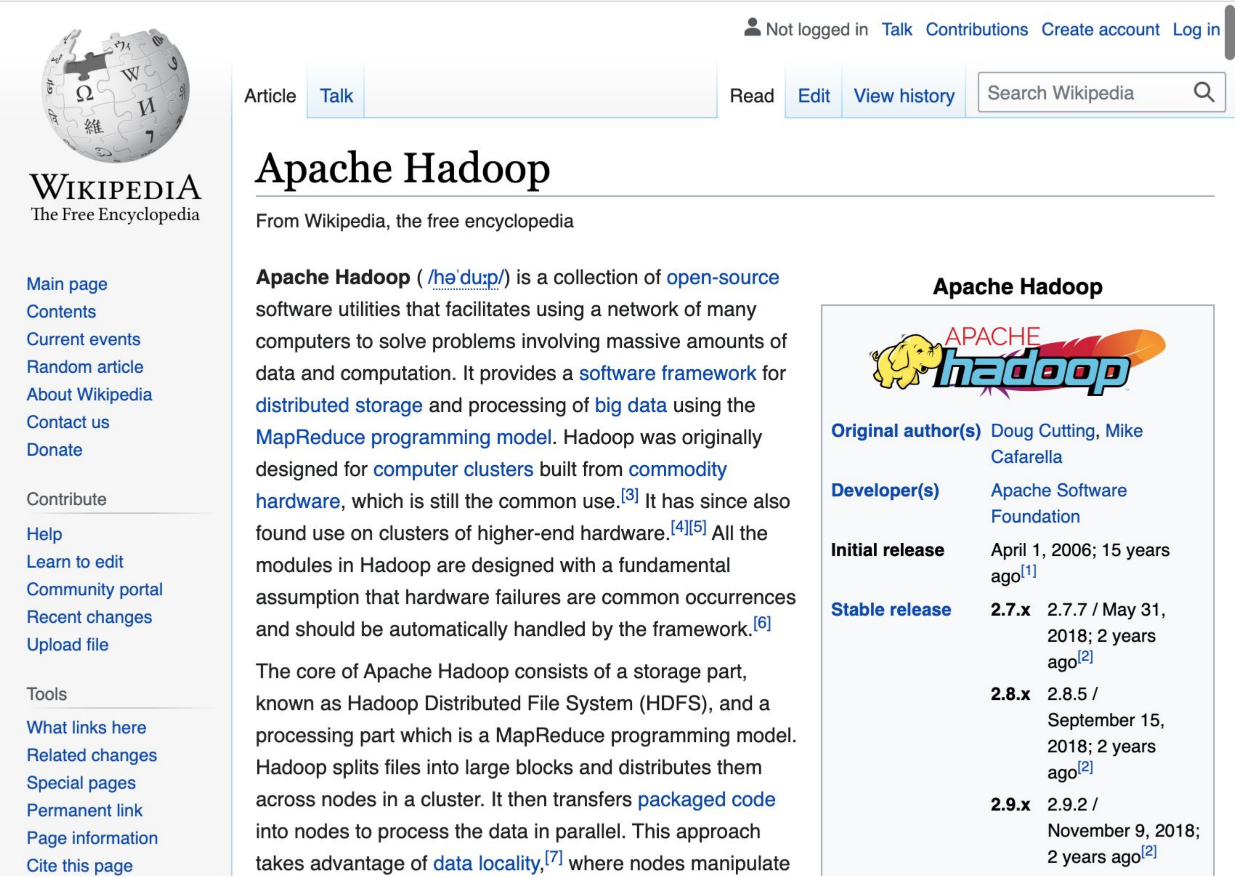
Пример страницы Википедии



Contents [hide]

