Науковий вісник НЛТУ України
Scientific Bulletin of UNFU
https://nv.nltu.edu.ua
https://doi.org/10.15421/40280826
Article received 24.09.2018 p.
Article accepted 25.10.2018 p.
УДК xa4abu@ukr.net
ISSN 1994-7836 (print)
ISSN 2519-2477 (online)

@ Correspondence author
N. O. Semenyshyn
xa4abu@ukr.net

М. В. Дендюк, П. І. Рожак, Н. О. Семенишин
Національний лісотехнічний університет України, м. Львів, Україна

МЕТОДИКА ЗБІЛЬШЕННЯ ШВИДКОСТІ ПАРАЛЕЛЬНОГО РЕНДЕРИНГУ ЗА ДОПОМОГОЮ BITTORRENT ПРОТОКОЛУ

Останнім часом дедалі частіше використовують так званий паралельний рендеринг, оскільки це істотно економить час. Існує чимало засобів для такого виду рендерингу, але вони переважно мають певні обмеження. До прикладу, рендеринг може здійснюватись тільки в межах локальної мережі (напр. у V-Ray чи Corona Renderer), або ще одне обмеження – рендеринг виконується тільки на GPU (напр. Octane Render), для цього необхідно мати потужні відеокарти, вартість яких може досягати кількох тисяч доларів. Ще один засіб паралельного рендерингу – це рендер-ферми, оренда яких може коштувати від кількох доларів до кількох тисяч доларів за добу користування. Тут треба додати ще одну проблему, а саме – обмежена швидкість передачі даних та ресурсів від одного комп'ютера на інший. У цій роботі представлено методику і приклад програми, яка дає змогу обійти більшість зазначених вищих обмежень, а саме: збільшити швидкість передачі даних, використовуючи BitTorrent протокол, і цим самим збільшити швидкість всього паралельного рендерингу; цей підхід працює не тільки в межах локальної мережі, але й у глобальній мережі WAN; представлена програма може істотно здешевити паралельний рендеринг, оскільки всі комп'ютери-вузли працюють за принципом "ти мені, а я тобі" і можуть надавати один одному потужності свого комп'ютера (а саме CPU) в обхід так званих рендер-ферм. Наведено результати паралельного рендерингу (кінцеве зображення) з використанням створеної програми та визначено затрачений час. Як 3d-модель для рендера використано лісосушильну камеру, створену засобами 3ds Max від компанії Autodesk. В основі самого процесу рендерингу використано програмний продукт V-Ray Chaos Group.

Ключові слова: паралельний рендеринг; BitTorrent протокол; peer-to-peer архітектура; лісосушильна камера.

Вступ. На сьогодні рендеринг є важливим інструментом для створення фотореалістичної візуалізації, мета якої затверджувати та впроваджувати концепції, виявляти проблеми та продавати продукти. Очевидно, що концептів та ідей, які згодом втілюються в кінцевий продукт, стає щораз більше, проблема тільки в тому, щоб швидко втілити цю ідею, і ефективним засобом для цього є комп'ютер та графічний редактор.

Функціональні графічні редактори повинні включати серед всього іншого засоби для рендерингу, і процес фотореалістичної візуалізації може обмежуватись тільки втіленням ефективних математичних алгоритмів в рамках ПЗ та потужністю комп'ютера. І якщо удосконалення першого може покращити рендеринг в рази, то удосконалення другого може дати пришвидшення в сотні разів і один із таких способів є паралельний рендеринг, де кластер з багатьох комп'ютерів створює, так би мовити, один потужний "суперкомп'ютер".

Актуальність теми рендерингу, зокрема паралельного рендерингу та існуючі результати в цій області дали поштовх для проведення цього дослідження та шляхів пришвидшення процесу візуалізації зображення, а також визначення ідейно нових шляхів використання обчислювальних ресурсів на добровільно-безоплатній основі.

Рендеринг. У комп'ютерній графіці рендеринг – це обчислювальний процес, за допомогою якого ми отримуємо 2d-зображення 3d-моделі з допомого програми "рендера" (напр. V-Ray, Scanline, Mental ray, Maxwell та ін). Паралельний рендеринг – це обчислювальний процес створення растрового зображення на підстаі 3d-моделі, розподілений між комп'ютерами.

