XTF - МАСШТАБИРУЕМЫЙ ФОРМАТ ХРАНЕНИЯ ТРАССИРОВОЧНОЙ ИНФОРМАЦИИ

Д.В. Ежов, А.В. Ершов, А.Д. Модяев, А.В. Огородников

Современные параллельные программные комплексы, созданные с использованием MPI, становятся все сложнее; они используют все большее количество вычислительных ядер, а, следовательно, и MPI процессов. Инструментальные средства оптимизации и отладки MPI приложений обычно накапливают информацию в виде файлов трасс, объем которых растет с увеличением размера приложения, и может достигать десятков и сотен гигабайт. Трассы такого объема трудно анализировать с применением существующих средств, так как время ожидания результата при обработке и анализе становится неприемлемо большим. Основные причины этого следующие:

- последовательный характер организации контейнера данных трассы. Файлы трассы состоят из последовательностей записей переменной длины и формата, что делает время доступа к отдельной записи неприемлемо большим — O(N) для трассы длиной N записей;
- последовательный характер методик обработки и визуализации данных, ориентированных на чтение, обработку и изображение в графическом представлении всех данных в некотором интервале трассы (в пределе — всей трассы целиком). Время ожидания и здесь оказывается не лучше, чем O(N).

Таким образом, для улучшения масштабируемости инструментальных средств отладки и оптимизации MPI приложений необходимо ослабить зависимость времени обработки трассы от объема данных.

Анонсируемый формат трассы XTF (eXtended Trace Format) решает эту проблему путем изменения способа организации контейнера, при этом время доступа к отдельной записи в трассе оказывается порядка O(log(N)), то есть слабо зависит от роста объема трассы. Основой формата является кешируемый странично-структурированный контейнер данных с индексно-последовательной организацией, с прямым доступом к индивидуальной записи, использующий для формирования индексов технологию сильно ветвящихся деревьев (варианты B-tree).

Пространство файла распределяется блоками фиксированного размера (страницами). Каждая страница содержит служебный заголовок, и массив записей одного типа, формата и размера. Массив однотипных записей в файле может занимать несколько страниц, не обязательно расположенных последовательно. Таким образом, страничное структурирование дает возможность хранить в файле разнотипные массивы данных, пополняемые независимо друг от друга и без предварительной разметки файлового пространства.

Для доступа к данным файла используется техника неявного ввода-вывода в форме программно реализованного страничного кеша, оснащенного сборщиком мусора с дисциплиной вытеснения страниц типа LRU (Least Recent Used). При такой дисциплине часто используемые страницы будут находиться в памяти (резидентны) постоянно, или почти постоянно. Для подкачки страниц в кеш могут использоваться как обычные блочные операции ввода-вывода, так и техника отображения файла на память, поддерживаемая современными операционными системами (mmap).

Для обеспечения прямого доступа к данным каждый массив имеет индексные таблицы (индексы):

- индекс для доступа по номеру процесса (ключ P);
- индекс для доступа по времени (временной отметке) записи (ключ T);
- индекс для доступа по номеру записи (ключ N).

Упорядочение данных по времени и номеру в пределах каждого массива достигается автоматически, вследствие естественного направления течения времени при выполнении трассировки. Упорядочения, вводимые отношениями сравнения для ключей T и N, для каждого массива совпадают; таким образом, достигается объединение в один общий индекс TN, с возможностью доступа по любому из ключей. При этом второй ключ может при поиске рассматриваться как часть данных, что дает возможность получать эффективное отображение значений ключей T и N друг в друга.

Индекс TN организован в форме разновидности сильно ветвящегося дерева, известной как B+tree. Дополнительно, страницы дерева одного уровня соединены ссылками в двусвязный список, что позволяет легко итерировать по данным в любом направлении. Индексные деревья имеют следующие свойства:

- узлами дерева являются страницы двух видов
  - индексные: записи состоят из ключей и ссылки на страницу-потомка;
  - листья: записи состоят из ключей и собственно данных.
- высота дерева H одинакова для любого листа данного дерева;
- все страницы имеют 100% заполнение, кроме страниц «правого края» дерева (с наибольшими значениями ключей).

Последнее свойство для функционирования индексов не обязательно, но возникает автоматически вследствие естественного направления течения времени при трассировке.

Для дерева с длиной массива (числом записей в листьях) N и числом записей на индексной странице S и листе Z оценкой высоты дерева будет $H = \log_S(N/Z) \sim= O(\log_S N)$. При поиске заданного ключа потребуется не более H затратных операций подкачки страниц (если не резидентны в кеше), и некоторого количества значительно менее дорогостоящих операций извлечения записи и сравнения ключей для H страниц в памяти.

Например, при использовании бинарного поиска ключа на странице, общая оценка числа сравнений ключей будет $C = O(\log_S N * \log_2 S) = O(\log_2 N)$.

