В. А. Роганов, В. И. Осипов, Г. А. Матвеев

Интеграция приложения Tachyon с системой параллельного программирования OpenTS

Аннотация. Работа описывает интеграцию приложения Tachyon из библиотеки Intel® Threading Building Blocks с системой параллельного программирования OpenTS

Ключевые слова и фразы: Т-система с открытой архитектурой, OpenTS, язык программирования T++, tachyon, Ray Tracing.

Программа Tachyon, реализующая алгоритм трассировки лучей, предназначена для рендеринга изображений. Исходное описание изображения описано на языке описания сцен в текстовом формате. В исходном файле описывается расположение и размер объектов, источника света, расположение и ориентация камеры. Приложение создает изображение, содержащее тени и отражения объектов. Приложение Tachyon разработанное Джоном Е. Стоуном [1], было включено в бенчмарк SPEC MPI [2], затем в библиотеку Intel® Threading Building Blocks [3].

В данной статье описывается интеграция приложения Tachyon с системой параллельного программирования OpenTS [4–7], приведены результаты сравнительных испытаний с приложением PovRay и реализациями Tachyon на OpenMP, Cilk и Intel® TBB. Для интеграции была выбрана последовательная версия приложения из Intel® Threading Building Blocks. В качестве модели использовалось изображение, описанное в файле (модели) balls.dat. После работы последовательной версии приложения было создано изображение на рис. 1. Испытания проводились на кластере, работающем под операционной системой Linux.

Параллельная версия приложения tachyon.with_OpenTS имеет дополнительные параметры, количество строк и количество столбцов. Запускается оно так:

Shell –

```
1 mpirun --np N./tachyon.with_OpenTS NRows NColumns, где N = NRows*NColumns
```

Например, если запущена команда

Рис. 1. Последовательная версия изображения 7381 сфер «Sphereflake»

Shell —

```
1 mpirun --np 6 ./tachyon.with_OpenTS 2 3
```

то параллельно создаются 6 кусков изображения на рис. 2

Интеграция приложения состоит в распараллеливании вложенного цикла `for` в функции `draw_task()`.

Вложенный цикл `for` для последовательной версии:

T-System —

```
1 for (int y = starty; y < stopy; y++)
2     for (int x = startx; x < stopx; x++)
3         Рендерить_пиксель(x,y);
```

Рис. 2. Результат работы на 6 вычислительных узлах приложения `tachyon.with_OpenTS`. Каждый узел параллельно создает только часть изображения (всего 6 частей)

Здесь `startx`, `stopx`, `starty`, `stopy` — границы изображения по координатам x и y .

Вложенный цикл `for` для параллельной версии приложения `tachyon.with_OpenTS`:

```

1 for (int y = y1; y < y2; y++)
2     for (int x = x1; x < x2; x++)
3         Рендерить_пиксель(x,y);

```

Здесь

Рис. 3. Графики $time\%_T$ для приложений tachyon и povray, интегрированных с OpenTS

T-System —

```

1  int y1 = rowID * stopy / NRows;
2  int y2 = (rowID + 1) * stopy / NRows;
3  int x1 = columnID * stopx / NColumns;
4  int x2 = (columnID + 1) * stopx / NColumns;

```

где

T-System —

```

1  int columnID = threadID / NRows;
2  int rowID = threadID % NRows;

```

$threadID$ – номер текущего вычислительного узла.

Приложение `tachyon.with_OpenTS` параллельно запускает N задач, по одной задаче на каждом вычислительном узле. Каждая из задач создает свою часть изображения и записывает ее в файл формата `bmp`.

Ранее [8] была произведена интеграция системы параллельного программирования OpenTS с приложением PovRay, реализующим алгоритм трассировки лучей. Результаты сравнительных испытаний приложений Tachyon и PovRay, интегрированных с OpenTS, представлены на рис. 3.