Формальним математичним описом рендерингу є рівняння переносу випромінювання, яке визначає кількість світлового випромінювання у певному напрямку як суму власного та відбитого випромінювань (Kajiya, 1986). Тоді кількість вихідного випромінювання (L_0) – це сума випроміненого (L_e) і відбитого світла. Відбите світло може подаватися у вигляді суми випромінювання (L_i), що проходить по всіх напрямках, помноженого на коефіцієнт відбиття з даного кута, з часом t, в заданій точці x та довжиною хвилі λ.

$$L_0(x,\omega,\lambda,t) = L_e(x,\omega,\lambda,t) + \\ + \int_{\Omega} f_r(x,\omega',\omega,\lambda,t) L_i(x,\omega',\lambda,t)(-\omega' \cdot n) d\omega', \quad (1)$$

Інформація про авторів:

Дендюк Михайло Володимирович, канд. техн. наук, доцент, кафедра інформаційних технологій. **Email:** dendiuk@ukr.net
Рожак Петро Ігорович, аспірант, асистент, кафедра інформаційних технологій. **Email:** petrorozhak@gmail.com
Семенишин Назар Олегович, аспірант, асистент, кафедра інформаційних технологій. **Email:** xa4abu@ukr.net

Цитування за ДСТУ: Дендюк М. В., Рожак П. І., Семенишин Н. О. Методика збільшення швидкості паралельного рендерингу за допомогою bittorrent протоколу. Науковий вісник НЛТУ України. 2018, т. 28, № 8. С. 132–135.
Citation APA: Dendiuk, M. V., Rozhak, P. I., Semenyshyn, N. O. (2018). The method for increasing the speed of parallel rendering using the bittorrent protocol. *Scientific Bulletin of UNFU, 28*(8), 132–135. https://doi.org/10.15421/40280826

де: $f_r(x,\omega',\omega,\lambda,t)$ – двонаправлена функція розподілу відбиття, кількість випромінювання, відбитого від ω' до ω в точці x, в час t, на довжині хвилі λ; $-\omega' \cdot n$ – поглинання вхідного випромінювання за заданим кутом. Усі конкретні алгоритми можуть бути представлені як вирішення деякого формулювання цього рівняння.

Для того, щоб виконати рендеринг сцени, створена програма використовує утиліту 3dsmaxcmd.exe, яка йде в комплекті з програмою 3ds Max і представляє собою інтерфейс командного рядка з можливістю задавати безліч параметрів процесу. Серед таких параметрів найважливішими є розмір сцени, назва вихідного зображення, шлях до сцени, налаштування камери, назва пресетів, фрагмент зображення, який потрібно обробити, та ін. Власне сам процес рендерингу виконується вказаним у 3ds Max рендером, і для цього дослідження було обрано V-Ray компанії Chaos Group.

BitTorrent протокол. Цей протокол є відкритим і описує обмін інформацією у мережах типу peer-to-peer (P2P). Основна ідея протоколу в тому, що навантаження розподіляється на всіх членів роздачі того чи іншого файлу (*Bittorrent Protocol Specification*). Тобто завантаження відбувається не напряму з комп'ютера сервера в повному об'ємі, а по пакетах передається спочатку одному користувачу, згодом іншому вже від двох попередніх і так далі.

Розглянемо цей протокол детальніше. Нижче наведено формули, які порівнюють швидкість передавання даних для клієнт-серверної архітектури та P2P архітектури. Отже, час, витрачений на передавання одного пакету даних всім користувачам, можна обчислити (Somani, 2012):

$$T_{CS} = t \cdot N; \quad (2)$$

$$T_{P2P} = t \cdot (1 + \log_2(N)), \quad (3)$$

де: T_{CS} – час, витрачений на передавання одного пакету даних всім користувачам для клієнт-серверної архітектури; T_{P2P} – час, витрачений на передавання одного пакету даних всім користувачам для P2P архітектури; N – кількість вузлів; t – час на відправлення цілого пакету від одного вузла до іншого. Як бачимо, передавання даних для P2P архітектури буде вдвічі швидшим порівняно з клієнт-серверною архітектурою за кількості вузлів 8 та в 11 разів швидше за кількості вузлів 1024. На рис. 1 показано залежність часу передачі даних від кількості комп'ютерів при t=1 для обох архітектур.

