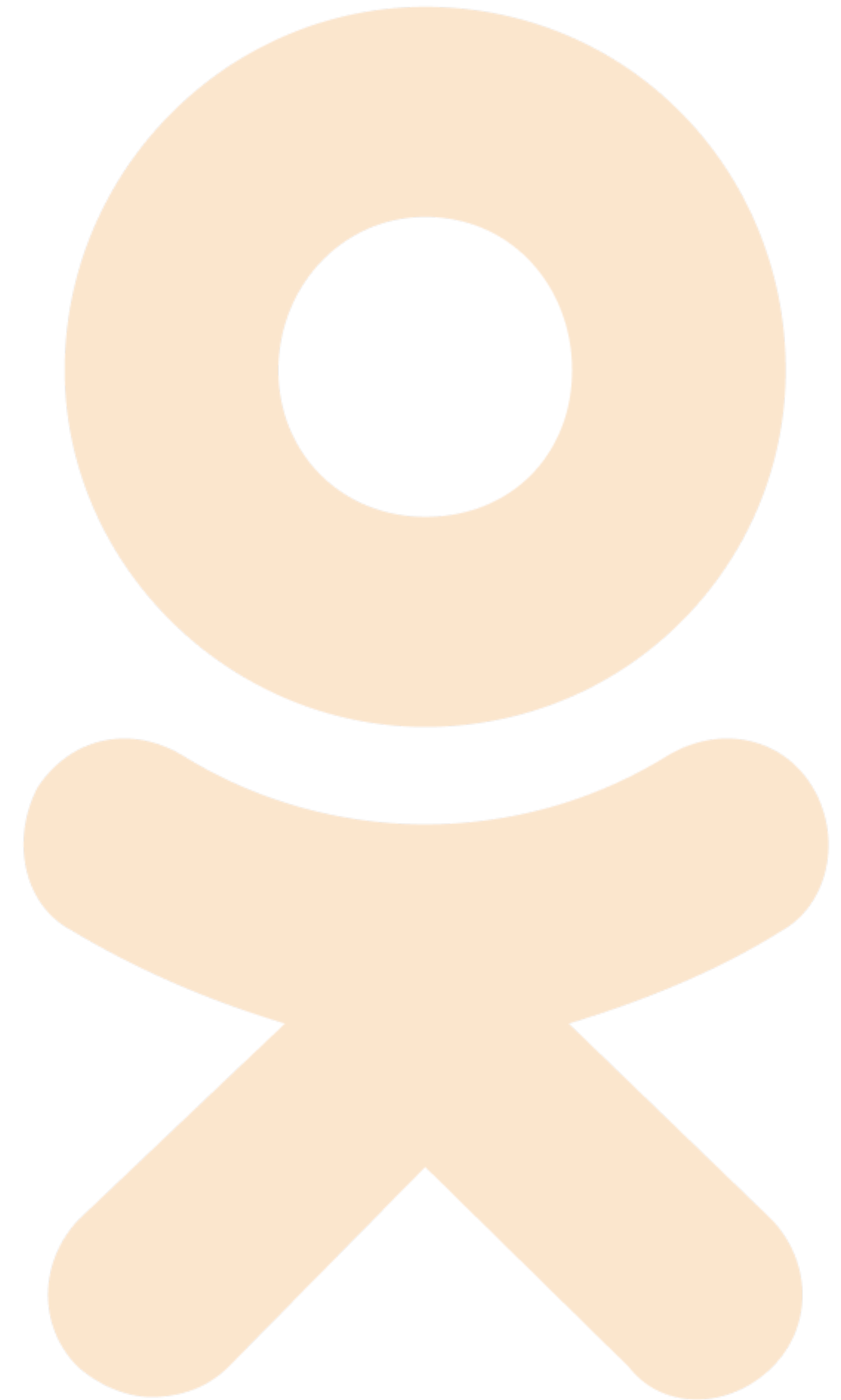


Hadoop 3: Erasure Coding Catastrophe

Денис Ефаров

Опыт реального применения ЕС на масштабе в 100 PB



О следователях



КОМАНДА

DataPlatform - 6 человек

Разрабатываем собственные
инструменты для приема и
обработки данных

Адаптируем и поддерживаем
сторонние решения

ДОКЛАДЧИК

Когда-то Java - разработчик

Последние 7+ лет - в BigData с Hadoop

4 года в ОК, ведущий разработчик

О месте происхождения



ОКРУЖЕНИЕ

Kafka
200+ брокеров

Hadoop
1000 серверов

Clickhouse
100 серверов



ТЕХНОЛОГИИ И ЯЗЫКИ

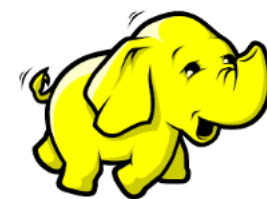
Centos

Java 8+, Scala 2.11+

Kafka 2.4+

Hadoop 2.7.3 и 3.1.4

Spark 2.3+



НАГРУЗКА

Десятки ТВ / день

Hadoop 150 PB

Clickhouse 1 PB

~2000 расчетов



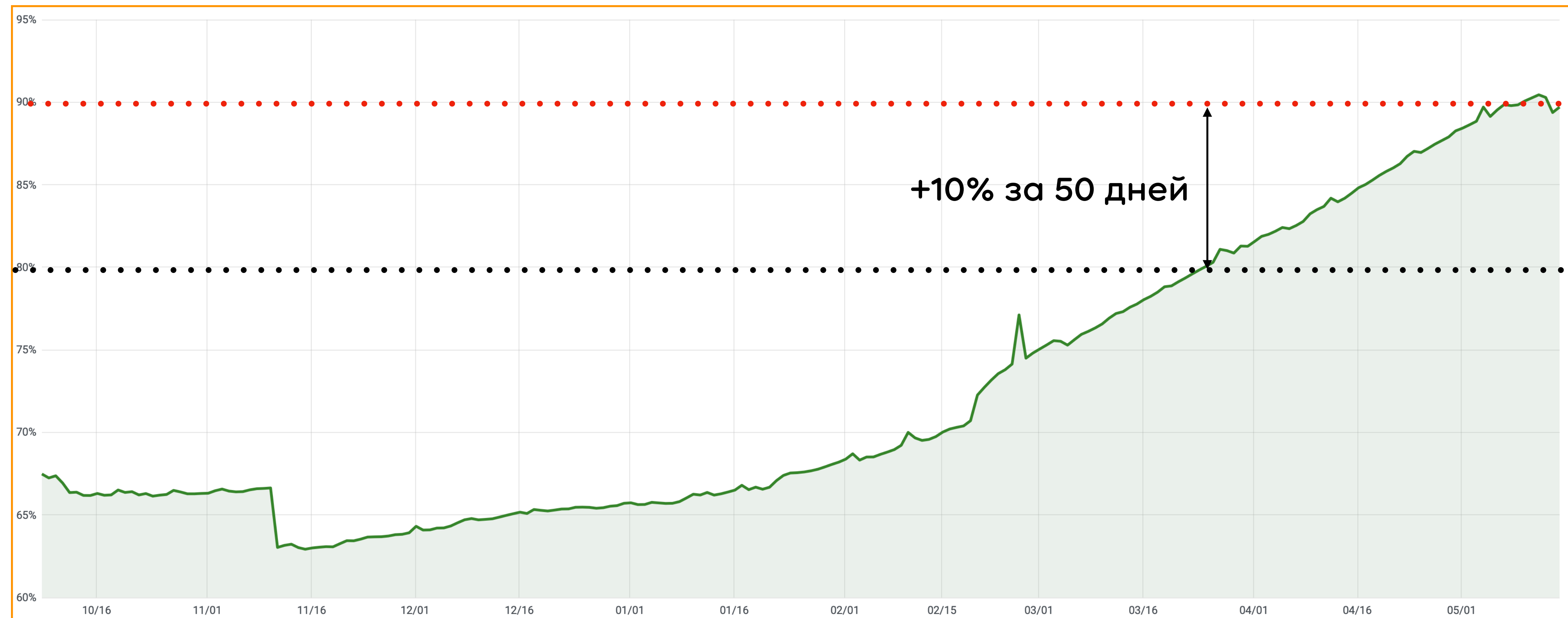


Предыстория

Ситуация



- Петабайты данных + постоянный рост



Почему так много?



