**BC/NW 2017 № 2 (31):6.1**

**АВТОМАТИЗАЦИЯ ОСМОТРА ЛИНИЙ ЭЛЕКТРОПЕРЕДАЧ**

Сафоненко Д.П Крюков А.Ф

В рамках данной статьи рассмотрено решение проблемы автоматизации осмотра линий электропередач, а также преимущества использование NoSQL баз данных на примере MongoDB и продемонстрирована простота работы. В конце статьи вы узнаете о минусах использования SQL решений на примере MS SQL.

Процесс автоматизации работы, связанной с осмотром линий электропередач включает в себя:

* Сбор данных;
* Контроль за работой персонала;
* Фотофиксацию дефектов линий электропередач;
* Генерацию предварительный ответов;
* Выгрузка данных на компьютер;
* Формирование отчетной документации;

Сбора данных осуществляется на планшетном компьютер без доступа к интернету, поэтому отправлять и обрабатывать эти данные сразу нельзя, так как нет возможности обеспечить бесперебойный доступ к сети. Данные собираются с помощью программы, которую специально разработали на кафедре ВМСС для компании МОЭСК.

Контроль за персоналом обеспечивается на уровне программного обеспечения. Проверка осуществляется по GPS координатам, которые получает планшет в ходе работы и сверяет эти координаты с данными с координатами столбов ЛЭП. Персонал должен пройти всю линию целиком и заполнить данные только в том случае, если он достаточно близко подошел к конкретному столбу ЛЭП. Фотофиксация таблички столба ЛЭП с номером обеспечивает 100% контроль за выполнением работы.

Фотофиксация производится с записью GPS данных, а так же автоматически сортирует фотографии по папкам, что позволяет ускорить работу сотрудников энергетических компаний, которые обслуживают линии. Раньше они выполняли сортировку фотографий вручную, что увеличивало затраты времени на осмотр одной линии.

Во время работы на линии есть возможность генерации предварительного отчета в формате HTML о проделанной работе, что позволяет выполнить осмотр в течении нескольких дней без потерь данных. Что немаловажно, это дает возможность оценить внесенные данные и проверить их на ошибки, и сразу же их исправить перед выгрузкой на компьютер, например, во время дороги от линии до места отдыха бригады.

Выгрузка данных с планшета осуществятся по USB с помощью программы HTML to XLS, которая является второй частью системы, разработанной на кафедре. Программа сама определяет папку с отчетами на съёмном диске (планшете), или папка указывается вручную. Далее в выпадающем списке можно выбрать любой отчет, который есть на планшете и сгенерировать все нужные для официальных файлов отчетности о проделанной работе на линии в формате XLS.

Для обеспечения надежности по желанию можно выгрузить данные в базу данных MongoDB. Для примера ниже приведен весь код, который нужен для выполнения этой операции.

 IMongoDatabase database = client.GetDatabase("CollReport");

 BsonDocument doc = new BsonDocument();

 doc = report.ToBsonDocument();

 var collection = database.GetCollection<BsonDocument>("othet");

 collection.InsertOne(doc);

Report представляет из себя класс со всеми данными из отчета полученного в ходе работе на линии. При этом в случае добавления каких-либо данных в ответ или изменение его структуры в коде программы изменение в данном коде не потребуется. Также не требуется создание, и тем более изменение структуры в базе данных, что сильно ускоряет процесс разработки, так как программисту требуется лишь работать со своим кодом и не думать, как данные будут хранится в базе данных. Все заботы на себя берет база MongoDB. Все, что требуется от программиста, это создание индексов в базе по заданным полям. Пример того как данные хранятся в базе MongoDB:

 "InspectionType" : "Вид осмотра:Плановый",

 "Difectlist" : [

 [

 "1-2",

 "Пролёт: Наброс на проводе: Фаза: Ж."

 ],

 [

 "2;АМ(Анкер металлический)",

 "Опоры: ДКР в теле опоры: Высота 2."

 ],

 "\_id" : "5a0c43182c12c700d4da8e30",

 "InspectionDate" : "18.6.2017",

 "InspectionArea" : "Район(участок):Бутырская РМС",

 "InspectionFromTo" : "Осмотр проведен от опоры №1 до опоры №7",

 "Receiving" : "Листок осмотра принял:сафоненко/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/",

 "Name" : "Наименование:Вернадского- Воздушная линия Uном=110 кВ ",

 "Company" : "Предприятие:Московские высоковольтные сети"

Из минусов можно отметить, что формат не слишком человекочитаемый, но зато очень удобный для хранения и работы с ним в пользовательском приложении, так как поддерживает конвертацию в объекты классов.

 Перейдем к прямому сравнению MongoDB и MSSQL. Начнем с MondoDB и его плюсов и минусов. К плюсам можно отнести:

* Автоматическая масштабируемость по средствам шардинга данных. В терминах SQL баз данных можно сказать что результаты JOIN для конкретного отчета будут находится на одном сервере, а не собираться с разных серверов.
* Легкость для программистов, так как не требуется разработка схемы хранения для каждого класса (например, отчета). Схемы хранения создадутся автоматически;
* Возможность простого построения сложных агрегация данных;
* Простая работа с данными геолокации;
* Локальность данных (Более быстрый доступ к данным);
* Динамические схемы данных;
* Бесплатное использование без каких-либо ограничений;
* Полная совместимость с JSON;

Из минусов можно отметить следующее:

* Новый язык запросов отличающийся от SQL, что требует привыкания.

Пример:

|  |  |
| --- | --- |
| MySQL | MongoDB |
| INSERT INTO users (user\_id, age, status)VALUES ('bcd001', 45, 'A') | db.users.insert({ user\_id: 'bcd001', age: 45, status: 'A'}) |
| SELECT \* FROM users | db.users.find() |
| UPDATE users SET status = 'C'WHERE age > 25 | db.users.update( { age: { $gt: 25 } }, { $set: { status: 'C' } }, { multi: true }) |

* Относительная современность технологии;
* Более высокая требовательность к оперативно памяти. А точнее к ее объёмам;

К плюсам MSSQL, если сравнивать напрямую с MongoDB, можно отнести:

* Большое количество специалистов в этой области;
* SQL язык запросов;
* Большой выбор аналогов с простым способом переноса в обе стороны;
* Проверенная временем реляционная модель базы данных;

К недостаткам MSSQL можно отнести:

* Отсутствие динамической схемы данных;
* Более сложные и затратные способы масштабируемости;
* Более сложное администрирование;
* Стоимость;
* Производительность;

**Заключение**

Подводя итоги можно сказать следующее. NoSQL базы данных и, в частности, MongoDB, обеспечивают много преимуществ в отношении не только MSSQL, но и вообще всех баз данных на основе реляционной модели. При том, что технология относительно новая, она позволяет упростить работу программиста с сохранением всех плюсов старых технологий. NoSQL требует длительного переучивания старых специалистов и переносе логики и данных, что потребует определенное количество затрат и времени. Однако, оно полностью себя оправдывает.