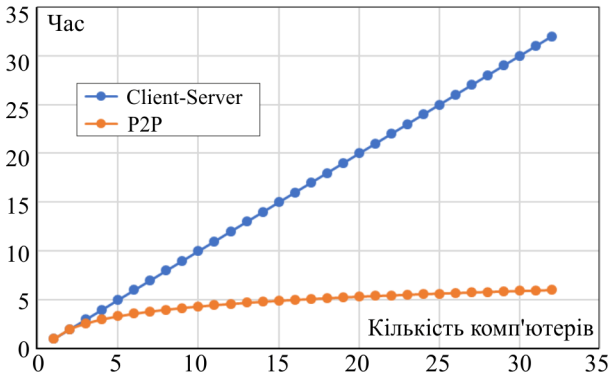

Рис. 1. Залежність часу передачі даних від кількості комп'ютерів для клієнт-серверної та P2P архітектури

Для того, щоб використати переваги BitTorrent протоколу, ми використали вільну бібліотеку Libtorrent, написану на C++. Ця бібліотека надає такі можливості, як створення торент-файлу (з розширенням.torrent), запуск торент-сесії і роздачі файлу – тобто процесу, в ході якого файл розбивається на пакети і передається іншим пірам (учасникам роздачі), обмін повідомленнями пірів та ін. Також можна вибудовувати будь-яку політику розподілу навантаження, тобто вказувати відношення відправлених пакетів до отриманих для кожного клієнта, перекидати навантаження на інших пірів у разі перевантаження якогось одного, вказувати розмір одного пакету та швидкість завантаження/відвантаження даних.

Переваг цього протоколу є дуже багато, саме тому він став таким популярним. Але з'явились також і деякі негативні наслідки. Наприклад, почало поширюватись цифрове "піратство" на таких сайтах, як "thepiratebay", де можна скачати торент-файли з неліцензійним програмним забезпеченням. Але автори статті закликають не плутати поняття "піратського" торент-файлу зі самою технологією (протоколом) передавання даних BitTorrent.

Ще одним важливим етапом торент-роздачі є торент-трекер – сервер, який зберігає IP-адреси та номери портів пірів учасників роздачі та на вимогу повідомляє ці адреси новому учаснику роздачі. Адресу торент-трекера, як правило, вказують у торент-файлі. У цьому проекті ми використали вільний XBT-трекер, написаний на C++.

Розпаралелювання задач для рендерингу. У розробленій програмі нашою основною задачею є рендеринг 3d-моделі у кінцеве растрове зображення. Для цього потрібно здійснити ряд математичних процедур серед яких: затінення, базисом якого може бути рівняння (1); 3D проекція, яка відображає тривимірні точки на двовимірній площині; растеризація – перетворення векторного зображення у растрове та ряд інших (Fletcher, 2002).

Для рендерингу ми використовуємо утиліту 3dsmaxcmd.exe, якій потрібно тільки вказати номер рядка кінцевого растрового зображення, який буде згенерований після візуалізації. В цьому випадку задача розпаралелення полягає в тому, що ми генеруємо та передаємо на обробку множину номерів рядків кожному комп'ютеру. Сукупність таких рядків, яка залежить від числа вузлів мережі та становить підзадачу, яку буде виконувати (рендерити) окремий вузол.

Задача розподілу підзадач між процесорами в нас виконується статично, тобто кількість рядків у підзадачі обчислюється до початку рендерингу і дорівнює N/p, де N – кількість всіх рядків матриці, а p – кількість вузлів. Тоді, відповідно, на кожен вузол відправляється одна підзадача. Загалом час на виконання рендерингу всієї сцени (T_R) можна оцінити як

$$T_R = \tau \cdot (\mathrm{N/p}) + T_{P2P}, \quad (4)$$

де: τ – час виконання рендерингу одного рядка растрового зображення; T_{P2P} – час на відправлення/отримання даних між вузлами, отриманий за формулою (3).

Для визначення ефективності паралельного алгоритму існують показники прискорення та ефективності (Gergel, 2007). Якщо, наприклад, у нас є p процесорів,

тоді прискорення паралельного алгоритму (S_p) порівняно з послідовним буде

$$S_p = T_1 / T_p \,, \qquad (5)$$

де: T_1 – час виконання послідовного алгоритму; T_p – час виконання паралельного алгоритму. Ефективність експлуатації процесорів паралельним алгоритмом при вирішенні задачі визначається відношенням

$$\mathrm{E}_p = T_1 / (p \cdot T_p) \,. \qquad (6)$$

Матеріали та методи дослідження. Завдяки технічній підтримці та співпраці з Renderua (команда фахівців, що займається 3d-моделюванням) було створено бета версію програмного продукту, що покликаний об'єднати людей, які займаються 3d-дизайном в одну відкриту спільноту, і надавати обчислювальні потужності власних комп'ютерів для рендерингу створених моделей. На цьому етапі продукт працює тільки для користувачів редактора 3ds Max та його плагіну для рендерингу V-Ray і потребує фінансової підтримки для охоплення інших редакторів та удосконалення наявних алгоритмів.

На момент написання статті ця програма має такий вигляд (рис. 2).