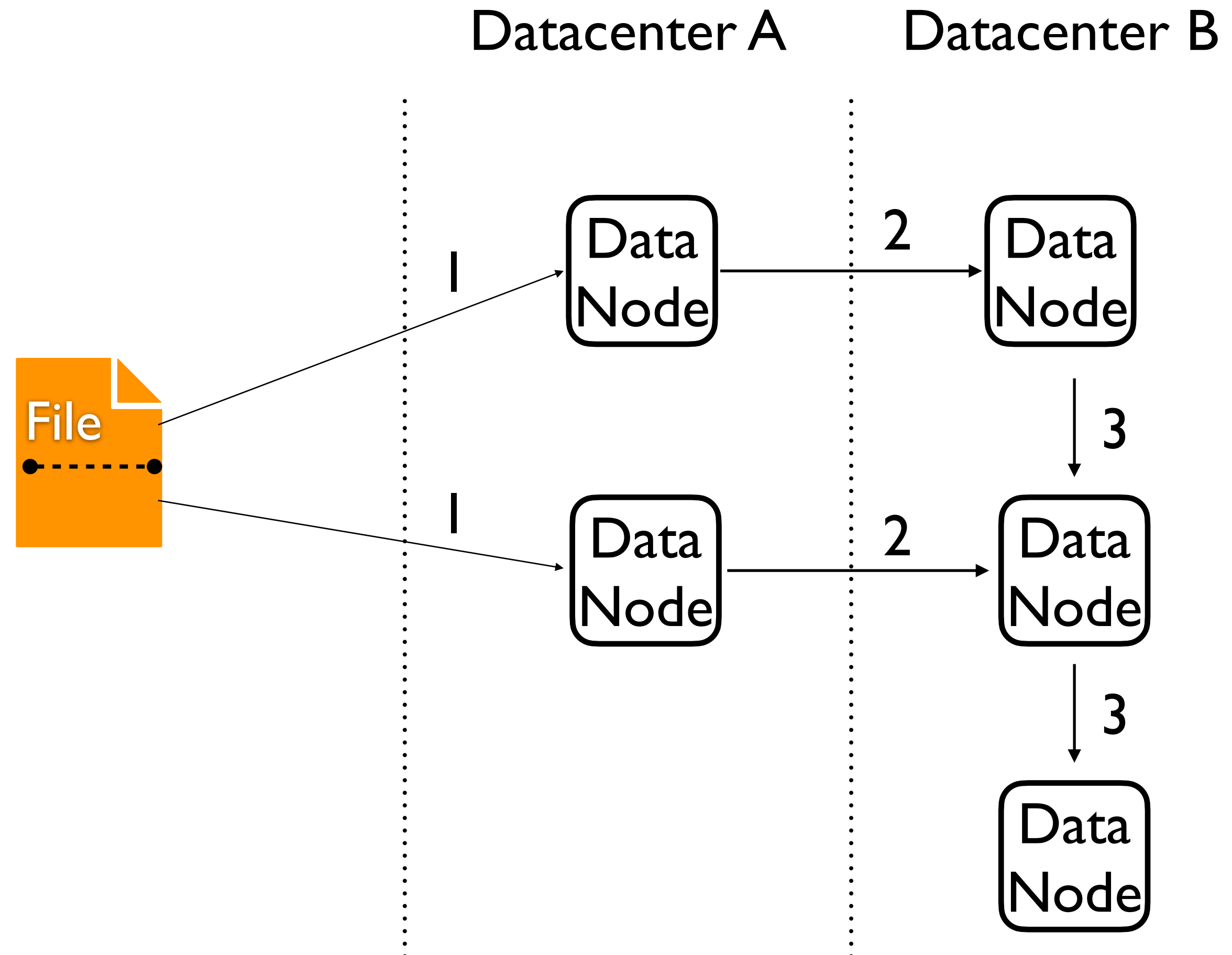
- Огромный проект
 - Десятки сервисов
 - Десятки миллионов пользователей
 - Терабайты статистики в день
 - Храним историю за 10+ лет
- Дубли с разных стадий обработки
- Производные расчеты



Почему так много?