Рис. 4. Зависимость времени работы от количества узлов вычислительной системы для приложений `tachyon.with_OpenMP`, `tachyon.with_OpenTS`, `tachyon.with_tbb` и `tachyon.with_Cilk`

Здесь $\text{time\%}_T(N) = \text{time}_T(N) / \text{time}_C$, где time_C , $\text{time}_T(N)$ — времена выполнения в секундах C-реализации и T-реализации соответственно, зависящие от числа процессоров N .

Из графика видно, что алгоритмы распараллеливания приложений Tachyon и PovRay имеют одинаковую эффективность.

График зависимости времени работы приложения от количества вычислительных узлов изображен на рис. 4. На этом же графике приведены результаты испытаний для реализаций OpenMP, Cilk и Intel®TBB.

Из графика видно, что эффективности алгоритмов распараллеливания приложения Tachyon в реализациях на OpenMP, Cilk, Intel®TBB и OpenTS близки друг к другу.

Список литературы

- [1] *Tachyon Parallel Multiprocessor Ray Tracing System*, URL: <http://jedi.ks.uiuc.edu/~johns/raytracer/> ³¹⁹
- [2] *SPEC MPIR 2007 benchmark suite*, URL: <https://www.spec.org/mpi2007/> ³¹⁹

- [3] *IntelR Threading Building Blocks*, URL: <https://software.intel.com/en-us/intel-tbb> ↑ ³¹⁹
- [4] С. М. Абрамов, В. А. Васенин, Е. Е. Мамчиц, В. А. Роганов, А. Ф. Слепухин. «Динамическое распараллеливание программ на базе параллельной редукции графов. Архитектура программного обеспечения новой версии Т-системы», *Научная сессия МИФИ-2001*, Сборник научных трудов. Т. 2 (Москва, 22–26 января 2001 г.), с. 234–235. ↑ ³¹⁹
- [5] С. М. Абрамов, А. А. Кузнецов, В. А. Роганов. «Кроссплатформенная версия Т-системы с открытой архитектурой», *Труды Международной научной конференции «Параллельные вычислительные технологии (ПаВТ'2007)»*. Т. 1 (Челябинск, 29 января–2 февраля 2007 г.), Изд. ЮУрГУ, Челябинск, с. 115–121. ↑ ³¹⁹
- [6] С. М. Абрамов, А. А. Кузнецов, В. А. Роганов. «Кроссплатформенная версия Т-системы с открытой архитектурой», *Вычислительные методы и программирование*, 8:1(2) (2007), с. 175–180, URL: http://num-meth.srcc.msu.ru/zhurnal/tom_2007/v8r203.html ↑ ³¹⁹
- [7] *OpenTS. Руководство программиста*, <http://www.opents.net/index.php/ru/ruk-progr>. ↑ ³¹⁹
- [8] С. М. Абрамов, И. М. Загоровский, М. Р. Коваленко, Г. А. Матвеев, В. А. Роганов. «Миграция от MPI к платформе OpenTS: эксперимент с приложениями PovRay и ALCMD», *Международная конференция «Программные системы: теория и приложения»*. Т. 1 (Переславль-Залесский, октябрь 2006), Наука. Физматлит, М., 2006, с. 265–275. ↑ ³²²

Рекомендовал к публикации

Программный комитет

Шестого национального суперкомпьютерного форума **НСКФ-2017**

Пример ссылки на эту публикацию:

В. А. Роганов, В. И. Осипов, Г. А. Матвеев. «Интеграция приложения Tachyon с системой параллельного программирования OpenTS», *Программные системы: теория и приложения*, 2017, 8:4(35), с. 319–326.

