

HDFS CRASH TESTS

Кощенко Екатерина

Hadoop

- **Hadoop** используется для надежных, масштабируемых и распределенных вычислений, и применяется как хранилище данных, способное вместить петабайты данных. Состоит из двух ключевых компонентов: **HDFS** и **MapReduce**.
- **HDFS**: Распределенная файловая система **Hadoop**, которая отвечает за хранение данных на кластере **Hadoop**.
- Система **MapReduce**: предназначена для вычислений и обработки больших объемов данных на кластере

HDFS

- Для хранения метаданных используется специальный сервер – сервер имён (NameNode), а данные разбросаны по серверам данных (DataNode).
- Многократно копирует блоки данных и распределяет эти копии по вычислительным узлам кластера, тем самым обеспечивая высокую надежность и скорость вычислений.
- Благодаря репликации обеспечивается устойчивость распределённой системы к отказам отдельных узлов.

Чем хороша HDFS

- Надежность .
- Высокая отказоустойчивость и автоматическое восстановление после отказов. Отказоустойчивость в Hadoop достигается путем перезапуска в других узлах задач, которые выполнялись на отказавших узлах.
- Сервер имён раскрывает для всех желающих расположение блоков данных на машинах (важно для MapReduce)

Цель.

Написать на **Go** систему, которая будет ломать **HDFS**:

- заливать данные с разными свойствами (**replication factor**)
- менять объем данных и их иерархию
- портить данные

Следить за реакцией **HDFS** на все эти действия.

Искать недокументированные сбои.

Что было использовано.

- How to start multi-node cluster:
<http://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-multi-node-cluster/>
- Hdfs Go client: <https://github.com/colinmarc/hdfs>
- Тutorials и статьи по HDFS

Этапы.

- Создать основу для тестов
- Тесты
- Запуск на `single-node cluster`
- Запуск на `multi-node cluster` (две машины)

Типы тестов.

1. Генерация пустых файлов в одной директории.
2. Пустых папок в одной директории.
3. Ветви пустых папок.
4. Файлов с randomным содержимым в одной директории.
5. Рандомная иерархия папок и файлов в них с randomным содержимым.
6. Файлы с randomным содержимым, `replication factor > 1`.
7. Файлы-копии какого-то локального.

Результаты тестов.

1. S/M: Тест до 1 000 000 пустых файлов. Прошел успешно.
2. S/M: Тест до 1 000 000 пустых папок. Прошел успешно.
3. S/M: При попытке сгенерировать ветвь глубины 1000 падает. "Create call failed"
4. Тест до (10 000 S)/(100 000 M) файлов с содержимым не более 1 000 000 символов прошел успешно.
5. Максимальная глубина - 100, максимальный размер папки - (100 S)/(1000 M), максимальный размер файла - 1 000 000 символов. Прошел успешно.
6. M: Тест до 10 000 файлов с rep=2,3 прошел успешно.
7. S: Тест до 10 000 файлов прошел успешно.

Результаты работы.

1. Выполнена первая часть работы: основа для тестовой системы, удаление данных и различные виды генерации.
2. Найдено проблемное место, требующее более подробного рассмотрения.
3. Не были проведены тесты с намеренной поломкой данных, то есть не проверена отказоустойчивость.

**Спасибо за
внимание**