УДК 004.272

Отказоустойчивая реализация параллельного решения двумерного уравнения теплопроводности

В. А. Марков

В работе выполнено исследование отказоустойчивой реализации параллельного решения двумерного уравнения теплопроводности методом последовательных итераций Якоби. Реализован подход обеспечения отказоустойчивого выполнения параллельной программы на уровне пользователя. Контрольные точки формируются каждым процессом и периодически записываются на носитель информации. Приведены результаты анализа накладных расходов на формирование контрольных точек.

Ключевые слова: параллельные вычисления, отказоустойчивость, контрольные точки, fault tolerant MPI, user-level checkpoint, checkpointing.

1. Введение

Современные высокопроизводительные вычислительные системы (ВС) являются большемасштабными. Количество процессорных ядер в таких системах составляет более одного миллиона. В частности, ВС Sunway TaihuLight, занимающая первое место в списке TOP500 (редакция ноябрь 2016 года), имеет 10 649 600 процессорных ядер [1].

В перспективных ВС ожидается увеличение числа вычислительных узлов на несколько порядков. По известным оценкам среднее время наработки на отказ в системах такого уровня, составит порядка 30 минут [2]. Поэтому актуальной является разработка программных средств обеспечения отказоустойчивого выполнения параллельных программ на большемасштабных ВС.

Основой техники обеспечения отказоустойчивого выполнения параллельных программ являются контрольные точки восстановления (КТ). Суть этой техники заключается в периодическом сохранении состояния параллельной программы в КТ, на носитель информации, с целью последующего возобновления вычислительного процесса из неё.

В данной работе рассматривается параллельная отказоустойчивая реализация решения двумерного уравнения теплопроводности. Распараллеливание выполнено в стандарте MPI путем двумерной декомпозиции расчетной области. Формирование КТ выполняется на уровне пользователя (user-level checkpoint). Экспериментальный анализ эффективности выполнен на кластерных вычислительных системах с сетями связи стандартов Gigabit Ethernet и InfiniBand QDR.

2. Методы отказоустойчивого выполнения параллельных программ

Под отказоустойчивостью понимают способность параллельной программы (ПП) выполнять работу даже при наличии отказов [3]. Рассмотрим методы обеспечения отказоустойчивого выполнения ПП в распределенных ВС.

Методы восстановления (rollback-recovery) [4, 5, 6] вычислительного процесса после возникновения отказа принято разделять на методы прямого (forward recovery) и обратного восстановления (backward recovery).

Методы прямого восстановления возвращают ПП в безошибочное состояние на основе текущих данных и по результатам анализа отказа, без обращения к предыдущим состояниям параллельной программы. Эти методы основаны на специальных алгоритмах, например, широко известны стабилизирующиеся алгоритмы [7].

Методы обратного восстановления также заключаются в восстановлении ПП в корректное состояние из КТ, однако они основаны на использовании информации о полном или частичном состоянии ПП.

Однако создание КТ связано с большой нагрузкой на сеть и систему хранения данных ВС. Среднее время сохранения КТ может быть весьма значительным и для некоторых систем составляет 20-30 минут [8]. С целью уменьшения затрат на частое создание КТ были разработаны методы, которые включают в себя дополнительные механизмы сохранения, позволяющие сократить время записи КТ. Приведем описание трех групп подобных методов.

В первую группу входят методы, основанные на механизмах создания журналов передачи сообщений (log-based rollback-recovery). Создание журналов передачи сообщений заключается в том, что на стороне посылающего сохраняется содержание сообщения, информация о получателе и о моменте получения данного сообщения. Использование такого механизма позволяет снизить нагрузку на сеть, так как запись локальных КТ может осуществляться в произвольное время. Во время восстановления, используя журналы передачи сообщений, необходимо произвести дополнительную работу на поиск согласованного состояния ПП.