URL: http://psta.psiras.ru/read/psta2017_4_319-326.pdf

Об авторах:

Владимир Александрович Роганов

Научный сотрудник ИПС им. А.К. Айламазяна РАН. Разработчик современных версий Т-системы, ведущий разработчик системы OpenTS. Принимал активное участие в суперкомпьютерных проектах Союзного государства России и Беларуси, в том числе в проектах «СКИФ» и «СКИФ-ГРИД»

e-mail:

var@pereslavl.ru

Валерий Иванович Осипов

К.ф.-м.н., научный сотрудник ИПС им. А.К. Айламазяна РАН. Один из разработчиков системы OpenTS. Принимал участие в суперкомпьютерных проектах Союзного государства России и Беларуси

e-mail:

val@pereslavl.ru

Герман Анатольевич Матвеев

Ведущий инженер-исследователь ИЦМС ИПС им. А.К. Айламазяна РАН. Один из разработчиков системы OpenTS. Принимал участие в суперкомпьютерных проектах Союзного государства России и Беларуси

e-mail:

gera@prime.botik.ru

Vladimir Roganov, Valerii Osipov, German Matveev. *Integration of Ray Tracing system Tachyon and T-system with an open architecture (OpenTS)*.

ABSTRACT. This paper describes integration of Ray Tracing system Tachyon of Intel® Threading Building Blocks library and T-system with an open architecture (OpenTS). (In Russian).

Key words and phrases: dynamic parallelization, T-system with an open architecture, OpenTS, T++ language, parallel programming, tachyon, Ray Tracing, Intel TBB.

References

- [1] *Tachyon Parallel Multiprocessor Ray Tracing System*, URL: <http://jedi.ks.uiuc.edu/~johns/raytracer/>
- [2] *SPEC MPIR 2007 benchmark suite*, URL: <https://www.spec.org/mpi2007/>
- [3] *IntelR Threading Building Blocks*, URL: <https://software.intel.com/en-us/intel-tbb>
- [4] S. M. Abramov, V. A. Vasenin, Ye. Ye. Mamchits, V. A. Roganov, A. F. Slepukhin. “Dynamic parallelization of programs based on parallel reduction of graphs. Software architecture of the new version of the T-system”, *Nauchnaya sessiya MIFI-2001*, Sbornik nauchnykh trudov. V. 2 (Moskva, 22–26 yanvarya 2001 g.), pp. 234–235 (in Russian).
- [5] S. M. Abramov, A. A. Kuznetsov, V. A. Roganov. “Cross-Platform Version of the T-System with an Open Architecture”, *Trudy Mezhdunarodnoy nauchnoy konferentsii “Parallel’nyye vychislitel’nyye tekhnologii (PaVT’2007)”*. V. 1 (Chelyabinsk, 29 yanvarya–2 fevralya 2007 g.), Izd. YuUrGU, Chelyabinsk, pp. 115–121 (in Russian).
- [6] S. M. Abramov, A. A. Kuznetsov, V. A. Roganov. “Cross-Platform Version of the T-System with an Open Architecture”, *Vychislitel’nyye metody i programirovaniye*, 8:1(2) (2007), pp. 175–180 (in Russian), URL: http://num-meth.srcc.msu.ru/zhurnal/tom_2007/v8r203.html
- [7] *The OpenTS parallel programming system*, <http://www.opents.net/index.php/ru/ruk-progr>.
- [8] S. M. Abramov, I. M. Zagorovskiy, M. R. Kovalenko, G. A. Matveyev, V. A. Roganov. “Migrating from MPI to OpenTS Platform: experience with the PovRay and ALCMD applications”, *Mezhdunarodnaya konferentsiya “Programnyye sistemy: teoriya i prilozheniya”*. V. 1 (Pereslavl’-Zalesskiy, oktyabr’ 2006), Nauka. Fizmatlit, M., 2006, pp. 265–275 (in Russian).

Sample citation of this publication:

Vladimir Roganov, Valerii Osipov, German Matveev. “Integration of Ray Tracing system Tachyon and T-system with an open architecture (OpenTS)”, *Program systems: Theory and applications*, 2017, 8:4(35), pp. 319–326. (In Russian). URL: http://psta.psiras.ru/read/psta2017_4_319-326.pdf