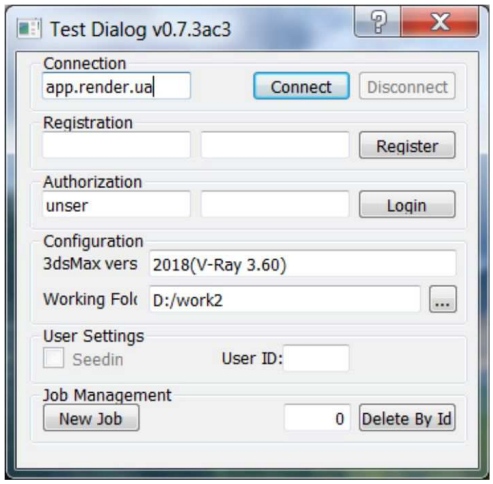

Рис. 2. Вигляд вікна задач програми клієнта

Елементи інтерфейсу дають змогу підключатись до сервера (Connect). Реєструвати нового користувача (Register) та логінитись в систему (Login). І головна можливість це додати (New Job) та видалити (Delete By Id) задачу для рендерингу. За бажанням користувач може стати учасником процесу рендерингу і надати потужності свого комп'ютера для обчислень (Seeding).

Для того, щоб розподіляти виконання задач всіх клієнтів, існує комп'ютер-сервер, який запущений під операційною системою Ubuntu та дає змогу повністю керувати процесом обчислень, а також зберігає в базу даних інформацію про весь процес рендерингу. Ще одне важливе призначення сервера в тому, що він може керувати політикою розподілу задач і потужностей комп'ютерів та обмежувати доступ для тих вузлів, які тільки користуються послугами інших вузлів, без залучення до процесу власних обчислювальних можливостей.

Результати дослідження. Було проведено дослідження ефективності рендерингу в цій програмі на прикладі моделі лісосушильної камери (рис. 3) та з використанням десяти обчислювальних вузлів. На всіх комп'ютерах було встановлено 3ds Max 2018 та V-Ray 3.60. Ці продукти можна скачати у вигляді студентських версій або безоплатно у вигляді 30-денної пробної версії. Для визначення ефективності оцінювались показники (4) і (5).

Внаслідок експерименту було визначено, що для рендерингу моделі лісосушильної камери на одному комп'ютері з розширенням 1000×1000 потрібно 620 с, а на десяти комп'ютерах (з двоядерними процесорами Intel Pentium G2020, 2.90 GHz та 4096MB RAM) потрібно 97 с. Звідси випливає, що $\mathrm{S_p} = 6{,}38$ і $\mathrm{E_p} = 0{,}638$ за формулами (5), (6). Готове зображення представлено на рис. 4.

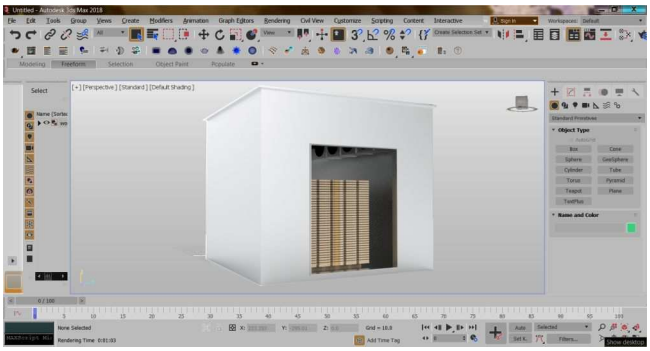

Рис. 3. Вигляд 3d-моделі лісосушильної камери

Рис. 4. Вигляд кінцевого зображення лісосушильної камери

Висновки. За результатами дослідження можна зробити такі основні висновки:

1. Проведено рендеринг 3ds Max моделі лісосушильної камери та отримано кінцеве зображення високої якості.
2. Наведено алгоритм розпаралелення рендерингу за даними для передачі підзадач обчислювальним вузлам.
3. Показано шляхи як можна прискорити наявні методи паралельного рендерингу за рахунок передавання даних, використовуючи BitTorrent протокол.
4. Представлено бета варіант програми, в якій втілена ідея рендерингу peer-to-peer, що може істотно здешевити процес паралельного рендерингу.

Перелік використаних джерел

Fletcher, Dunn. (2002). *3D math primer for graphics and game development*, by Fletcher Dunn and Ian Parberry, 280 p.

Gergel, V. P. (2007). *Teoriia i praktika parallelnykh vychislenii.* Moscow: Internet-Un-t Inform. Tekhnologii: BINOM. Lab. znanii, 423 p. [In Russian].