1 History
2 Architecture
  2.1 File systems
    2.1.1 Hadoop distributed file system
    2.1.2 Other file systems
  2.2 JobTracker and TaskTracker: the MapReduce engine
    2.2.1 Scheduling
      2.2.1.1 Fair scheduler
      2.2.1.2 Capacity scheduler
  2.3 Difference between Hadoop 1 and Hadoop 2 (YARN)
  2.4 Difference between Hadoop 2 and Hadoop 3
  2.5 Other applications
3 Prominent use cases
4 Hadoop hosting in the cloud
5 Commercial support
  5.1 Branding
6 Papers
7 See also
8 References
9 Bibliography
10 External links

History [edit]



Apache Hadoop

From Wikipedia, the free encyclopedia

Apache Hadoop (/həˈduːp/) is a collection of open-source software utilities that facilitates using a network of many computers to solve problems involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data using the MapReduce programming model. Hadoop was originally designed for computer clusters built from commodity hardware, which is still the common use.[3] It has since also found use on clusters of higher-end hardware.[4][5] All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common occurrences and should be automatically handled by the framework.[6]

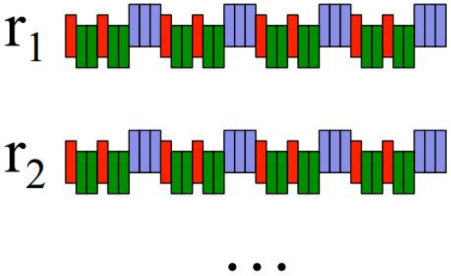
The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part which is a MapReduce programming model. Hadoop splits files into large blocks and distributes them across nodes in a cluster. It then transfers packaged code into nodes to process the data in parallel. This approach takes advantage of data locality,[7] where nodes manipulate

Apache Hadoop

Original author(s)	Doug Cutting, Mike Cafarella	
Developer(s)	Apache Software Foundation	
Initial release	April 1, 2006; 15 years ago[1]	
Stable release	**2.7.x**	2.7.7 / May 31, 2018; 2 years ago[2]
	2.8.x	2.8.5 / September 15, 2018; 2 years ago[2]
	2.9.x	2.9.2 / November 9, 2018; 2 years ago[2]