Методы, второй группы [9, 10] основаны на сохранении изменений между двумя последовательными КТ, к ним относятся инкрементный (incremental checkpointing) и дифференциальный (differential checkpointing) методы. Первый предусматривает сохранение в текущей промежуточной КТ изменений относительно предыдущей. Второй сохраняет все изменения относительно базовой КТ.

В третью группу входят методы [11, 12, 13], которые предполагают сохранение КТ в память вычислительных узлов.

В данной работе подробно рассматривается формирование КТ на уровне пользователя. Использование данного подхода к формированию КТ позволяет контролировать время и частоту создания КТ во время выполнения ПП, например, выбирается наиболее удобный момент для создания КТ.

Таким образом накладные расходы на формирование КТ могут быть значительно уменьшены, однако, потребуется дополнительная работа прикладного программиста для реализации отказоустойчивости в программе.

3. Параллельная программа решения двумерного уравнения теплопроводности

Решение двумерного уравнения теплопроводности, реализовано методом последовательных итераций Якоби. Метод распараллелен в стандарте MPI путем двумерной декомпозиции расчетной области. Каждому процессу назначена подматрица и теневые ячейки по всем четырем направлениям, для расчета значений на границах выделенной области.

В методе последовательных итерация Якоби новое значение в каждой точке подматрицы равно среднему из предыдущих значений четырёх её соседних точек (слева, справа, сверху и снизу). Для расчета требуется две подматрицы, одна нужна для представления области и её границы, а другая – для хранения новых значений. Границы подматриц инициализируется соответствующими граничными условиями, а внутренние точки начальными значениями. Обмен теневыми ячейками выполняется на каждом шаге итерационного процесса (рис. 1).

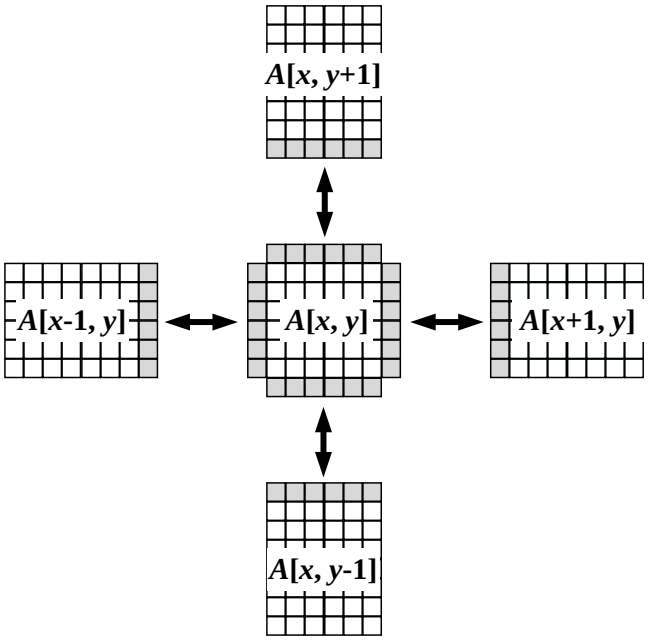

Рис. 1. Схема обмена теневыми ячейками подматрицы $A[x, y]$

Вычисления завершаются, когда разность между новым, полученным на итерации, и предыдущим значением по модулю не большие заданной величины.

Таким образом главный цикл вычислений по методу последовательных итераций Якоби, распараллеленный в стандарте MPI путем двумерной декомпозиции, имеет следующий вид (рис. 2).

```
while true do
      for 1 to ny
            for 1 to nx
                  // Расчет значений внутренних точек
            end for
      end for

      for 1 to ny
            for 1 to nx
                  maxdiff = fmax(maxdiff,...)
            end for
      end for

      if maxdiff < EPS then // Проверка условия на выход
            break
      end if

      MPI_Irecv(...) // Прием теневых ячеек от соседа сверху
      MPI_Isend(...) // Передача теневых ячеек соседу сверху
     // ... Обмен теневыми ячейками с другими соседями
      MPI_Waitall(...)
end while
```

Рис. 2. Главный цикл вычисления значений в подматрице и обмена теневыми ячейками

4. Формирование контрольных точек

Для обеспечения отказоустойчивого моделирования в программу добавлена возможность сохранения КТ и возобновления вычислительного процесса из них.