Таким образом, время доступа к записи по заданному значению ключа имеет логарифмический характер роста при увеличении объема трассы. Время ожидания определяется, в основном, числом затратных операций подкачки нерезидентных страниц, не превосходящее высоты дерева.

Рассмотрим реалистичный пример: массив данных размером 100Gb при размере страницы 4Kb, размере индексной записи 24B, записи данных 64B. Тогда N=1677721600, S=170, Z=64, H=5. Число страниц в дереве, по уровням, начиная от корня: 1, 6, 908, 154203, 26214400. Объем индекса составляет 155118 страниц, то есть приблизительно 0.6% от объема данных. Поиск по ключу любой записи данных посетит цепочку из 5 страниц и выполнит не более 31 сравнения ключей в памяти. При частых обращениях корневая страница индекса с высокой вероятностью окажется резидентной в кэше (возможно, и некоторые страницы 1-го уровня). Таким образом, в среднем потребуется выполнить лишь 3-4 подкачки на один поиск записи.

Использование ключевого доступа к записям в индексно-последовательном контейнере дает возможность реализовать эффективные базовые примитивы доступа к массивам записей и навигации по ним:

- search(key): поиск записи по указанному ключу (ключам), вырабатывающий ссылку на страницу и номер записи на странице, либо признак «не найден»;
- size(key_1,key_2): определение числа записей массива, расположенных в листьях в интервале трассы, ограниченном указанными значениями ключей. Для примитива не требуется затратное итерирование по всему интервалу данных, достаточно выполнить восхождение вверх по двум цепочкам страниц дерева от страниц, содержащих ключи key_1 и key_2, к общей для данных ключей странице, являющейся корнем поддерева, ограниченного этими ключами;
- next(key,step): перемещение по массиву от записи с указанным значением ключа, к записи, отстоящей от нее на «step» записей вперед или назад по массиву. Реализация тривиальна, если шаг «step» не выводит за пределы одной страницы; в противном случае потребуется восхождение по дереву вверх, с последующим спуском обратно на тот же уровень дерева, но уже по другой цепочке страниц. Как и для примитива size(key_1,key_2), затратное итерирование не требуется.

На основе данного формата могут быть разработаны принципиально более эффективные методики доступа и обработки, анализа и визуализации данных трассы, и интерактивной навигации по графическому представлению, опирающиеся на особенности нового формата, и имеющие также логарифмический характер роста времени ожидания от размера трассы:

- быстрое определение числа записей в произвольном интервале трассы без необходимости полного сканирования данных интервала дает возможность получать графические представления всей трассы или ее участков с произвольной степенью детализации;
- операция определения числа записей в интервале может быть обобщена до быстрого определения (приближенных) статистических характеристик интервалов трассы, актуальных для текущего анализа, таких как число вызовов функций, число переданных байт, число отправленных или принятых сообщений и т.д., которые могут быть использованы в графических представлениях всей трассы или ее интервалов;
- быстрое позиционирование на конкретную запись дает возможность построения развитой системы интерактивной навигации по данным трассы для пользователя анализирующего инструмента;
- быстрое позиционирование и прямой доступ к записи дает возможность отказаться от весьма затратной по времени постобработки всей собранной трассы, в той или иной форме присутствующей в имеющихся на рынке инструментальных средствах анализа параллельных приложений. Например, возможно отказаться от весьма трудоемкого предварительного полного слияния компонент записей данных коллективных MPI операций и точечных обменов между разными процессами. Используя прямой доступ к записям, слияние компонент можно выполнять во время анализа непосредственно «на лету» и лишь для записей, актуальных для текущего анализа, при текущем уровне детализации графических представлений данных.

В настоящее время формат XTF реализован, и работоспособность его проверена в интеграции с инструментом трассировки XMT (eXtended MPI Tracer), базирующемся на программном обеспечении с открытым кодом VampirTrace.

ЛИТЕРАТУРА:

1. Message Passing Interface Forum, "MPI: A Message Passing Interface". // In Proceedings of Supercomputing '93. IEEE Computer Society Press, November 1993, pp. 878-883.
2. Matthias S. Muller, Andreas Knupfer, Matthias Jurenz, Matthias Lieber, Holger Brunst, Hartmut Mix, Wolfgang E. Nagel "Developing Scalable Applications with Vampir, VampirServer and VampirTrace" // Proceedings of Parallel Computing: Architectures, Algorithms and Applications, ParCo 2007, Forschungszentrum Julich and RWTH Aachen University, Germany, 4-7 September 2007, pp.637-644.
3. Г.И. Воронов, В.Д. Трущин, В.В. Шумилин, Д.В. Ежов “Создание программного комплекса S-MPI для обеспечения разработки, оптимизации и выполнения высокопараллельных приложений на суперкомпьютерных кластерных и распределенных вычислительных системах”. // Тезисы докладов XIV Международной конференции "Супервычисления и математическое моделирование", Саров, 1-5 октября 2012, с.54-56.