ISSN 2079-3316 ПРОГРАММНЫЕ СИСТЕМЫ: ТЕОРИЯ И ПРИЛОЖЕНИЯ № 3(7), 2011, с. 53–60

УДК 519.687.4

А. А. Кузнецов, В. А. Роганов

Поддержка отказоустойчивых хранилищ данных в системе OpenTS

Аннотация. В статье рассматривается технология поддержки внешних отказоустойчивых хранилищ для счетных данных в системе параллельного программирования OpenTS. Такие хранилища могут быть использованы для улучшения свойства отказоустойчивости счета параллельных T++-приложений.

Ключевые слова и фразы: облачные хранилища данных, распределенные вычисления, динамическое распараллеливание программ, T-система с открытой архитектурой, отказоустойчивость.

1. Система параллельного программирования OpenTS

OpenTS — это система параллельного программирования [1], разработанная в ИПС РАН в 2000–2004 годах в рамках суперкомпьютерного проекта «СКИФ» Союзного государства России и Беларуси. После успешного завершения программы «СКИФ» разработка системы OpenTS [2–5] была продолжена в рамках программы «СКИФ-ГРИД» и различных академических проектов.

Система OpenTS (Open T-System) представляет собой современную и наиболее удачную реализацию концепции T-системы — программной среды параллельного программирования с поддержкой динамического распараллеливания T-приложений, которая сочетает в себе функциональную и императивную парадигмы программирования. OpenTS — это среда поддержки исполнения приложений, написанных на языке T++. Данный язык программирования является параллельным функциональным расширением Си++ и дополняет исходный язык всего семью новыми ключевыми словами. Среда поддержки исполнения T++-приложений (T-приложений) берет на себя основную часть работы по организации параллельного счета (синхронизация, распределение нагрузки, транспортировка сообщений).

Тем самым, система OpenTS позволяет снизить затраты на разработку параллельных программ, увеличить глубину параллелизма и более полно использовать возможности аппаратной части мультипроцессора за счет распараллеливания в динамике.

В рамках программ Союзного государства «Триада» и «СКИФ-ГРИД» [6] разработана архитектура и проведена экспериментальная реализация распределенной отказоустойчивой системы «SkyTS» [7–9], которая позволяет объединить разнородные ресурсы компьютеров в сети Интернет для счета T-приложений, решающих ресурсоемкие научно-прикладные задачи. SkyTS — управляющая надстройка над OpenTS для глобальных гетерогенных распределенных сред, реализованная на простом командном языке TCL. Включает в себя также пользовательский Web-интерфейс (на PHP) для практичного управления запуском множества T-приложений.

2. Использование распределенных хранилищ данных

При работе T++-приложений в распределенной системе SkyTS очень остро встает вопрос об обеспечении отказоустойчивости. Механизм отказоустойчивости T++-приложений реализован на базе модели перевычислений с использованием т. н. порталов. Они нужны для фиксации переходов T-подзадач из одного вычислительного подпространства в другое. В случае потери связи с узлом или выхода последнего из строя портал содержит в точности те T-функции, которые нужно перевычислить. Если в каком-то вычислительном подпространстве отсутствует (вышел из строя) необходимый ресурс, то T-функция планируется на исполнение в ближайшее подпространство, где данный ресурс есть в наличии. Возможно, и там не удастся выполнить подзадачу из-за сбоев — тогда последует новый переповтор. То есть, планировщик осуществляет назначение T-функции на узлы в соответствии с ее и их атрибутами.

В ИПС РАН разработан новый метод обеспечения отказоустойчивости работы T++-приложений, дополняющий существующие методы. Среда времени исполнения «OpenTS» параллельных T++-приложений доработана для поддержки счета этих приложений с использованием внешнего распределенного хранилища данных для T-задач. При запуске параллельных приложений в этом режиме значительно повышается свойство отказоустойчивости счета, и возрастает вероятность беспрепятственного продолжения счета после аварийного сбоя какого-либо счетного узла.

При работе T++-приложения в отказоустойчивом режиме генерируется множество счетных гранул — T-функций, которые перемещаются по сети между узлами для осуществления счета. При этом они с собой несут какие-либо данные, которые часто имеют большой объем и не всегда нужны в данный момент. Поэтому неэффективно держать их постоянно рядом с T-функциями во время распределенного счета. Разработан более эффективный подход, в котором вместе с T-функциями по сети переносятся не данные, а только их дескрипторы (ссылки на них). Данные же хранятся в отказоустойчивом хранилище и извлекаются по требованию. Проведен анализ отказоустойчивых распределенных хранилищ, выбрано подходящее решение и реализован механизм прозрачного для приложения сохранения и дереференсинга (извлечения при обращении) данных в/из суперпамяти T++-приложения. Таким образом, создан новый вид суперданных, которые устойчивы к сбоям и реальная доставка которых осуществляется только по требованию, что в условиях распределенных вычислений есть существенное преимущество.

