УДК 004.422.635.33

И. С. Колосова – магистрант кафедры вычислительных систем и сетей
А. В. Аксенов (ст. преп) – научный руководитель

Разработка и анализ способов вращения древовидных структур данных с приоритетами элементов

В современном мире ежедневно через человека проходит огромное количество информации. Малая часть ее необходима и важна, но остальное скорее всего больше никогда не будет использоваться. Расстановка приоритетов распределяет данные по своим местам. Чем больше приоритет, тем весомей факт, и тем быстрее должен быть доступ к ней.

Представим всё это множество знаний в виде дерева поиска со взвешенными элементами и попробуем сократить путь до самых значимых узлов.

Для этого будем использовать принципы малых и больших вращений.

Малое вращение затрагивает лишь два уровня дерева, но сильнее изменяет соотношение весов поддеревьев, чем большое вращение, использующее три уровня. Выбранный критерий вращения - сбалансированность дерева, то есть модуль разности суммы весов в поддеревьях.

На основе этих принципов, были разработаны следующие алгоритмы балансировки:

TKOL (The King Of Limbs) - алгоритм спускается по дереву в поисках элемента, а после добавления элемента или увеличения приоритета существующего, поднимается к корню, проверяя сбалансированность поддеревьев и при необходимости совершая малые вращения.

BTKOL (Big TKOL) - алгоритм аналогичен предыдущему, но используются не только малые, но и большие вращения. В связи с этим требуется больше времени на вычисление критерия, так как делается выбор между тремя состояниями: не выполнять вращение, выполнить малое вращение или выполнить большое вращение.

DSTKOL (Down Small TKOL) - этот алгоритм также спускается по дереву в поисках элемента, однако вращения выполняются только на спуске. Этот способ требует всего один проход по дереву, что повышает его эффективность, однако есть свои жертвы - при таком варианте сложно и неэффективно использовать большие вращения.

NATKOL (No Ask TKOL) - этот способ экономит на другом - вычислении критерия. При увеличении приоритета элемент поднимается на уровень вверх, то есть происходит малое вращение.

Для сравнения вышеперечисленных способов между собой использовались следующие критерии:

AWD(Average Weight Depth) - средневзвешенный путь до узла. Чем меньше значение AWD, тем меньше средний путь до произвольного узла от корня, следовательно, время доступа к этому узлу сокращается.

Суммарный проход по дереву - количество узлов, пройденных алгоритмов за время построения и балансировки дерева. Стоит отметить, что учитываются узлы только при спуске, а это значит, что в TKOL и BTKOL эту величину следует удвоить, так как балансировка происходит при подъеме.

Количество вращений также является немаловажным критерием. Каждое вращение занимает время. Идеально сбалансированное дерево, которое было достигнуто при огромном количестве вращений требует много времени для построения, следовательнобудет неэффективным.

Средний баланс показывает среднее значение отношения весов поддеревьев во всем дереве.

Для каждого поддерева высчитывается отношение меньшего веса поддерева к большему, суммируется и делится на количество поддеревьев.

На основе проведенных экспериментов, при выполнении которых, на вход подавались тексты различных содержания и длины, была составлена таблица.

Критерий	**Наилучший показатель**	**Наихудший показатель**
AWD	BTKOL	NATKOL
Суммарный проход по дереву	DSTKOL	BTKOL
Количество вращений	TKOL/ DSTKOL	NATKOL
Средний баланс	NATKOL	BTKOL

Суммарных проход по дереву и количество вращений в первую очередь влияют на скорость построения

дерева. AWD и средний баланс указывают на скорость обращения к элементам в уже построенном дереве.

Исходя из данных таблицы получаем, что DSTKOL требует меньше времени на построение по двум параметрам, однако не имеет лучших показателей скорости обращения к элементам. NATKOL и BTKOL требуют больше всего времени при построении, а также проигрывают по одному из критериев скорости обращения. Это значит, что BTKOL поднимает выше к корню узлы с большим приоритетом, однако узлы с меньшим приоритетом находятся в беспорядке. И наоборот, NATKOL балансирует дерево на всех уровнях, однако не поднимает "тяжелые" элементы достаточно близко к корню.

DSTKOL по своим параметрам мало чем отличается от TKOL, однако имеет одно большое преимущество - для поиска и балансировки требуется только один проход по дереву, что вдвое уменьшает затраты времени и памяти на его построение.

Таким образом, получаем, что алгоритм построения и балансировки двоичных деревьев поиска со взвешенными элементами DSTKOL является наиболее эффективным из разработанных.

Библиографический список

1 Колосова И.С. Разработка и анализ критерия вращения в древовидной структуре данных с приоритетами элементов, Сборник докладов Научной сессии ГУАП, посвященной Всемирному дню космонавтики, СПб, ГУАП, 2014 г.

2 Аксенов А.В., Колосова И.С. Анализ эффективности балансируемых структур данных с приоритетами элементов**,** Сборник докладов Научной сессии ГУАП, посвященной Всемирному дню космонавтики, СПб, ГУАП, 2013 г.