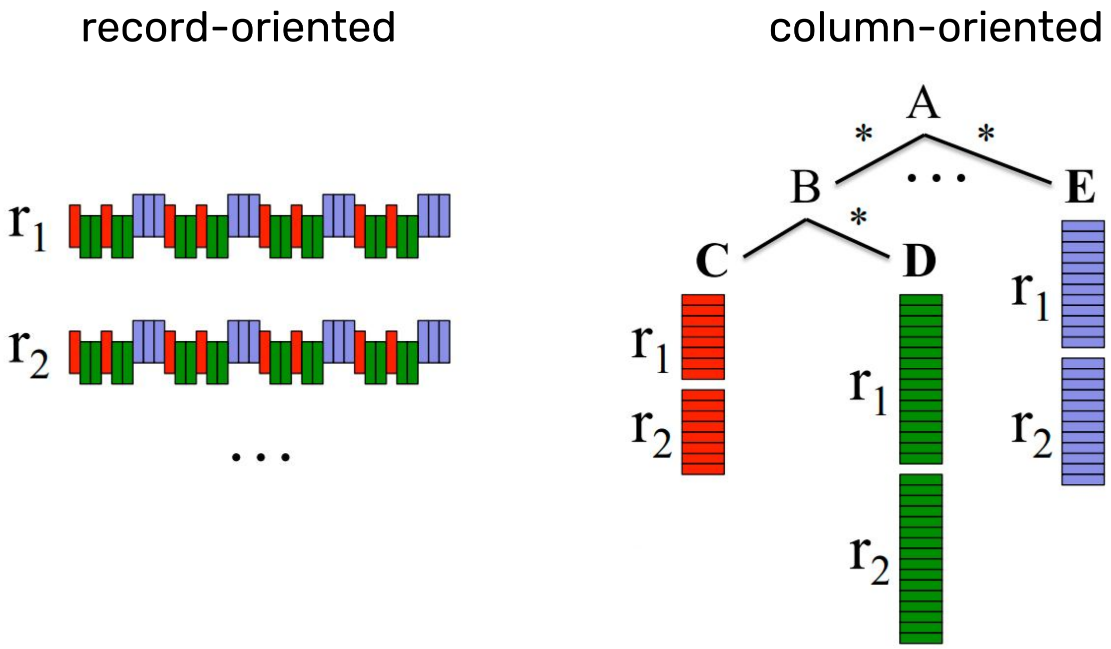
Google Dremel

record-oriented



https://research.google/pubs/pub36632/

Google Dremel

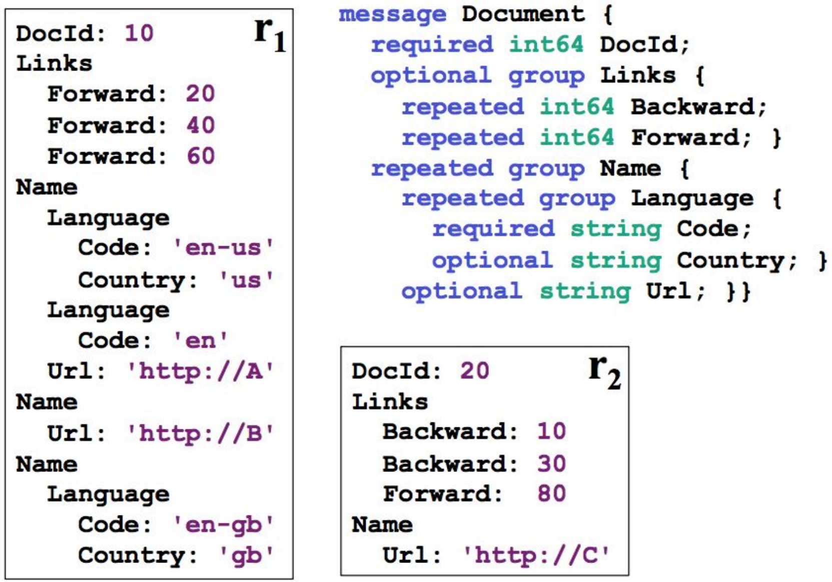


record-oriented

column-oriented

https://research.google/pubs/pub36632/

Google Dremel*



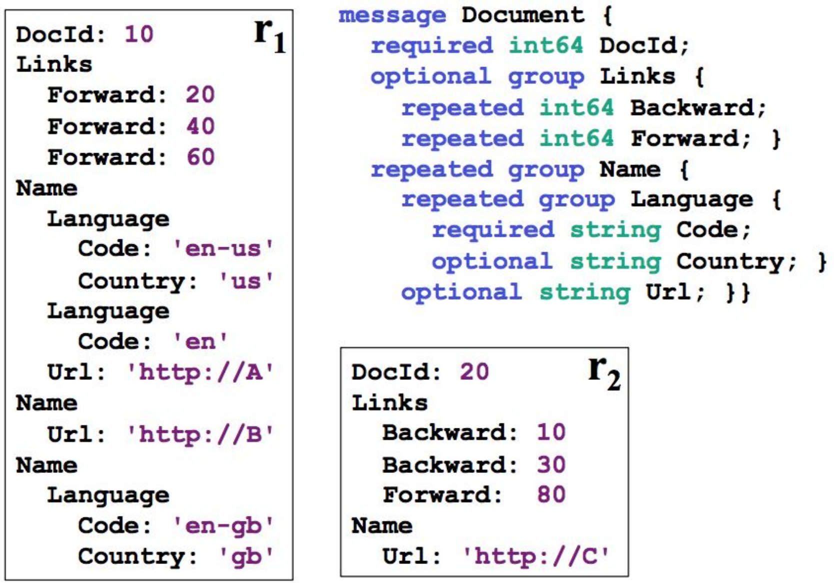
```
DocId: 10                r1
Links
  Forward: 20
  Forward: 40
  Forward: 60
Name
  Language
    Code: 'en-us'
    Country: 'us'
  Language
    Code: 'en'
  Url: 'http://A'
Name
  Url: 'http://B'
Name
  Language
    Code: 'en-gb'
    Country: 'gb'
```

```
message Document {
  required int64 DocId;
  optional group Links {
    repeated int64 Backward;
    repeated int64 Forward; }
  repeated group Name {
    repeated group Language {
      required string Code;
      optional string Country; }
    optional string Url; }}
```

```
DocId: 20                r2
Links
  Backward: 10
  Backward: 30
  Forward:  80
Name
  Url: 'http://C'
```

