

## БИБЛИОТЕКА C++ ШАБЛОНОВ ДЛЯ СРАВНЕНИЯ И СЛИЯНИЯ STL КОЛЛЕКЦИЙ В ПРИЛОЖЕНИЯХ ОПТИМИСТИЧЕСКОЙ РЕПЛИКАЦИИ ДАННЫХ

Семенов Г.В. (Университет ИТМО)

Научный руководитель – д.ф.-м.н., проф. Семенов В.А.

(Институт системного программирования им. В.П. Иванникова РАН)

Обсуждаются проблемы применения технологий оптимистической репликации для семантически сложных моделей данных. Предлагается и описывается библиотека шаблонов на языке C++ для реализации операций автоматического сравнения и слияния коллекций стандартной библиотеки STL. Применение библиотеки иллюстрируется на примере разработанного приложения адресной книги с функцией синхронизации реплик книги.

**Введение.** Технологии оптимистической репликации находят широкое распространение в распределенных информационных системах, к которым предъявляются повышенные требования доступности данных. Примерами успешного применения технологий являются нереляционные базы данных NoSQL, сервера DNS, системы контроля версий, системы комплексной инженерии, MMO-игры. В отличие от пессимистических политик управления транзакциями на основе блокировок данных, оптимистические политики предусматривают создание реплик данных для каждого клиента (вычислительного узла или приложения) и предоставление к ним неограниченного доступа в оптимистическом предположении, что расходящиеся реплики могут быть приведены к единому согласованному представлению, а возможные конфликты при слиянии могут быть разрешены.

Отказ от транзакционных требований ACID (Atomicity, Consistency, Isolation, Durability) в пользу принципов BASE (Basically Available, Soft state, Eventually consistent), и в частности, переход от постоянной согласованности к согласованности данных в конечном счете позволяют улучшить доступность данных (Availability). Это согласуется с известной CAP-теоремой, утверждением которой является невозможность проектирования распределенной информационной системы, в которой одновременно бы обеспечивалась согласованность данных, доступность и устойчивость к разделению (Partition Tolerance).

Вместе с тем, серьезным препятствием в применении оптимистического подхода является необходимость идентификации и разрешения конфликтов между расходящимися репликами таким образом, чтобы результирующее согласованное представление объединяло в себе максимальное число изменений и при этом удовлетворяло всем семантическим ограничениям, предписанных моделью данных. Проблема существенно упрощается при использовании структур данных с коммутативными операциями (так называемых, CRDT-типов). Однако для более значимых для практики случаев использования семантически сложных моделей, например, моделей с алгебраическими ограничениями, формально специфицированными на декларативных языках информационного моделирования EXPRESS, UML/OCL, OWL, SQL, проблема становится крайне сложной и требует новых системных подходов к решению.

Разработка программных приложений оптимистической репликации также вызывает большие трудности, поскольку в реализации необходимых функций сравнения и слияния реплик (или более точно, функций нахождения и объединения разностных преобразований diff & merge) также должны учитываться разнообразные семантические ограничения модели и порождаемые ими многочисленные зависимости между данными.

**Основная часть.** В работе предлагается подход к разработке приложений оптимистической репликации на языке C++ в предположении, что модель данных представляется коллекциями стандартных видов `set`, `multiset`, `list`, `vector`, `map` и реализуется в целевом или унаследованном приложении на основе стандартной библиотеки STL. Подход предполагает применение шаблонных реализаций для операций сравнения и слияния коллекций в рамках трехсторонней схемы (3-way merge) в соответствии со следующими определениями:

*Пусть из исходной версии  $A$  оптимистически была получена реплика  $B$ , реплики  $A$  и  $B$  полиморфны одному типу  $T$ , на котором определено множество операций  $\phi(T)$ . Тогда  $\Delta$ -преобразованием из  $A$  в  $B$  на типе  $T$  называется минимальная последовательность операций, при применении которой к  $A$  получается  $B$ .*

*Слиянием двух реплик  $B$  и  $C$ , полученных оптимистически из  $A$ , одинакового типа  $T$  называется действительное  $\Delta$ -преобразование над  $A$ , в последовательность операций которого в качестве подпоследовательностей входят как  $\Delta(A, B)$ , так и  $\Delta(A, C)$ , при этом такая последовательность содержит максимально возможное число неконфликтных операций и, будучи примененной к  $A$ , порождает новую реплику  $\Omega(A, B, C)$  типа  $T$ .*

Разработанная библиотека шаблонов предоставляет готовые к использованию реализации  $\Delta$ -преобразований, в дальнейшем называемые дельтами. Дельты представляются как композиции или последовательности операций, которые могут быть вычислены по заданной паре реплик коллекции (например,  $\text{diff}(A, B) \rightarrow \Delta(A, B)$ ), применены для обновления реплики коллекции ( $\text{patch}(A, \Delta(A, B)) \rightarrow B$ ), инвертированы для реализации обратного преобразования ( $\text{invert}(\Delta(A, B)) \rightarrow \Delta(B, A)$ ), объединены ( $\text{compose}(\Delta(A, B), \Delta(B, C)) \rightarrow \Delta(A, C)$ ) или слиты ( $\text{merge}(\Delta(A, B), \Delta(A, C)) \rightarrow \Delta(A, \Omega(A, B, C))$ ).

Дельты представляются в виде типизированных объектов и конструируются в зависимости от вида коллекций и типа элементов, для работы с которыми они предназначены. Тип дельты предопределяет набор допустимых операций и их сигнатуры. Для неупорядоченных коллекций это операции добавления, удаления и модификации элементов соответствующего типа. Для упорядоченных коллекций это операции вставки, удаления и замены элементов в заданных позициях. Тип дельты также предопределяет набор алгоритмов, который может быть использован для сравнения и слияния коллекций. Например, для вычисления  $\Delta$ -преобразований для упорядоченных коллекций применяются реализованные в библиотеке версии алгоритмов Вагнера-Фишера, Майерса и Ханта-Шиманского.

Предусмотренные способы параметризация шаблонов обеспечивают их корректное использование для коллекций элементов примитивных и конструируемых типов языка C++, вложенных коллекций, коллекций объектов и т.п.

**Выводы.** Таким образом, предложен подход к разработке приложений оптимистической репликации на языке C++ на основе библиотеки шаблонов для операций сравнения и слияния коллекций. Подход успешно апробирован при создании программного приложения адресной книги с функцией синхронизации реплик книги и представляется перспективным для разработки других классов приложений. Направлениями дальнейших исследований будут развитие подхода для произвольных объектно-ориентированных моделей данных, а также анализ способов автоматизации процессов разработки программных приложений оптимистической репликации.