

УДК 004.75

РАЗРАБОТКА АЛГОРИТМА ОФФЛАЙНОВОЙ ПОДГОТОВКИ ДАННЫХ ДЛЯ ВЫСОКОНАГРУЖЕННОГО ПРИЛОЖЕНИЯ

Кустарева А. Д., Университет ИТМО

**Научный руководитель – д.т.н., профессор Шалыто А. А.
Университет ИТМО**

Высоконагруженные системы на данный момент являются одним из основных трендов в мире разработки программного обеспечения. В рамках данной работы рассматривается компонент с высоким *grps*, который занимается расчетом стоимости доставки товаров для пользователей. В работе описано решение, которое позволяет сделать предварительную подготовку данных с целью уменьшения числа ресурсов, затрачиваемых компонентом на обработку запросов.

Введение. В настоящий момент всё возрастающие объемы данных и запросы пользователей приводят к тому, что обычные приложения, не рассчитанные на высокую нагрузку, не могут гарантировать стабильную работу. В связи с этим есть два направления возможных действий, которые призваны решить данную проблему: первое — усложнение инфраструктуры дата центров и/или использование передового оборудования для обеспечения необходимого уровня отказоустойчивости, что дорого, второе — оптимизация подходов к разработке приложений и обеспечение требований по консистентности и отказоустойчивости на уровне кода, что сложнее, однако не требует таких материальных затрат, как первый вариант. В данной работе будет рассматриваться одна из возможных оптимизаций на уровне кода, а именно оффлайновая подготовка данных для высоконагруженного приложения.

Основная часть. Оффлайновая подготовка данных подразумевает под собой создание индекса, который готовится сторонним компонентом и в последствии раскладывается под компонент с высоким *grps*. Подготовка индекса (сам процесс называется варкой индекса) происходит раз в фиксированное время. В данном случае индекс рассматривается как полная информация о доставке, содержащая на определенный момент времени полный корректный срез всей имеющейся информации (которая является входными данными для данного компонента и приходит из разных внешних источников в разные моменты времени в разных форматах). Варка индекса позволяет собрать данную информацию и представить ее в максимально удобном для дальнейшего использования виде: сформировать легковесные поисковые сетки, которые будут использоваться для быстрого поиска подходящих опций доставки по указанным параметрам и жить в памяти высоконагруженного компонента для ускорения работы, а информацию о деталях положить отдельно в хранилище, из которого она будет доставляться потребителям отдельно. Для ускорения обновления информации индекс варится не единым монолитным куском, а по мере обновления данных. Указанные кусочные обновления происходят в рамках так называемых поколений: собирается информация, которую необходимо обновить, происходит ее подготовка, формируется новое поколение и отправляется под компонент с высоким *grps* для раскладки. Таким образом, индекс есть поочередное наложение всех поколений, сваренных на определенный момент времени. Поколения также содержат мета информацию о сваренных данных, необходимую для очистки ставших неактуальными поколений. Взаимодействие между компонентом, занимающимся варкой данных, и компонентом с высоким *grps* строится на основе очереди событий. Указанная очередь организована на основе БД. А именно, в БД записывается мета информация о новом сваренном поколении, записи присваивается монотонно возрастающий идентификатор. Читающий компонент поддерживает у себя информацию о том, какое сообщение было прочитано последним, и на основе этой информации понимает, есть ли в БД обновления.

Данный механизм позволяет восстановить и в полном объеме заново разложить корректный индекс после рестарта компонента. Также можно заметить, что индекс после каждого поколения содержит полную корректную информацию о доставке. Таким образом, в случае поломки обновления данных или варки, компонент, отвечающий за расчет доставки, продолжит работать штатно и на стабильность его работы данные проблемы не повлияют.

Выводы. В рамках данной работы был разработан и реализован подход к снижению ресурсозатрат в высоконагруженном компоненте, а именно способ офлайновой подготовки данных для высоконагруженного приложения. При этом гарантируется консистентность подготовленных данных и отказоустойчивость всей системы.