Figure 2: Two sample nested records and their schema

Google Dremel*



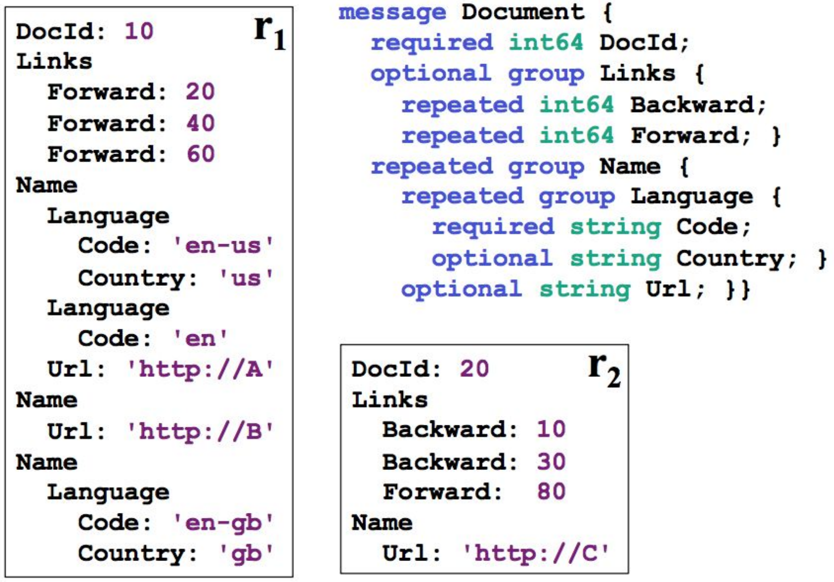
```
DocId: 10                r1
Links
  Forward: 20
  Forward: 40
  Forward: 60
Name
  Language
    Code: 'en-us'
    Country: 'us'
  Language
    Code: 'en'
  Url: 'http://A'
Name
  Url: 'http://B'
Name
  Language
    Code: 'en-gb'
    Country: 'gb'
```

```
message Document {
  required int64 DocId;
  optional group Links {
    repeated int64 Backward;
    repeated int64 Forward; }
  repeated group Name {
    repeated group Language {
      required string Code;
      optional string Country; }
    optional string Url; }}
```

```
DocId: 20                r2
Links
  Backward: 10
  Backward: 30
  Forward:  80
Name
  Url: 'http://C'
```

Figure 2: Two sample nested records and their schema

Google Dremel*

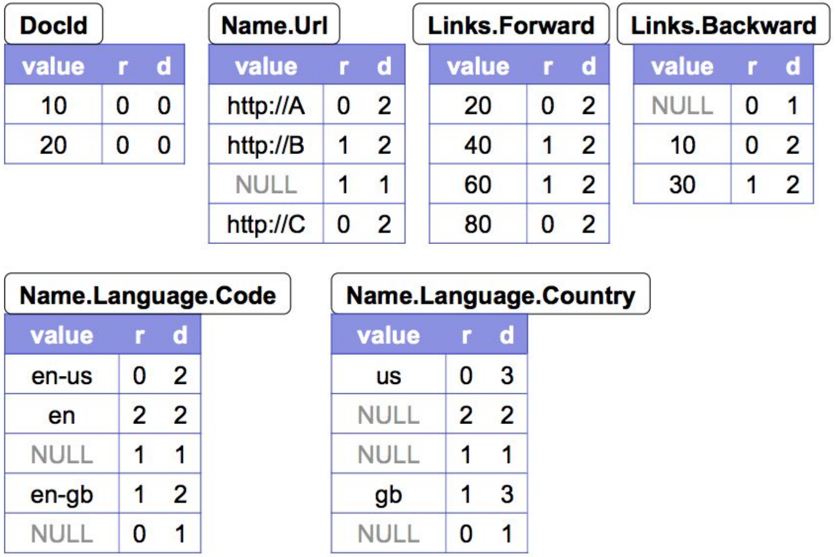


```
DocId: 10                r1
Links
  Forward: 20
  Forward: 40
  Forward: 60
Name
  Language
    Code: 'en-us'
    Country: 'us'
  Language
    Code: 'en'
  Url: 'http://A'
Name
  Url: 'http://B'
Name
  Language
    Code: 'en-gb'
    Country: 'gb'
```

```
message Document {
  required int64 DocId;
  optional group Links {
    repeated int64 Backward;
    repeated int64 Forward; }
  repeated group Name {
    repeated group Language {
      required string Code;
      optional string Country; }
    optional string Url; }}
```

```
DocId: 20                r2
Links
  Backward: 10
  Backward: 30
  Forward:  80
Name
  Url: 'http://C'
```