Программа разбита на три блока:

- Блок 1 – инициализация значениями, формирование первой КТ;
- Блок 2 – основной цикл по итерациям, в конце каждой итерации периодически

формируется КТ;

- Блок 3 – завершение вычислений, формирование финальной КТ.

Формирование и создание КТ реализовано во внешней библиотеке. КТ формируется каждым MPI процессом и содержит:

- Подматрицу MPI процесса;
- Метаданные КТ (время выполнения, количество созданных КТ, время затраченное на создание КТ, признак целостности КТ), формируемые созданной библиотекой автоматически.

В случае возникновения отказа, пользователь производит перезапуск программы с заданной КТ (рис. 3).

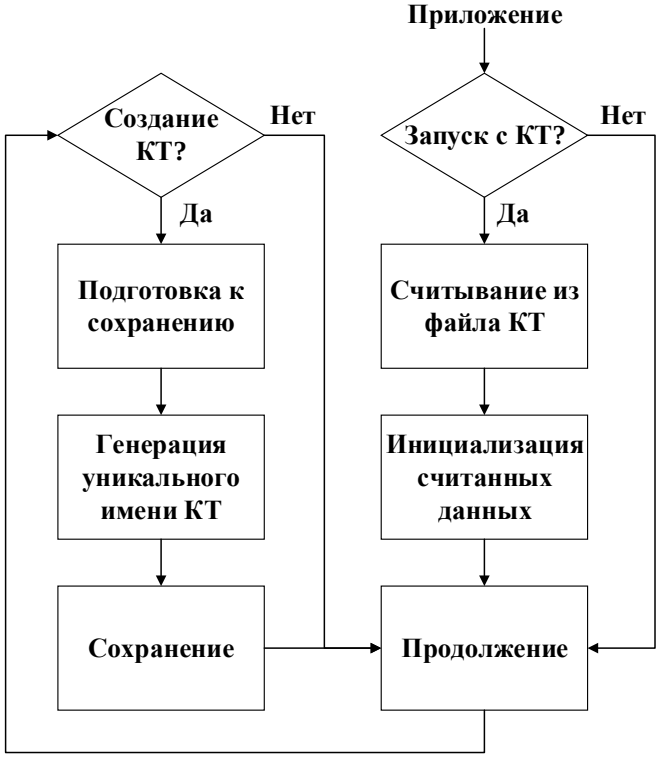

Рис. 3. Алгоритм создания КТ и возобновления вычислительного процесса

5. Эксперименты

Экспериментальное исследование проводилось на следующих кластерных ВС:

- Кластер Jet, 18 вычислительных узлов, конфигурация узла: два четырёх ядерных процессора Intel Xeon E5420, размер оперативной памяти – 8 GiB, сеть – Gigabit Ethernet, операционная система – GNU/Linux Fedora 20 x86_64 (версия ядра 3.12.10-300), компилятор – GCC 4.8.3, MPICH 3.2;
- Кластер Oak, 6 вычислительных узлов, конфигурация узла: два четырёх ядерных процессора Intel Xeon E5620, размер оперативной памяти – 24 GiB, сеть – InfiniBand QDR, операционная система – GNU/Linux x86_64 (версия ядра 2.6.32-573.22.1), компилятор - GCC 4.8.3, MVAPICH 2.2a.

Моделирование проводилось с формированием 3 КТ (рис. 4, 5), сохранение которых проводилось носитель информации, доступный по сети Gigabit Ethernet через файловую систему NFS (NFSv3). Накладные расходы на сохранение КТ в зависимости от размера расчетной области (показывают во сколько раз увеличилось время моделирования с использованием КТ ко времени моделирования без использования КТ) представлены в таблице 1 (для кластера Jet).

