

ИСПОЛЬЗОВАНИЕ АЛГОРИТМА ОБРАБОТКИ НАВИГАЦИОННЫХ ДАННЫХ ДЛЯ МИНИМИЗАЦИИ НАГРУЗКИ БАЗЫ ДАННЫХ

А.О. Игумнов, Л.О. Игумнов
Томский политехнический университет
loi1@tpu.ru

На 2018 год, все существующие системы диспетчерского управления и мониторинга транспорта оборудованы технологией контроля местоположения автомобилей. Каждый автомобиль, находящийся в области применения системы управления, включает в себя оборудование в виде датчика положения и передатчика координат. Существует множество систем позиционирования: системы позиционирования с использованием пассивных радиочастотных идентификаторов (RFID); системы позиционирования с использованием активных RFID; WiFi позиционирование; позиционирование в сотовых сетях; спутниковая система навигации.

Благодаря неограниченной площади применения, а также высокой точности позиционирования, наибольшую актуальность в определении координат транспортного средства имеют датчики спутниковой навигации GPS/ГЛОНАСС. Спутниковая связь обеспечивает непрерывную передачу данных о положении транспортного средства и передает данные подключенному в системе управления приемному оборудованию. Интервал получения данных приемником по протоколу NMEA варьируется от 1 до 15 раз в секунду, в зависимости от полноты получаемых навигационных данных. Для того, чтобы понизить временной интервал получения данных, приемники подключаются к терминалу, который обеспечивает прореживание данных перед посылкой. Технология прореживания данных может осуществляться либо по времени получения, либо по непосредственному положению транспортного средства, но стоит заметить, что для системы мониторинга положения нецелесообразно отправлять данные координат реже, чем раз в минуту. При большом количестве транспортных средств, подключенных к системе мониторинга, сервер будет принимать информацию о координатах местоположения несколько раз в секунду. Непрерывное сохранение всех точек координат требует повышенной вычислительной мощности и оказывает повышенную нагрузку на сервер. Такая нагрузка может приводить к длительным блокировкам базы данных системы управления и нуждается в разработке алгоритма, который позволил бы снизить нагрузку на базу данных.

Обычный алгоритм сохранения навигационных данных, можно представить в виде следующей последовательности:

- передатчик, оборудованный в автомобиле посылает данные о положении на сервер;
- сервер, используя эти данные, формирует объект точки для сохранения в базе;

- ORM формирует запрос и сохраняет данные в базу;

Для снижения нагрузки на базу данных необходимо обеспечить прореживание транзакций, обращенных к базе данных. Такое прореживание можно реализовать, введя дополнительный модуль обеспечивающий обработку данных на сервере и, после преобразования данных в объект для сохранения, будет обеспечиваться передача данных в дополнительный модуль обработки вместо передачи в ORM.

Дополнительный модуль содержит буфер точек, в который поступают все передаваемые прибором автомобиля точки координат. Все поступающие точки кладутся в буфер, а не сохраняются немедленно. Для сохранения данных запускается отдельный поток, который с заданным временным интервалом осуществляет изъятие всех точек из буфера, тем самым очищая его. Буфер освобождается для последующего заполнения, а точки из него отправляются на сохранение в ORM. Таким образом, сохранение точек координат положения транспортного средства будет вызываться не чаще чем в определенный заданный временной интервал.

Интервал сохранения точек следует подбирать исходя из таких критериев как важность и актуальность получаемых навигационных данных и может варьироваться от 1 до 3 секунд. Например, для осуществления мониторинга специализированного городского транспорта наиболее важное значение имеет общее соблюдение конкретным транспортным средством своего маршрута, нежели наиболее актуальное положение транспорта в момент времени. В случае же систем мониторинга пассажирского городского транспорта решающее значение имеет положение транспортного средства и как следствие важно постоянное обновление данных по мере поступления точек.

Важно учитывать, что все сохраняемые объекты точек преобразуются в SQL запрос и в зависимости от вида используемой СУБД нужно отдельно отслеживать заполнение буфера в соответствии с введенными ограничениями конкретной СУБД на одновременное внесение данных. Отсюда следует, что по мере поступления данных на сервер, должна осуществляться проверка буфера на его наполненность. Если в результате проверки, в буфере будет находиться установленное ограничение СУБД количество записей, таймер сохранения будет обнуляться и накопленные объекты точек будут передаваться в ORM для последующего сохранения [2].

Основным преимуществом в использовании дополнительного буфера обработки с применением

отложенного сохранения является возможность избежать полной блокировки таблицы навигационных данных и снизить общую нагрузку на сервер базы данных. Также, введение дополнительного буфера позволяет реализовывать в работе системы управления мониторингом вспомогательный механизм программных транзакций [3,4]. Дополнительно, может применяться модуль взаимодействия с буфером точек, для отслеживания последнего актуального положения все транспортных средств.

При проектировании крупномасштабных распределённых систем баз данных, также используется распределённая система управления базой данных. За запись данных в СУБД отвечают отдельные программные компоненты, которые масштабируются на некоторое количество аппаратного обеспечения. Вспомогательный механизм программных транзакций оборачивает сохранение получаемых точек в собственную транзакцию, которая при срабатывании условий отложенного сохранения выделяет еще одну транзакцию, направленную на другую, по аппаратному положению базы данных.

По мере увеличения отслеживаемых в системе управления транспортных средств, увеличивается и количество принимаемых сервером навигационных точек, поступающих за единицу времени - g . Отдельное сохранение каждой полученной точки приводит к возникновению временной блокировки базы данных, что в большинстве случаев затрудняет работу системы управления мониторингом. Чем больше данных в виде точек поступает на сервер, тем существеннее становятся задержки для следующих данных - t . При учете интервала сохранения - i и лимита единовременной записи в базе данных - I , модуль отложенного сохранения точек позволит снизить общее время блокировки БД от $(g*t)/i$ до $((g/I)*t)$ раз. Помимо прочего, встроенный вспомогательный модуль транзакций, будет способствовать сокращению временных издержек при

значениях g превышающих допустимые нормы для одного сервера СУБД.

Введение дополнительного модуля в алгоритм сохранения навигационных данных позволит существенно снизить нагрузку на базу данных, при работе с большим количеством поступающих навигационных данных. Также сохранение нескольких навигационных точек единовременно положительно скажется на снижении нагрузки на клиентском ПО при отображении местоположения транспорта на карте, благодаря снижению количества оповещений о смене положения транспортного средства.

Список использованных источников

- 1) Feng T., Wu Q., Zhang H. Key parameters generation of the navigation data of GPS Simulator //arXiv preprint arXiv:1402.2056. – 2014.
- 2) Sridhar K. T., Sakkeer M. A. Optimizing Database Load and Extract for Big Data Era //Database Systems for Advanced Applications. – Springer International Publishing, 2014. – С. 503-512.
- 3) Cheung A. et al. Using Program Analysis to Improve Database Applications //IEEE Data Eng. Bull. – 2014. – Т. 37. – №. 1. – С. 48-59.
- 4) Manegold S., Boncz P. A., Kersten M. L. Optimizing database architecture for the new bottleneck: memory access //The VLDB Journal—The International Journal on Very Large Data Bases. – 2000. – Т. 9. – №. 3. – С. 231-246.
- 5) Cheung A., Solar-Lezama A., Madden S. Optimizing database-backed applications with query synthesis //ACM SIGPLAN Notices. – 2013. – Т. 48. – №. 6. – С. 3-14.