Kajiya, James T. (1986). *The rendering equation* (PDF), Siggraph 1986, 143. https://doi.org/10.1145/15922.15902

Protocol. (2017). *Bittorrent Protocol Specification v1.0.* Retrieved from: https://wiki.theory.org/index.php/BitTorrentSpecification

Somani, Mahesh. (2012). Bittorrent for package distribution in the enterprise, Ebay Tech Blog, 320 p. Retrieved from: https://www.ebayinc.com/stories/blogs/tech/bittorrent-for-package-distribution-in-the-enterprise

М. В. Дендюк, П. И. Рожак, Н. О. Семенишын

Национальный лесотехнический университет Украины, г. Львов, Украина

МЕТОДИКА УВЕЛИЧЕНИЯ СКОРОСТИ ПАРАЛЛЕЛЬНОГО РЕНДЕРИНГА С ПОМОЩЬЮ BITTORRENT ПРОТОКОЛА

В последнее время все чаще прибегают к использованию так называемого параллельного рендеринга, поскольку это существенно экономит время. Существует немало средств для такого вида рендеринга, но они встречают определенные ограничения. К примеру, рендеринг может осуществляться только в пределах локальной сети (напр., В V-Ray или Corona Renderer), или еще одно ограничение – рендеринг выполняется только на GPU (например. Octane Render), для этого необходимо иметь мощные видеокарты, стоимость которых может достигать нескольких тысяч долларов. Еще одно средство параллельного рендеринга – это рендер-фермы, аренда которых может стоить от нескольких долларов до нескольких тысяч долларов за сутки пользования. Здесь следует добавить еще одну проблему, а именно – ограничения на скорость передачи данных и ресурсов от одного компьютера на другой. В данной работе представлена методика и пример программы, которая позволяет обойти большинство вышеупомянутых ограничениях за счет использования подхода peer-to-peer. Данный подход является продуктивным, если осуществляется на двух уровнях. Первый уровень касается передачи данных и ускоряет этот процесс с помощью использования BitTorrent протокола. Второй уровень касается рендеринга в целом и позволяет автору 3d-сцены использовать вычислительные мощности "пиров", объединенных в единый большой кластер по принципу "ты мне, а я тебе". Исследование проведено на примере одной из самых популярных программ для 3d-моделирования 3ds Max от компании Autodesk и ее плагина для рендеринга V-Ray Chaos Group, который использует мощности CPU и пользуется огромной популярностью в мире. В качестве сцены для рендера использована лесосушильная камера.

Ключевые слова: параллельный рендеринг; BitTorrent протокол; peer-to-peer архитектура; лесосушильная камера.

M. V. Dendiuk, P. I. Rozhak, N. O. Semenyshyn

Ukrainian National Forestry University, Lviv, Ukraine

THE METHOD FOR INCREASING THE SPEED OF PARALLEL RENDERING USING THE BITTORRENT PROTOCOL

The so-called parallel rendering is currently becoming more and more popular, as it considerably saves time. There are many tools for this type of rendering, but they usually encounter certain limitations. For example, rendering can only be done within a local area network (eg V-Ray or Corona Renderer), or one more limitation is rendering only on the GPU (for example, Octane Render), for which you need to have powerful video cards that can cost several thousands of dollars. Another means of parallel rendering is a render farm, which renting can cost from a few dollars to several thousand dollars per day of use. This should include another problem, namely, the limited transfer of data and resources from one computer to another. Therefore, the article presents a methodology and example program that allows bypassing most of the above limitations by using the peer-to-peer approach. This approach is productive if implemented on two levels. The first level relates to data transfer and speeds up this process by using BitTorrent protocol. The second level relates to rendering as a whole and allows the author of the 3 d scene to use the computing power of "peers", united into a single large cluster based on the "logrolling" principle. The research was carried out on an example of a rendering model with the created wood drying chamber, prepared in one of the most popular programs for 3 d-modeling 3ds Max from Autodesk. As a renderer, the V-Ray Chaos Group plug-in was used. The authors present the program that implements the idea of rendering peer-to-peer, which can significantly reduce the cost of this process. Row-wise block-striped data decomposition algorithm for parallel rendering is also given. Acording to this algorithm every P2P rendering node receives its own portion of data to rendering. The User Interface of the main window of the client test program is presented as well. UI contains the main functionality, namely: button and field for connection to the server; buttons for registering a new user and its authorization; buttons to add and remove a new rendering task; checkbox for connecting your computer to the computational network.

Keywords: parallel rendering; BitTorrent protocol; peer-to-peer architecture; wood-drying chamber.