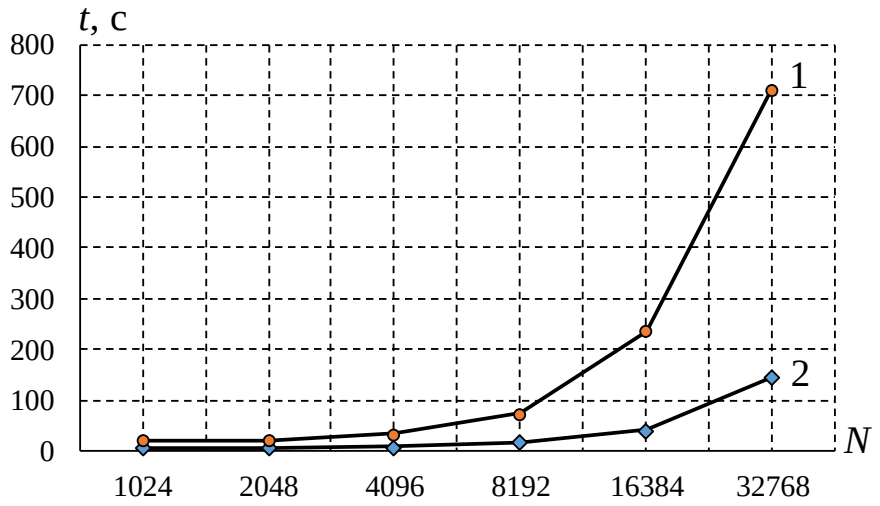

Рис. 4. Зависимость времени *t* выполнения параллельной программы от размера *N* расчетной области на кластере Oak (32 процесса – 4 вычислительных узла):
1 – время выполнения исходной параллельной программы;
2 – время выполнения параллельной программы с формированием КТ

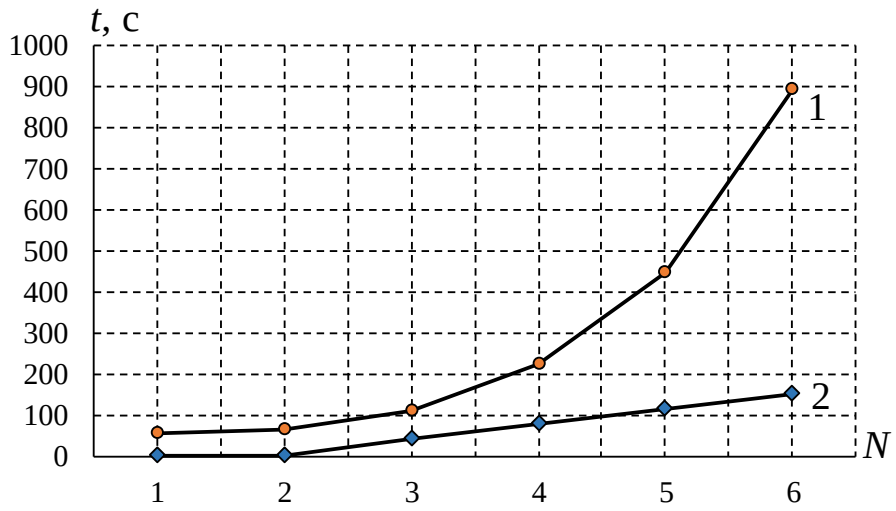

Рис. 5. Зависимость времени *t* выполнения параллельной программы от размера *N* расчетной области на кластере Jet (128 процессов – 16 вычислительных узлов):
1 – время выполнения исходной параллельной программы;
2 – время выполнения параллельной программы с формированием КТ

Таблица 1. Накладные расходы на формирование КТ (кластер Jet)

Размер расчетной области, *N*	Накладные расходы на сохранение КТ, %
1024	310
2048	312
4096	359
8192	441
16384	563
32768	621

Сохранение КТ на централизованное устройство хранения с использованием NFS, приводит к существенному увеличению времени работы моделирования, повышенной нагрузки на сеть, необходим большой объем дискового пространства для сохранения КТ.