Figure 2: Two sample nested records and their schema



DocId

value	r	d
10	0	0
20	0	0

Name.Url

value	r	d
http://A	0	2
http://B	1	2
NULL	1	1
http://C	0	2

Links.Forward

value	r	d
20	0	2
40	1	2
60	1	2
80	0	2

Links.Backward

value	r	d
NULL	0	1
10	0	2
30	1	2

Name.Language.Code

value	r	d
en-us	0	2
en	2	2
NULL	1	1
en-gb	1	2
NULL	0	1

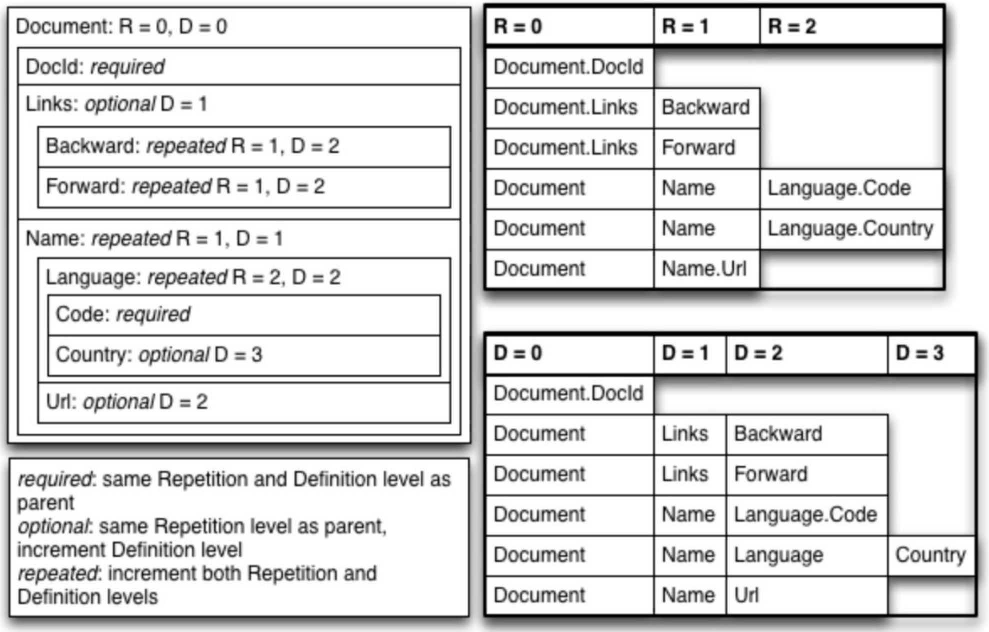
Name.Language.Country

value	r	d
us	0	3
NULL	2	2
NULL	1	1
gb	1	3
NULL	0	1

Figure 3: Column-striped representation of the sample data in Figure 2, showing repetition levels (r) and definition levels (d)