Вся логика и весь сетевой протокол работы с распределенным отказоустойчивым хранилищем реализованы в ядре OpenTS, поэтому прикладному программисту для работы с этим хранилищем нужно лишь указать его номер директивой tct(cloud(N)) в коде T++ программы, где N — это условный порядковый номер хранилища. Если число N отлично от 0, то вместо встроенного механизма работы с хранилищем (на основе MemcacheDB) может быть использован новый механизм, позволяющий программе работать с другим типом хранилища. В этом случае используется так называемый Dynamic Multi Cloud Interface (DMCI), которым можно управлять с помощью директивы tct(cloud(N)) и переменных окружения. Возможности этого интерфейса позволяют динамически подключать хранилища произвольной природы, логика и протокол работы с которыми реализованы в виде динамических библиотек.

2.1. Система хранения «MemcacheDB»

При поиске подходящих хранилищ данных была выбрана и протестирована работа системы MemcacheDB [10]. Система предоставляет возможность создания и использования распределенного перманентного (persistent) хранилища данных вида «ключ-значение».

Система может быть задействована в «облачном» режиме работы T-приложений как надежное хранилище для T-данных. В системе MemcacheDB в качестве базы данных используется Berkeley DB, которая обладает рядом преимуществ: высокая скорость работы, транзакционность и дублирование (replication) данных. Система MemcacheDB в качестве эксперимента была успешно запущена в режиме репликации на нескольких сетевых узлах под управлением ОС Linux.

В структуру исходного кода системы OpenTS добавлен новый программный модуль, в котором реализован сетевой интерфейс взаимодействия между T-приложением и облачным хранилищем данных по сетевому протоколу «memcached». Это хранилище представляет собой базу данных MemcacheDB, запущенную на нескольких Linux-серверах с использованием модели репликации (при которой каждый узел содержит копию БД).

T-приложение, функционирующее в облачном режиме, обменивается данными с этой БД по протоколу memcached. Для реализации взаимодействия приложений с БД по этому протоколу существует сторонняя библиотека libmemcached. Чтобы избавиться от зависимости от этой библиотеки, необходимые функции memcached_set/get() были реализованы с использованием кросс-платформенного sockets API. Их имплементация была добавлена в микроядро OpenTS. В результате, T-приложения, в которых имеются «облачные» (tct(cloud)) T-данные, могут работать во всех ОС, поддерживающих классический Berkeley sockets API.

2.2. Облачные T-данные

В облачном хранилище (БД) произвольной природы (собственной разработки или любом существующем) каждая активная T-задача, работающая в отказоустойчивом режиме, формирует так называемую Тучу. Ключи значений в одной и той же Туче имеют префикс, равный идентификатору T-задачи. В Тучу прозрачным образом «кристаллизуются» при замораживании (то есть финализации T-значений) так называемые «облачные T-величины». Последние формируются в T-переменных, порожденных в контексте с атрибутом tct(cloud).

Облачные T-переменные ведут себя точно так же, как и обычные. Но при замораживании облачной T-величины (например, возврате таковой из T-функции) ее уже более неизменное в соответствии

с T-семантикой значение попадает в Тучу, то есть отправляется в облачное хранилище, а в T-величине реально остается только ключ, по которому ее значение можно получить где и, главное, когда угодно. Пересылка таких T-величин в глобальных сетях — это уже пересылка всего нескольких десятков байт, что позволяет существенно ускорить передачу данных во время счета.

При операции tdrop() и сборке мусора облачные T-величины также ведут себя вполне обычно, но только при этом удаляются соответствующие их ключам данные из Тучи. Туча существует, пока не завершится T-задача. Последняя периодически сохраняет контрольную точку (T-величину со специальным служебным ключом) на случай аварийного завершения счета.

Для сборки мусора после аварийного сбоя перед восстановлением из контрольной точки Тучу при желании можно сканировать методом mark&sweep и освобождать T-величины, на которые не обнаружено ссылок. Впрочем, по окончании работы T-задач их Тучи все равно будут очищены, так что это скорее дополнительная опция, а не норма жизни.

Среди основных программных правок, внесенных в микроядро системы OpenTS, можно выделить следующие:

- расширение понятия неготовых, готовых данных до неготовых, готовых, облачных;
- добавление параметризованного C++ класса TDataCloud, ответственного за облачное хранение;
- реализация соответствующей логики работы с облачными T-данными (замерзание/оттаивание T-данных из Тучи);
- реализация динамического интерфейса DMCI.

3. Заключение

В ИПС РАН разработан новый метод обеспечения отказоустойчивости работы T++-приложений, дополняющий существующие методы. Среда времени исполнения «OpenTS» параллельных T++-приложений доработана для поддержки счета этих приложений с использованием внешнего распределенного хранилища данных для T-задач. Разработанные программные средства и результаты исследований могут быть использованы с целью:

- разработки отказоустойчивых параллельных приложений;

- разработки систем распределенных вычислений (Грид-систем) для территориально-распределенной вычислительной среды;
- повышения надежности длительного счета на постоянно изменяющемся вычислительном поле, состоящем как из кластеров, так и одиночных компьютеров, эпизодически предоставляющих свои свободные вычислительные мощности.