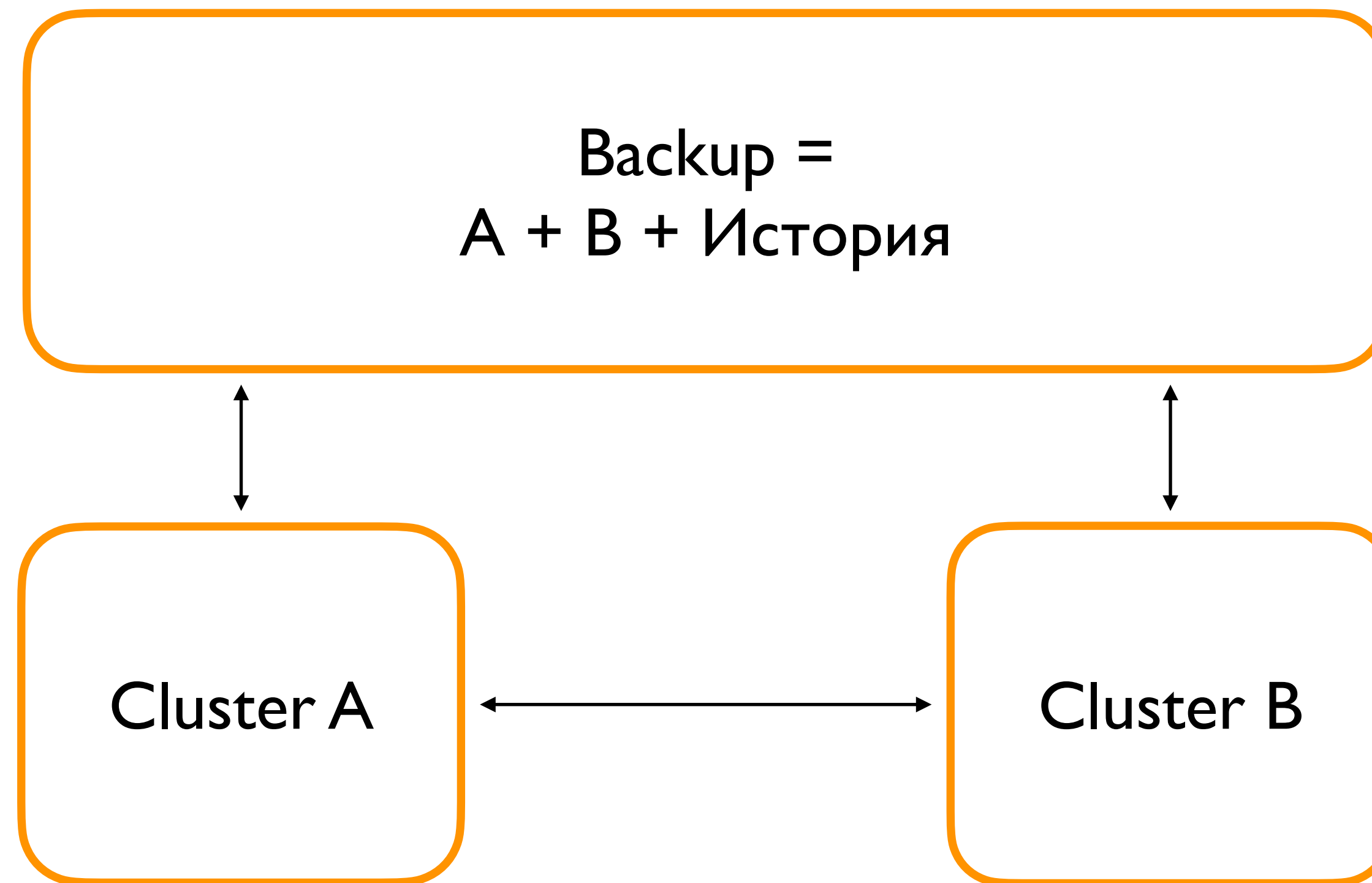
- Репликация x3



Почему так много?



- Бэкапы x2
- $x2 * x3 = x6$



Что делать?



- Ограничить рост?
- Выбрать форматы получше?
- Забросать проблему железом?
- Отказаться от бэкапов?
- Уменьшать репликацию?



Есть вариант



- Hadoop 3 принес Erasure Coding вместо репликации



Hadoop HDFS / HDFS-7285

Erasure Coding Support inside HDFS

▼ Details

Type:  New Feature

Priority:  Major

Affects Version/s: None

Status: **RESOLVED**

Resolution: Fixed

Fix Version/s: 3.0.0-alpha1



Изучаем



- Коды Рида-Соломона или XOR на выбор
- Дефолт - коды Рида-Соломона, RS-6-3-1024k :
 - Избыточность на диске всего 50%
 - Гарантии как у RF3
 - CPU (4x-5x) + Network IO overhead
 - при записи и восстановлении
- Экономия - десятки PB!



Применяем



- Hadoop 3 - в тестирование
- Проверяем ЕС
 - Успешно!
 - Но займет 8-10 месяцев для backup

Release 3.1.2 available

2019 Feb 6

This is the second stable release of Apache Hadoop 3.1 line.



Применяем



1. `mkdir /tmp/data`
2. `setPolicy /tmp/data RS-6-3-1024k`
3. `cp /data /tmp/data`
4. `mv /data /tmp/data.bak`
5. `mv /tmp/data /data`
6. `rm -r /tmp/data.bak`

Warning: setting erasure coding policy on a non-empty directory will not automatically convert existing files to RS-6-3-1024k erasure coding policy

Успех



- Экономия - работает
 - 5+ ТВ / сутки истории
- Перестаем заказывать железо





История

Интересный случай



- Не читается Parquet-файл
- Обрыв записи? Рутина

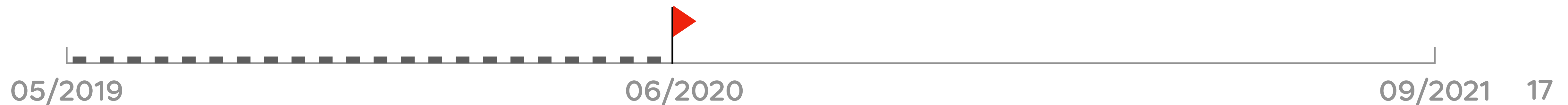
```
/part-00984-28877d33-f972-490f-91e1-17db0b4f7720-c000.gz.parquet is not a Parquet file. expected magic number at  
tail [80, 65, 82, 49] but found [49, 15, 14, -21]  
at org.apache.parquet.hadoop.ParquetFileReader.readFooter(ParquetFileReader.java:476)  
at org.apache.parquet.hadoop.ParquetFileReader.readFooter(ParquetFileReader.java:445)  
at org.apache.parquet.hadoop.ParquetFileReader.readFooter(ParquetFileReader.java:421)
```



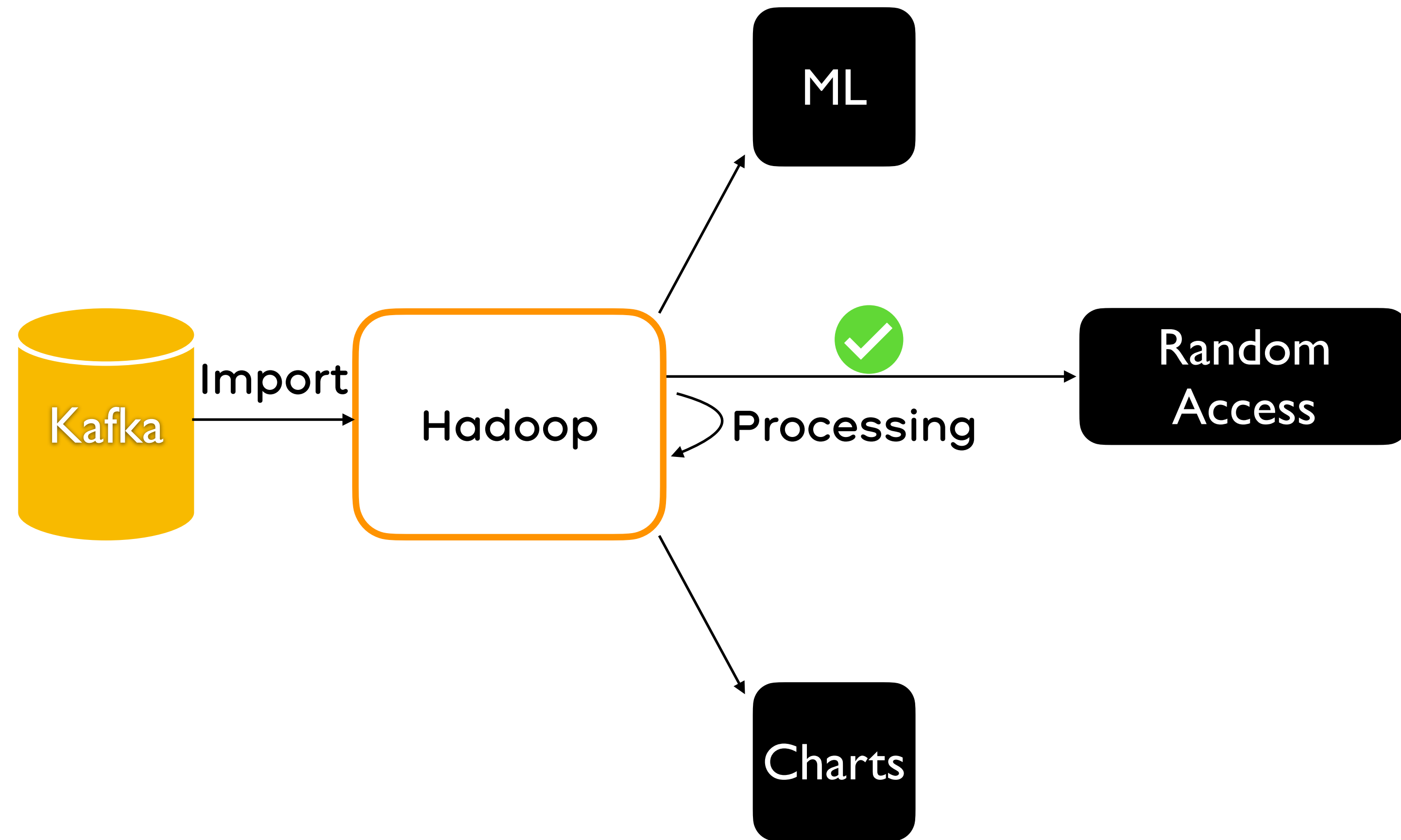
6 месяцев тишины...



