

Построение отказоустойчивых узлов доступа к данным ДЗЗ в системах мониторинга

Балашов И.В., Сычугов И.Г., Прошин А.А.

Институт Космических Исследований РАН, Москва, Россия

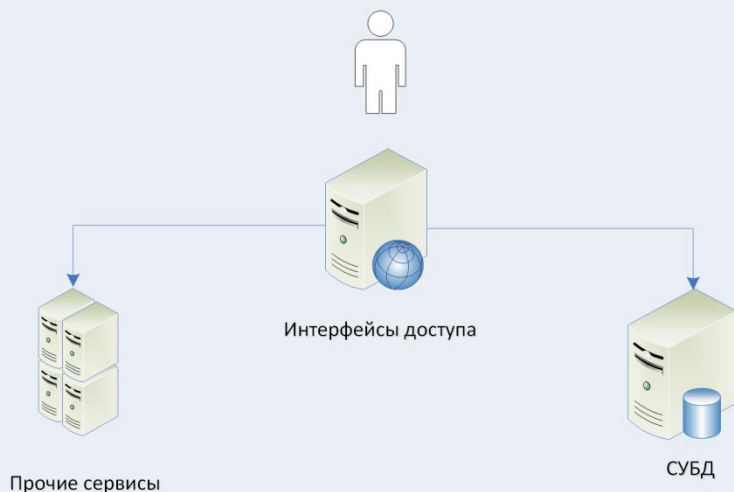
XVIII Всероссийская открытая конференция «Современные проблемы дистанционного зондирования Земли» ИКИ РАН, Москва, 16-20 ноября 2020 г.

Основные нештатные ситуации на вычислительных узлах

Основные нештатные ситуации на вычислительных узлах

- Выход из строя любого из серверов ИС
- Сбои в работе БД (ошибки доступа, потеря части данных)
- Ошибки при запросах к интерфейсам предоставления данных (Код 500, 502, 503, 504,...)
- Перегрузка БД или web-сервера запросами

Схема стандартной ИС предоставления данных



Основные используемые средства обеспечения отказоустойчивости

- Дублирование узлов и их связь с помощью программных компонент Nginx и ProxySQL, чтобы выход из строя одного из них не влиял на получение данных пользователем
- Репликация БД
- Балансировка запросов между узлами для уменьшения нагрузки на каждый

Общая схема построения отказоустойчивой ИС доступа к данным ДЗЗ

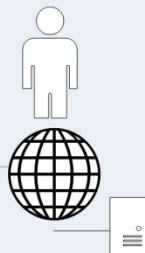
Схема построения отказоустойчивой ИС предоставления данных ЕГХ ПД

NGINX

APACHE
HTTP SERVER PROJECT

ProxySQL

MySQL



http-запрос

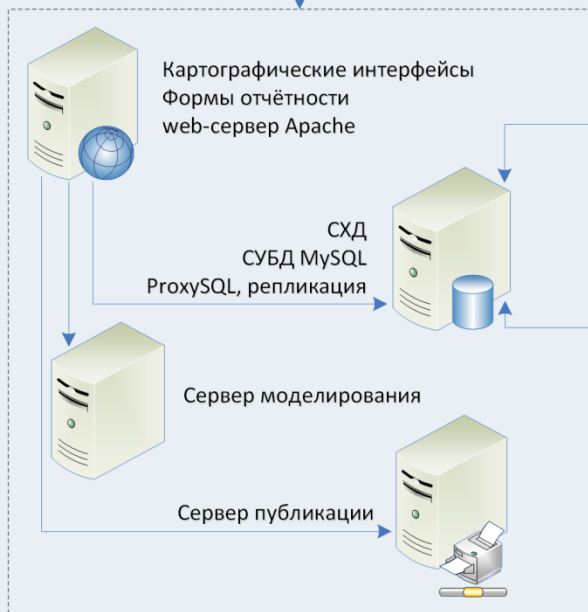
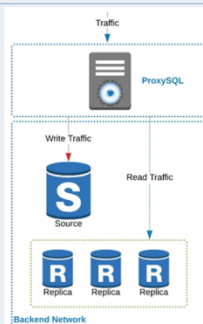
http-запрос

nginx-балансировщик web-сервер. Позволяет распределять трафик на несколько web-серверов с динамической проверкой их доступности
<https://nginx.org/ru/>

MySQL репликация
СУБД MySQL поддерживает репликацию данных – тиражирование изменений с основной БД (master) на зависимую(ые), создавая её копию на случай сбойных ситуаций и резервирования важных данных.

ProxySQL балансировщик БД
Утилита, позволяющая распределять запросы на группу БД, создавая кластер. Также поддерживает разделение запросов на разные группы БД (например, чтение / запись), поддерживает контроль репликации. Динамически проверяет доступность БД

<https://proxysql.com/>



Важные параметры настройки

Основные параметры настройки балансировки web-запросов посредством nginx

```
upstream server_group_name {
    server server1 IP;
    server server2 IP;
}

server {
    listen balancer_port;
    location / {
        proxy_pass http://server_group_name;
        proxy_next_upstream http_500 http_502 http_503 invalid_header error http_504;
        proxy_read_timeout 360s;
    }
}
```

Основные параметры настройки балансировки БД MySQL посредством proxysql

- `mysql_servers` – серверы с БД, находящиеся за балансировщиком

Параметр `hostgroup` используется для объединения серверов в группы, например, группа БД на чтение, группы на запись.

- `mysql_users` – правила доступа для MySQL-пользователей,
- `mysql_query_rules` – правила распределения запросов по разным группам (`hostgroup`).
Посредством правил / свойств разделять запросы на разные группы
- `mysql_replication_hostgroup` – при репликации определяются группы на чтение/запись

Базовая настройка утилиты ProxySQL

Создаём две группы из двух серверов:

hostgroup1 – группа для чтения

hostgroup2 – группа для записи

```
mysql_servers =
```

```
{
    {
        address="X.X.X.X"
        port=3306
        hostgroup=1
        max_connections=700
    },
    {
        address="Y.Y.Y.Y"
        port=3306
        hostgroup=2
        max_connections=700
        max_replication_lag=20
    },
},
```

```
mysql_users:
```

```
{
    {
        username = "user_read"
        password = ""
        default_hostgroup = 1
        active = 1
    },
},
```

Создаём два правила:

Запросы от пользователя на чтение идут в первую группу, на запись – во вторую группу

```
mysql_query_rules:
```

```
{
    {
        rule_id=1
        username="user_read"
        destination_hostgroup=1
        apply=1
    },
    {
        rule_id=2
        username="user_write"
        destination_hostgroup=2
        apply=1
    },
},
```

Пользователь user_read имеет только права на чтение, user_writer – имеет права на запись

```
{
    {
        username = "user_write"
        password = ""
        default_hostgroup = 2
        active = 1
    },
},
```