Благодарности

Работы, положенные в основу данной статьи, были выполнены в рамках проектов:

- проект «Разработка и реализация языков T++ и соответствующих ему средств для эффективной поддержки высокопроизводительного параллельного счета» по Программе фундаментальных научных исследований ОНИТ РАН «Архитектура, системные решения, программное обеспечение, стандартизация и информационная безопасность информационно-вычислительных комплексов новых поколений» (2009–2011 гг.);
- суперкомпьютерная программа «СКИФ-ГРИД»: «Разработка и использование программно-аппаратных средств ГРИД-технологий и перспективных высокопроизводительных (суперкомпьютерных) вычислительных систем семейства «СКИФ» (2007–2010 гг.);
- научно-техническая программа Союзного государства «Развитие и внедрение в государствах-участниках Союзного государства наукоемких компьютерных технологий на базе мультипроцессорных вычислительных систем» (шифр «ТРИАДА») (2005–2008 гг.).

Список литературы

[1] Официальный сайт системы программирования OpenTS : Электронный сетевой ресурс, http://www.opents.net. ↑1

[2] Абрамов С. М., Адамович А. И., Инюхин А. В., Московский А. А., Роганов В. А., Шевчук Ю. В., Шевчук Е. В. *T-система с открытой архитектурой* // Суперкомпьютерные системы и их применение SSA'2004: Труды Международной научной конференции, 26–28 октября 2004 г., Минск, ОИПИ НАН Беларуси. — Минск, 2004, с. 18–22 ↑1

[3] Абрамов С. М., Кузнецов А. А., Роганов В. А. *Кроссплатформенная версия Т-системы с открытой архитектурой* // Параллельные вычислительные технологии (ПаВТ'2007): Труды Международной научной конференции, 29 января –2 февраля 2007 г., Челябинск. — Челябинск : изд. ЮУрГУ, 2007 Т. **1**, с. 115–121 ↑

[4] Кузнецов А. А., Роганов В. А. *Поддержка топологии вычислительного пространства в системе OpenTS* // Программные системы: теория и приложения, октябрь 2010 г. **1**, № 3, http://psta.psiras.ru/read/psta2010_3_93-106.pdf, с. 93–106 ↑

[5] Кузнецов А. А., Роганов В. А. *Экспериментальная реализация отказоустойчивой версии системы OpenTS для платформы Windows CCS* // Суперкомпьютерные системы и их применение (SSA'2008): Труды Второй Международной научной конференции, 27–29 октября 2008 г., Минск. — Минск : ОИПИ НАН Беларуси, 2008 ISBN 978-985-6744-46-7, с. 65–70 ↑1

[6] Официальный сайт научно-технической программы Союзного государства «СКИФ-ГРИД» : Электронный сетевой ресурс, http://skif-grid.botik.ru. ↑1

[7] Абрамов С. М., Есин Г. И., Загоровский И. М., Матвеев Г. А., Роганов В. А. *Принципы организации отказоустойчивых параллельных вычислений для решения вычислительных задач и задач управления в Т-Системе с открытой архитектурой (OpenTS)* // Программные системы: теория и приложения (PSTA-2006): Труды Международной научной конференции, 23–28 октября 2006 г., Переславль-Залесский, ИПС РАН. — Переславль-Залесский, 2006, с. 257–264 ↑1

[8] Есин Г. И., Кузнецов А. А., Роганов В. А. *Экспериментальная реализация отказоустойчивой системы распределенных вычислений «SkyTS» для параллельного счета ресурсоемких T++ приложений в гетерогенной распределенной вычислительной среде* // Программные системы: теория и приложения (PSTA-2009): Труды Международной научной конференции, май 2009 г., Переславль-Залесский, ИПС им. А.К. Айламазяна РАН / ред. Абрамов С. М., Знаменский С. В. — Переславль-Залесский : изд. Университета города Переславля, 2009 Т. **1** ISBN 978-5-901795-16-3, с. 225–244 ↑

[9] Абрамов С. М., Московский А. А., Роганов В. А., Велихов П. Е. Суперкомпьютерные и GRID-технологии. // Пути ученого. Е.П. Велихов / ред.В.П. Смирнов. М. : РНЦ Курчатовский институт, 2007 ISBN 978-5-9900996-1-6. — 314–324 с. ↑1

[10] Официальный сайт системы MemcacheDB : Электронный сетевой ресурс, http://memcachedb.org. ↑2.1

A. A. Kuznetsov, V. A. Roganov. *Cloud data storage support in the OpenTS parallel programming system.*

Abstract. The article describes a software technology for support of external fault-tolerant data storage in the OpenTS parallel programming system. This type of storage may be used to improve the fault-tolerance capabilities of T++ applications.

Key Words and Phrases: cloud data storage, distributed computing, dynamic program paralleling, T-system with an open architecture, fault-tolerance.

Образец ссылки на статью:

А. А. Кузнецов, В. А. Роганов. *Поддержка отказоустойчивых хранилищ данных в системе OpenTS* // Программные системы: теория и приложения : электрон. научн. журн. 2011. № 3(7), с. 53–60. URL: http://psta.psiras.ru/read/psta2011_3_53-60.pdf