- Конверсия идет, проблем нет
- И снова то же самое
 - Баг в Parquet?
 - Баг в Spark?
 - Сломанные диски?



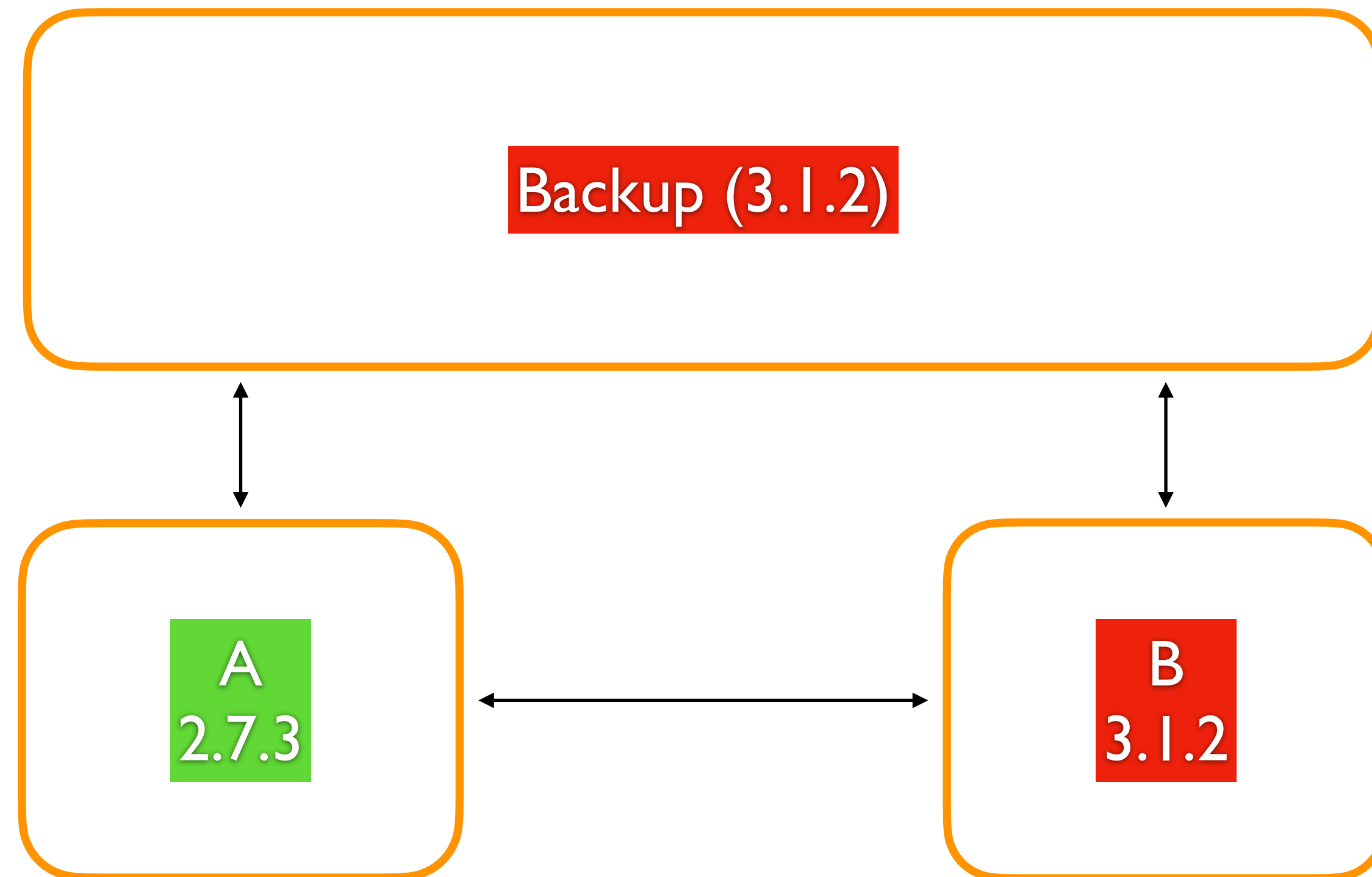
Где происходит?



Где происходит?



- Сканируем случайные папки
- Повреждения в 83 / 2454 папок, 3.3%
- случайные Parquet-файлы
- только 3 Hadoop!



Где происходит?



- Отличаются полосы размером 1 Мб
- Которые идут через 6!
- Отличается 29548791 байт из ~177 Мб
 - Ровно 1/6 !

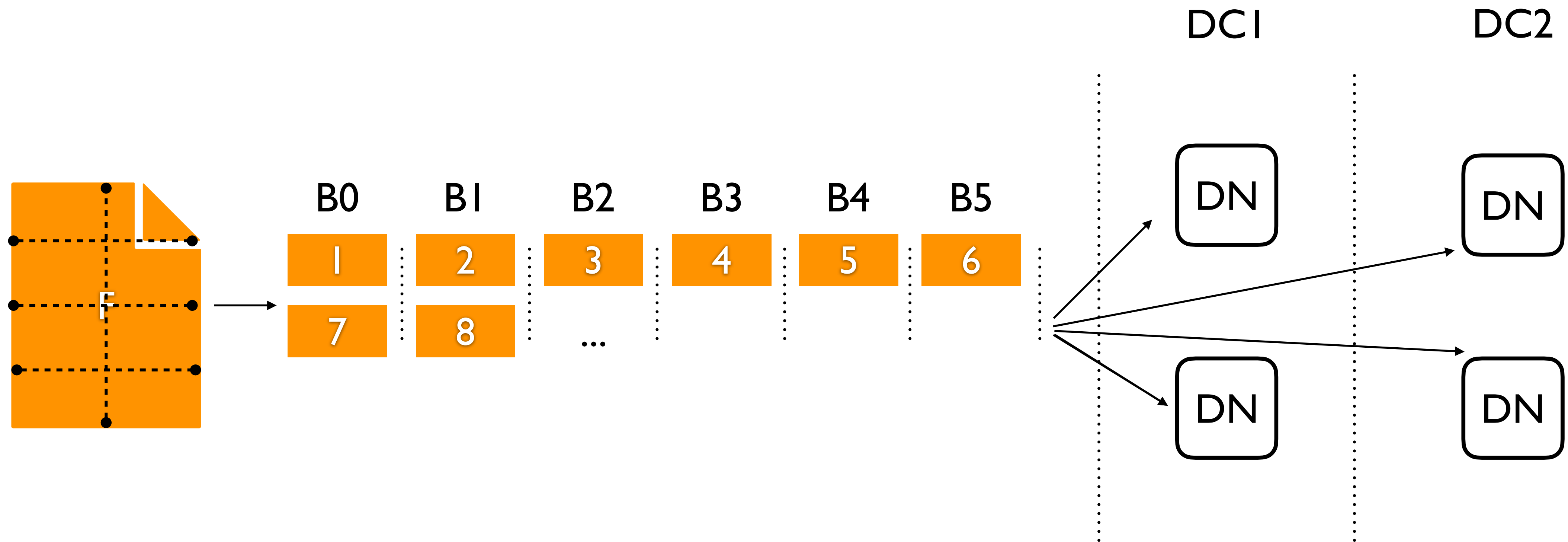
4194289	214	332
4194290	366	174
4194291	120	23
4194292	74	155
4194293	107	115
4194294	146	263
4194295	362	56
4194296	340	120
4194297	341	12
4194298	145	334
4194299	275	150
4194300	26	337
4194301	205	231
4194302	17	273
4194303	244	164
4194304	35	215
9437185	200	126
9437186	326	104
9437187	4	243
9437188	41	10
9437189	127	142
9437190	155	275
9437191	304	177
9437192	205	372
9437193	35	32
9437194	364	327
9437195	275	162
9437196	112	341
9437197	117	156
9437198	136	35
9437199	45	16
9437200	114	357
9437201	343	26
9437202	151	204
9437203	173	130
9437204	104	247