Лайфхаки для понимания

- repetition и definition levels



Document: R = 0, D = 0

DocId: *required*

Links: *optional* D = 1

Backward: *repeated* R = 1, D = 2

Forward: *repeated* R = 1, D = 2

Name: *repeated* R = 1, D = 1

Language: *repeated* R = 2, D = 2

Code: *required*

Country: *optional* D = 3

Url: *optional* D = 2

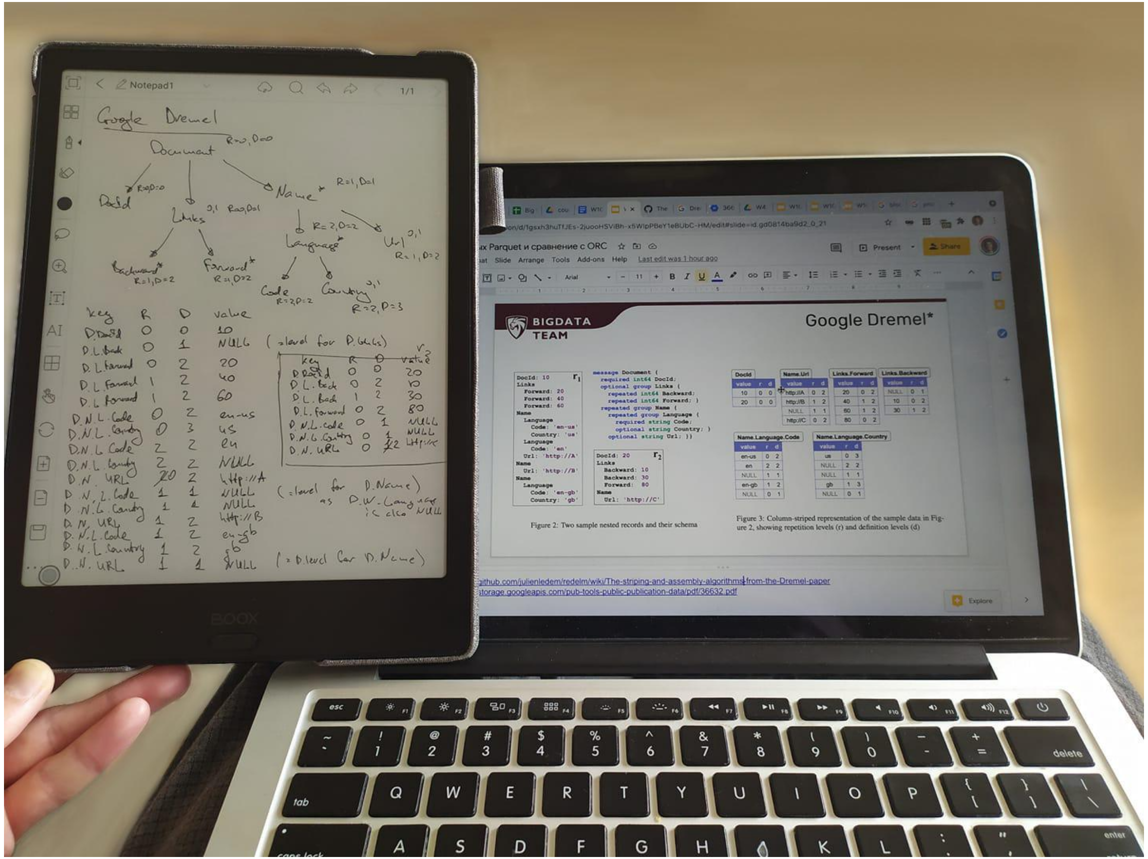
required: same Repetition and Definition level as parent
optional: same Repetition level as parent, increment Definition level
repeated: increment both Repetition and Definition levels

R = 0	**R = 1**	**R = 2**
Document.DocId		
Document.Links	Backward	
Document.Links	Forward	
Document	Name	Language.Code
Document	Name	Language.Country
Document	Name.Url	

D = 0	**D = 1**	**D = 2**	**D = 3**
Document.DocId			
Document	Links	Backward	
Document	Links	Forward	
Document	Name	Language.Code	
Document	Name	Language	Country
Document	Name	Url	