6. Заключение

Разработана параллельная отказоустойчивая реализация решения двумерного уравнения теплопроводности. Распараллеливание выполнено в стандарте MPI путем двумерной декомпозиции расчетной области. Реализовано формирование КТ на уровне пользователя. Экспериментальный анализ эффективности выполнен на кластерных вычислительных системах с сетями связи стандартов Gigabit Ethernet и InfiniBand QDR.

Эксперименты показали, что накладные расходы на формирование КТ зависят от размера входных данных.

Направление дальнейшей работы разработка алгоритмов, оптимизации накладных расходов на формирование КТ.

Литература

1. Список TOP500 URL: https://www.top500.org/lists/2016/11/
2. А. А. Бондаренко, М. В. Якобовский, Обеспечение отказоустойчивости высокопроизводительных вычислений с помощью локальных контрольных точек, Вестн. ЮУрГУ. Сер. Выч. матем. информ., 2014, том 3, выпуск 3. URL: http://cyberleninka.ru/article/n/obespechenie-otkazoustoychivosti-vysokoproizvoditelnyh-vychisleniy-s-pomoschyu-lokalnyh-kontrolnyh-tochek
3. Avizienis A., Laprie J.C., Randell B., Landwehr C. Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Transactions on Dependable and Secure Computing. 2004. Vol. 1, P. 11–33.
4. Elnozahy E.N., Alvisi L., Wang Y.,Johnson D.B. A Survey of Rollback-Recovery Protocols in Message-Passing Systems. ACM Computing Surveys. 2002. Vol.34, No. 3 P. 375– 408.
5. Koren I., Krishna C.M. Fault-Tolerant Systems. San Francisco, CA: Morgan Kaufmann Publishers Inc., 2007. 378 p.
6. Tanenbaum A.S., Steen M. Distributed Systems: Principles and Paradigms. New Jersey, Prentice Hall PTR, 2002. 803 p.
7. Tel G. Introduction to Distributed Algorithms. Cambridge University Press, 2000. 596 p.
8. Cappello, F. Fault tolerance in petascale/exascale systems: Current knowledge, challenges and research opportunities. International Journal of High Performance Computing Applications. 2009. Vol. 23, No. 3. P. 212–226.
9. Ferreira K.B., Riesen R., Bridges P.G., Arnold D., Brightwell R. Accelerating incremental checkpointing for extreme-scale computing. Future Generation Computer Systems. 2014. Vol. 30, No 1. P. 66–77.
10. Поляков А.Ю., Данекина А.А. Оптимизация времени создания и объёма контрольных точек восстановления параллельных программ // Вестник СибГУТИ. – Новосибирск: СибГУТИ, 2010. – № 2. – С. 87-100.
11. Vaidya N.H. A Case for Two-Level Distributed Recovery Schemes. Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems (Ottawa, Canada, May 15-19 1995) ACM, 1995. P. 64–73.
12. Plank J.S., Li K., Puening M.A. Diskless Checkpointing. IEEE Transanctions on Parallel Distributed Systems. 1998. Vol. 9, No 10. P. 972–986.
13. Dong X., Muralimanohar N., Jouppi N.P., Xie Y. A Case Study of Incremental and Background Hybrid In-Memory Checkpointing. Proceedings of the 2010 Exascale Evaluation and Research Techniques Workshop (Pittsburgh, PA, USA March 13 – 14, 2010), ACM, 2010. P. 119–147.

Марков Владислав Алексеевич
магистрант первого курса кафедры вычислительных систем СибГУТИ, e-mail: vl.markv@yandex.ru.

The fault-tolerant implementation of parallel solution of two-dimensional heat equation

V. Markov

This paper presents the study of fault-tolerant parallel implementation of two-dimensional heat equation solving by Jacobi iteration method. Implemented approach providing fault tolerance execution of the parallel program at the user level. Checkpoints are generated by each process, and periodically saved to the storage. The results of the analysis checkpointing overhead are presented.

Keywords: parallel computing, fault tolerance, fault tolerant MPI, user-level checkpoint, checkpointing.