Грустная инженерия pt. 1

Striped layout



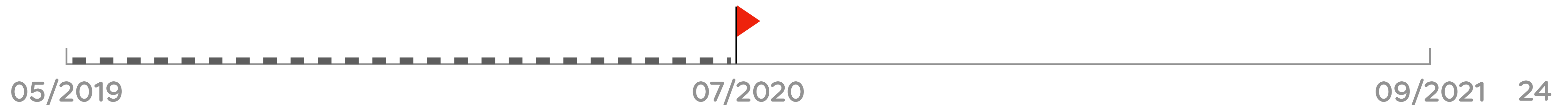


История

Обстановка



- В 2 / 3 кластеров системно теряются данные
 - Hadoop 2 не имел Striped Layout
- Ошибка уровнем ниже формата
 - Структура повреждений не связана с форматом
- Ищем баги

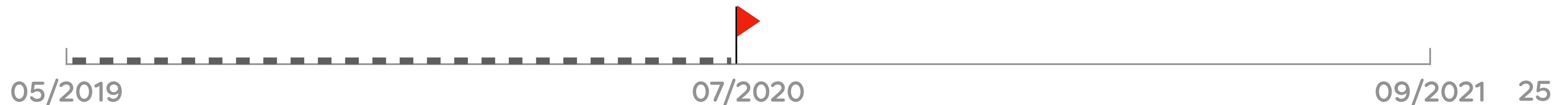


- <https://issues.apache.org/jira/browse/HDFS-14768>

 **Hadoop HDFS** / HDFS-14768
EC : Busy DN replica should be consider in live replica check.

▼ Details

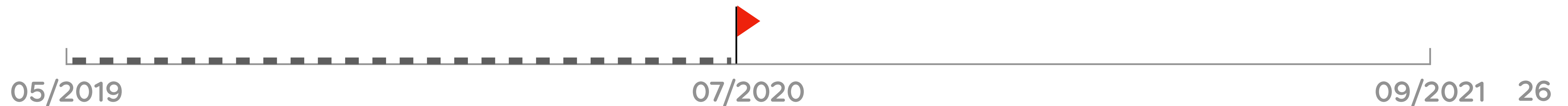
Type:	 Bug	Status:	RESOLVED
Priority:	 Major	Resolution:	Fixed
Affects Version/s:	3.0.2	Fix Version/s:	3.3.0
Component/s:	datanode, erasure-coding, hdfs, namenode		
Labels:	patch		
Target Version/s:	3.3.0		



- <https://issues.apache.org/jira/browse/HDFS-14768>

✓  guojh added a comment - 23/Aug/19 05:48

Hemanth Boyina Thanks for you attention. I post the detail Description above. The bug Damaged our production data, I think it is terrible.



- <https://issues.apache.org/jira/browse/HDFS-15186>



Hadoop HDFS / HDFS-15186

Erasure Coding: Decommission may generate the parity block's content with all 0 in some case

▼ Details

Type:  Bug
 Priority:  Critical
 Affects Version/s: 3.0.3, 3.2.1, 3.1.3
 Component/s: [datanode](#), [erasure-coding](#)
 Labels: None
 Target Version/s: [3.3.0](#), [3.1.4](#), [3.2.2](#)
 Hadoop Flags: Reviewed
 Flags: Patch, Important

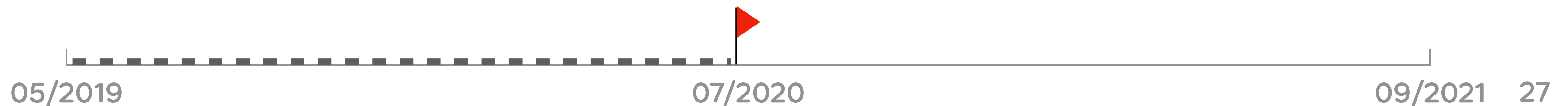
Status: **RESOLVED**
 Resolution: Fixed
 Fix Version/s: [3.3.0](#)

▼ People

Assignee:
 Reporter:
 Votes:
 Watchers:

▼ Dates

Created:
 Updated:
 Resolved:

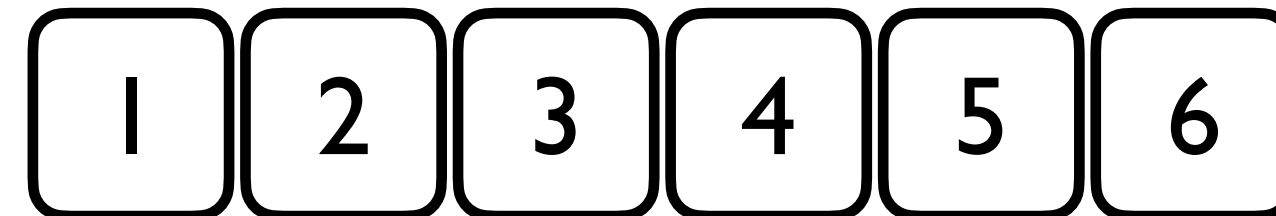


Баг - матчасть



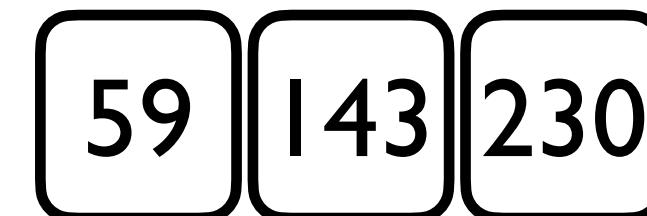
RS 6-3

Data:

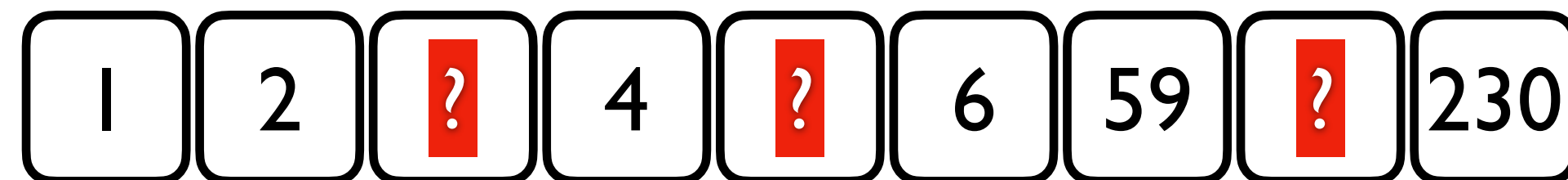


+

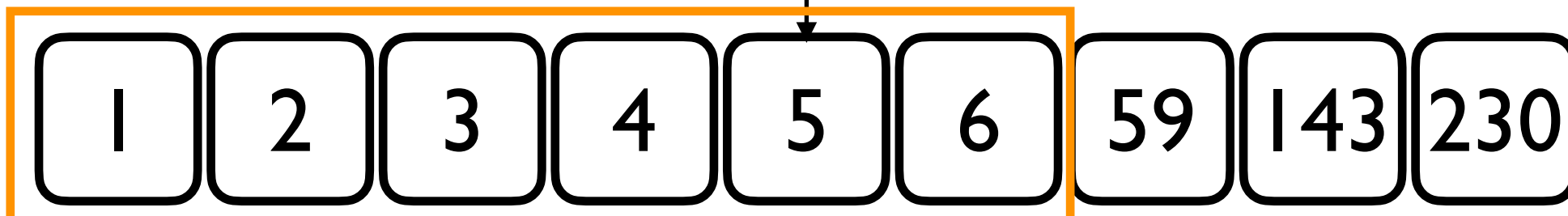
Parity:



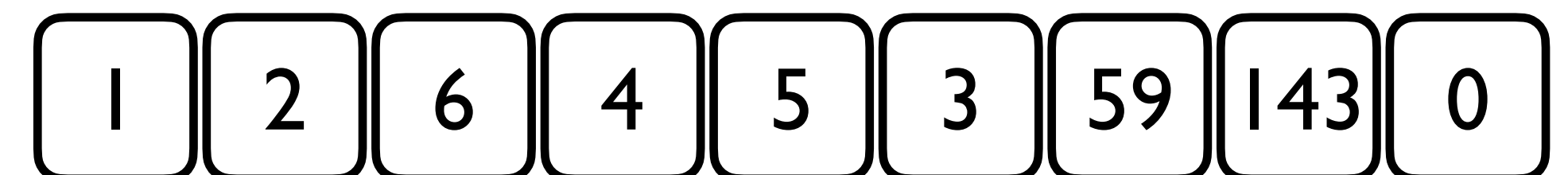
Восстановление по известным индексам



RS



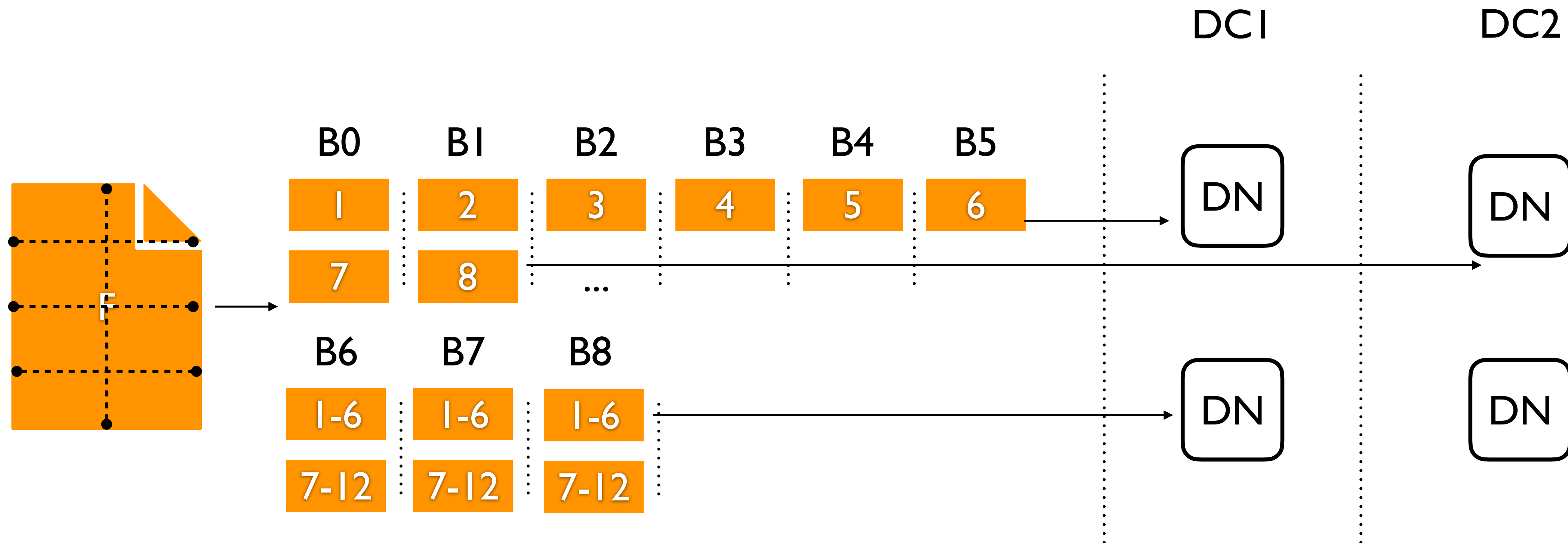
Детекция наличия ошибок



RS

В данных есть ошибки!