[The striping and assembly algorithms from the Dremel paper]

Q&A: найдите 1 ошибку



key	R	D	value
D.DocId	0	0	10
D.L.Back	0	1	NULL (=level for D.Links)
D.L.Forward	0	2	20
D.L.Forward	1	2	40
D.L.Forward	1	2	60
D.N.L.Code	0	2	en-us
D.N.L.Country	0	3	us
D.N.L.Code	2	2	en
D.N.L.Country	2	2	NULL
D.N.URL	~~1~~0	2	http://A
D.N.L.Code	1	1	NULL (=level for D.Name) as D.N.Language is also NULL
D.N.L.Country	1	1	NULL
D.N.URL	1	2	http://B
D.N.L.Code	1	2	en-gb
D.N.L.Country	1	2	gb
D.N.URL	1	1	NULL (= D.level for D.Name)

r_2

key	R	D	value
D.DocId	0	0	20
D.L.Back	0	2	10
D.L.Back	1	2	30
D.L.forward	0	2	80
D.N.L.code	0	1	NULL
D.N.L.Country	0	1	NULL
D.N.URL	0	~~1~~2	http://C

ix Parquet и сравнение с ORC

Slide Arrange Tools Add-ons Help Last edit was 1 hour ago

Google Dremel*

```
DocId: 10                r1
Links
  Forward: 20
  Forward: 40
  Forward: 60
Name
  Language
    Code: 'en-us'
    Country: 'us'
  Language
    Code: 'en'
  Url: 'http://A'
Name
  Url: 'http://B'
Name
  Language
    Code: 'en-gb'
    Country: 'gb'
```

```
message Document {
  required int64 DocId;
  optional group Links {
    repeated int64 Backward;
    repeated int64 Forward; }
  repeated group Name {
    repeated group Language {
      required string Code;
      optional string Country; }
    optional string Url; }}
```

```
DocId: 20                r2
Links
  Backward: 10
  Backward: 30
  Forward: 80
Name
  Url: 'http://C'
```

