УДК 622

ПРОВЕРКА XML ФАЙЛОВ БОЛЬШОГО РАЗМЕРА

М.Д. Кравцов, студент гр. ПИб-111, IV курс
Научный руководитель: С.А. Веревкин, старший преподаватель
Кузбасский государственный технический университет имени Т.Ф. Горбачева
г. Кемерово

В наши дни информация играет важную роль не только в жизнедеятельности, но и в работе программного обеспечения. XML-разметка – один из используемых расширяемых языков разметки, для обмена данными между ПО. Документ, описывающий данные с помощью XML-разметки, с легкостью интерпретируется и информацию из него можно использовать в ПО. Однако не все так просто. Взятие информации из такого документа усложняется, если его размер становится большим. Времени, на взятие информации из такого документа, уйдет много.

Помимо времени, потраченное на загрузку информации из xml-файла, может случиться так, что разметка документа будет нарушена и получить информацию будет невозможно. Чтобы избежать такой ситуации существуют методы проверки таких файлов. После проверки файла на целостность разметки, можно быть уверенным в том, что информация будет извлечена.

Подведем некоторые итоги:

- Большой размер xml-файла – затруднит работу с ним.
- Проверка файла даст уверенность в том, что информация из него будет извлечена.

Стоит упомянуть то, что при проверке, документ полностью загружается в память и после этого с ним с ним можно работать. Проверка на целостность - не исключение. Таким образом, мы не сможем проверить документ, если его размер крайне большей. К сожалению, стандартными средствами здесь не обойтись. Появляется необходимость извлекать информацию из документа небольшими частями и работать с ними.

Подход, который получает часть информации из документа решает проблему с нехваткой памяти, но проблема с временем обработки такого документа остается открытой. Время, конечно, которое необходимо на интерпретацию документа, зависит от характеристик компьютера, однако этот процесс займет продолжительное время.

Уменьшить время на обработку xml документа можно разбив документ на блоки и производить проверку этих блоков параллельно, в разных потоках. Для этого нам необходимо проиндексировать элементы документа. Делается это просто – построчно подгружаем документ в память и каждому элементу корневого элемента присваиваем индекс. Затем начинаем процесс разбиения на блоки. Допустим, каждому блоку принадлежат по 100

проиндексированных элементов. Теперь остается дело за малым: открываем новые потоки и загружаем в них данных соответствующих блоков, которые были определены ранее. Отслеживаем работу потоков и следим за результатами работы потоков. Таким образом мы решим проблему, связанную со временем обработки документа, при правильном определении количества потоков максимально используем возможности компьютера.