Баг - матчасть

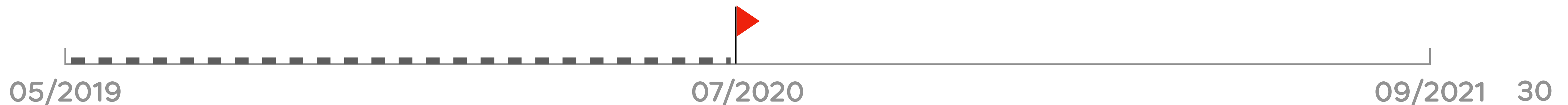


Баг - суть

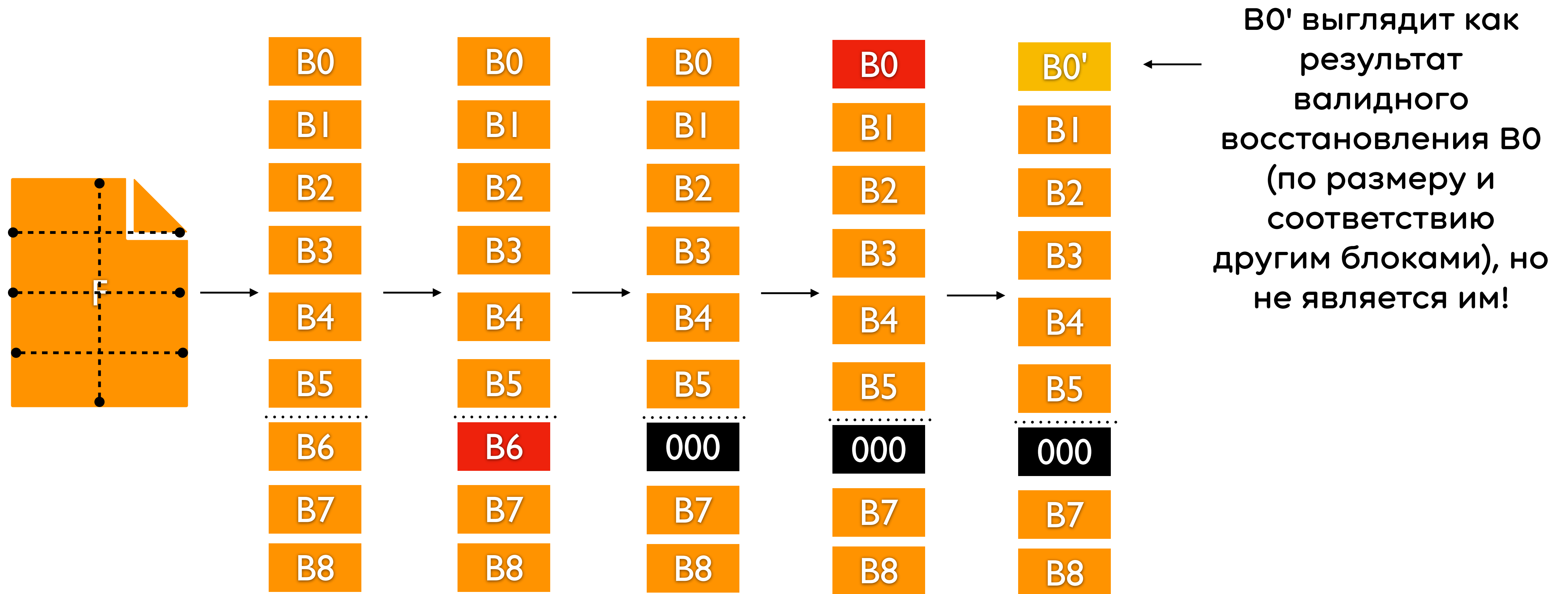


- Ошибка в числе блоков на восстановление
 - Нужен [6] блок, а в задаче [6, 0]
- Реализация RS в Hadoop зависима от порядка индексов
 - [0, 6] - корректно, [6, 0] - некорректно
- Ошибка не приводит к исключениям

```
org.apache.hadoop.io.erasurecode.rawcoder.RSRawDecoder.generateDecodeMatrix(int[] erasedIndexes)
```



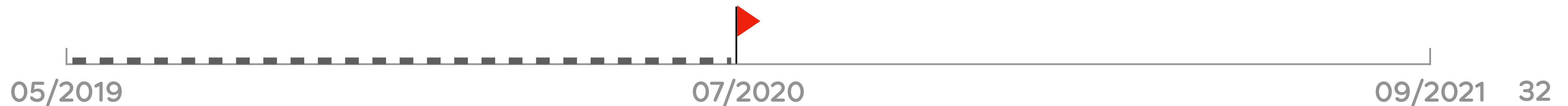
Баг - суть



Последствия



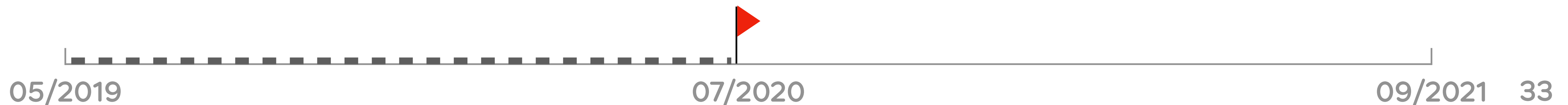
- Петабайты файлов в ЕС - потенциальный фарш
- Масштаб повреждений неизвестен
- Хранить данные в ЕС нельзя!



Что делать???



- Чинить баг самим?
 - Файлы уже повреждены
 - Пока чиним, повредится еще
 - Могут быть другие
- Файлы в ЕС - срочно назад
- Найти способ определять поврежденных
- Найти способ восстанавливать поврежденных!



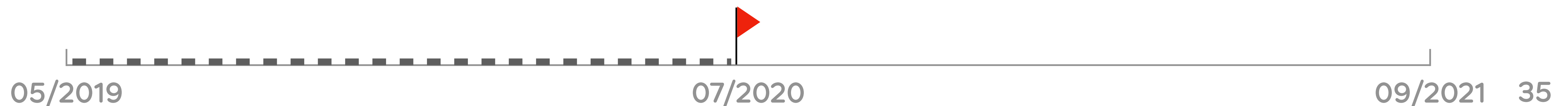


Грустная инженерия pt.2

Асимметрия Фарша



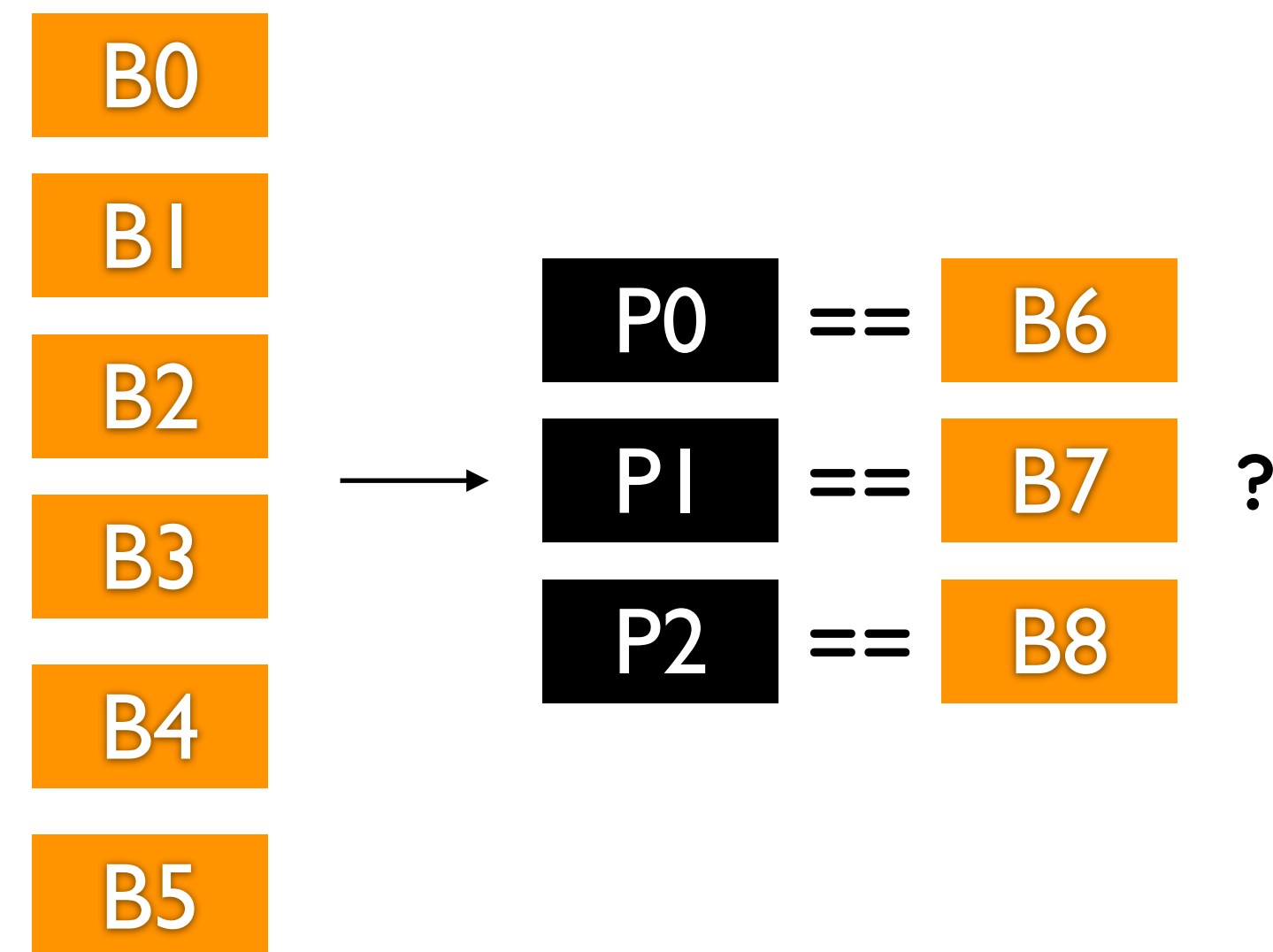
- При кодировании в ЕС размер уменьшается
 - А назад растёт. А места - нет
- При кодировании в ЕС достаточно просто закодировать
 - А при декодировании - проверить, что оригинал не был битым!