Figure 2: Two sample nested records and their schema

DocId

value	r	d
10	0	0
20	0	0

Name.Url

value	r	d
http://A	0	2
http://B	1	2
NULL	1	1
http://C	0	2

Links.Forward

value	r	d
20	0	2
40	1	2
60	1	2
80	0	2

Links.Backward

value	r	d
NULL	0	1
10	0	2
30	1	2

Name.Language.Code

value	r	d
en-us	0	2
en	2	2
NULL	1	1
en-gb	1	2
NULL	0	1

Name.Language.Country

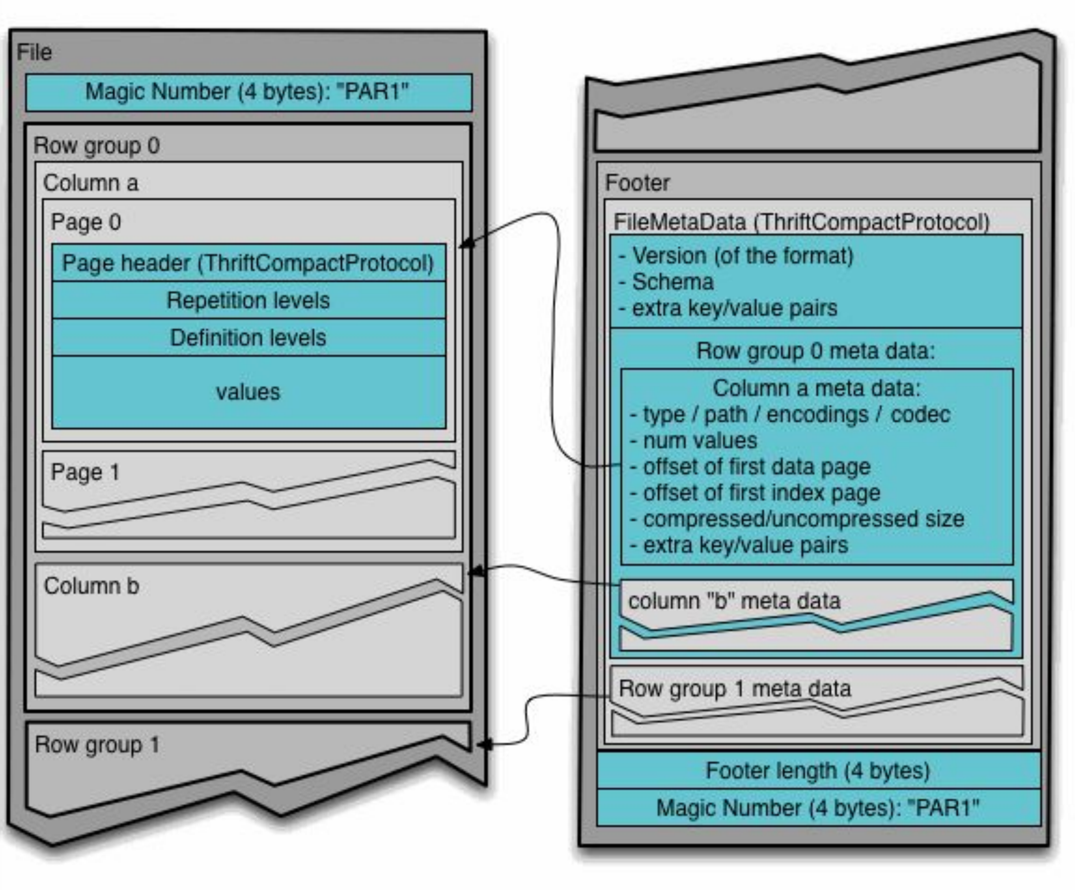
value	r	d
us	0	3
NULL	2	2
NULL	1	1
gb	1	3
NULL	0	1

Figure 3: Column-striped representation of the sample data in Figure 2, showing repetition levels (r) and definition levels (d)

github.com/julienledem/redelm/wiki/The-striping-and-assembly-algorithms-from-the-Dremel-paper

storage.googleapis.com/pub-tools-public-publication-data/pdf/36632.pdf

Внутренности Parquet



File

Magic Number (4 bytes): "PAR1"

Row group 0

Column a

Page 0

Page header (ThriftCompactProtocol)

Repetition levels

Definition levels

values

Page 1

Column b

Row group 1

Footer

FileMetaData (ThriftCompactProtocol)

- Version (of the format)
- Schema
- extra key/value pairs

Row group 0 meta data:

Column a meta data:

- type / path / encodings / codec
- num values
- offset of first data page
- offset of first index page
- compressed/uncompressed size
- extra key/value pairs

column "b" meta data

Row group 1 meta data

Footer length (4 bytes)

Magic Number (4 bytes): "PAR1"

- File ~ блок в HDFS
- Row group ~ stripe
- Column Chunk
- Page ~ Compression Chunk

Сравнение

Сравнение

	ORC	**Parquet**
Поколоночное хранение	✓	✓

Сравнение

	ORC	Parquet
Поколоночное хранение	✓	✓
Версионирование схемы	✓ (protobuf)	✓ (thrift)

Сравнение

	ORC	Parquet
Поколоночное хранение	✓	✓
Версионирование схемы	✓ (protobuf)	✓ (thrift)
Статистики и фильтры Блума	✓✓	✓❌

Сравнение

	ORC	Parquet
Поколоночное хранение	✓	✓
Версионирование схемы	✓ (protobuf)	✓ (thrift)
Статистики и фильтры Блума	✓✓	✓❌
Типы данных	14+	7 + nested

Сравнение

	ORC	Parquet
Поколоночное хранение	✓	✓
Версионирование схемы	✓ (protobuf)	✓ (thrift)
Статистики и фильтры Блума	✓✓	✓❌
Типы данных	14+	7 + nested
Ориентация на мир Hadoop и world-wide	✓❌	✓✓