Детекция



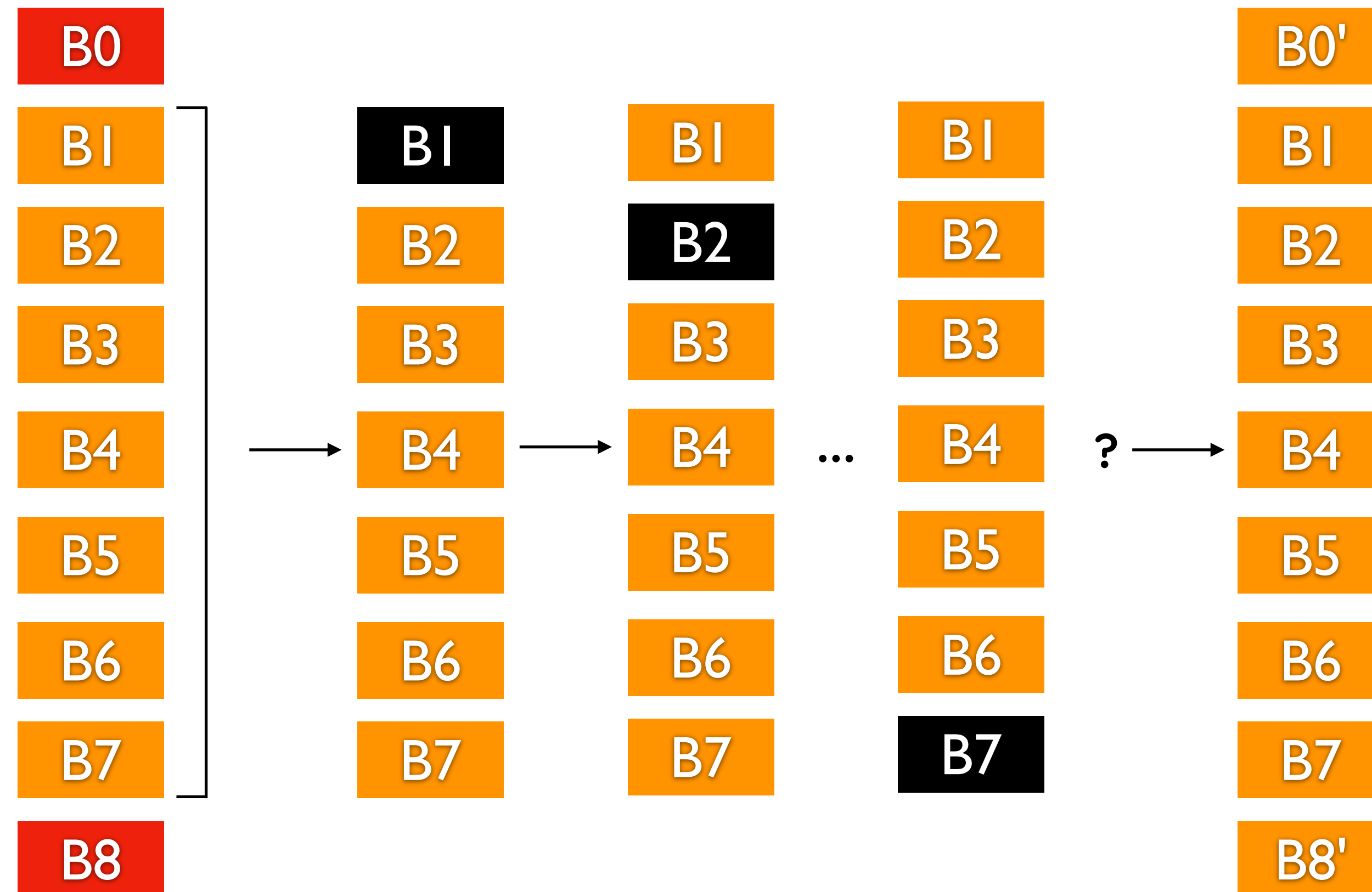
- Java API позволяет получить отдельные блоки
- Сами проверим RS



Восстановление



- RS (6, 3) - не поможет
- Перебор...
- Ищем целую 7-ку
- С из 9 по 7 = 36



Валидация



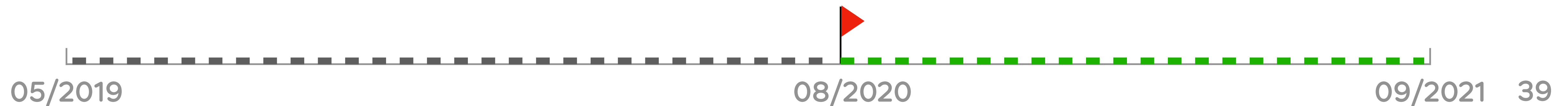
- Структурированные форматы!
 - Parquet - magic numbers, metadata, pages
- Любой формат со сжатием
 - Должен распаковываться без ошибок



Процесс



- Заказ железа на все деньги
- Параллельно восстанавливаем файлы
 - Из копии
 - Или нашим инструментом
- Примерно год...





Финал

В итоге



- Повреждено оказалось больше 4 000 000 файлов
 - 90% имели неповрежденный бэкап
 - 9.9% восстановлены
 - 0.1% (40 000 файлов) - потеряны
 - Примерно 115 TB потерь



Проблемы

- Сколько еще таких - неизвестно
- Исправления в разных версиях

Hadoop HDFS	Bug	Status: All	Assignee: All	erasure
Order by Priority				
	HDFS-15240 Erasure Coding: dirty buffer causes reconstruction block error			
	HDFS-14551 NN throws NPE if downgrade it during rolling upgrade from 3.x to 2.x			
	HDFS-14396 Failed to load image from FSImageFile when downgrade from 3.x to 2.x			
	HDFS-13596 NN restart fails after RollingUpgrade from 2.x to 3.x			
	HDFS-12840 Creating a file with non-default EC policy in a EC zone is not correctly ...			
	HDFS-12686 Erasure coding system policy state is not correctly saved and loaded d...			
	HDFS-12682 ECAdmin -listPolicies will always show SystemErasureCodingPolicies s...			
	HDFS-12480 TestNameNodeMetrics#testTransactionAndCheckpointMetrics Fails in...			
	HDFS-11643 Add shouldReplicate option to create builder			
	HDFS-10850 getEZForPath should NOT throw FNF			
	HDFS-15413 DFSStripedInputStream throws exception when datanodes close idle c...			
	HDFS-15398 EC: hdfs client hangs due to exception during addBlock			
	HDFS-15186 Erasure Coding: Decommission may generate the parity block's conte...			
	HDFS-14847			
				1 2 >





- Бэкапы надо иметь всегда
 - В максимально простой форме, без вычислений
 - В другой системе (физически и алгоритмически)
- Структурированные форматы данных имеют неочевидные плюсы
 - Постоянный контроль корректности!



- Любая достаточно сложная система будет иметь неочевидные ошибки
 - Вне зависимости от возраста и авторитета
- Ее работу нужно проверять извне - постоянно

Ссылки



- Инструмент для работы с поврежденными файлами: <https://github.com/odnoklassniki/hdfs-lifesaver>
- Коды Рида-Соломона
 - https://www.cs.cmu.edu/~guyb/realworld/reedsolomon/reed_solomon_codes.html
- EC в Hadoop
 - <https://issues.apache.org/jira/browse/HDFS-7285>
 - <https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HDFSErasureCoding.html>
 - <https://blog.cloudera.com/introduction-to-hdfs-erasure-coding-in-apache-hadoop/>
- Баги
 - <https://issues.apache.org/jira/browse/HDFS-14768>
 - <https://issues.apache.org/jira/browse/HDFS-15186>



Спасибо за
внимание