

Міністерство освіти і науки України
ДВНЗ «Донецький національний технічний університет»
ДНУ «Інститут модернізації змісту освіти»
Черкаський інститут пожежної безпеки імені Героїв Чорнобиля НУЦЗ
України



ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ
ТЕХНІЧНИЙ УНІВЕРСИТЕТ



МІНІСТЕРСТВО
ОСВІТИ І НАУКИ
УКРАЇНИ

«Наукові досягнення та відкриття сучасної молоді»

III Всеукраїнська наукова конференція студентів
та молодих вчених

Збірник матеріалів

29 травня 2024 року, м. Луцьк

УДК 001-053.6 Н 34

Наукові досягнення та відкриття сучасної молоді [Електронний ресурс] : зб.матер. III Всеукр. наук. конф. студ. та молодих вчених (Луцьк, 29 трав. 2024 р.) / Держ. вищ. навч. заклад «Донецький національний технічний університет». – Луцьк : ДВНЗ «ДонНТУ», 2024. – 190 с.

Збірник містить доповіді учасників III Всеукраїнської науково-практичної конференції «Наукові досягнення та відкриття сучасної молоді», які розподілені за секціями: «Комп'ютерні та технічні науки», «Економічні науки», «Цивільна та екологічна безпека».

Видання може бути корисним здобувачам вищої освіти, молодим науковцям та викладачам. Усі матеріали друкуються в авторській редакції і відповідальність за їх зміст несуть автори. Оргкомітет конференції претензії з цього приводу не приймає.

Відповідальна за випуск:

Ольга БОГОМАЗ – Голова Ради молодих вчених ДВНЗ «ДонНТУ», Ph.D, доцент, доцент кафедри природоохоронної діяльності

Рецензенти:

Віктор КОСТЕНКО – д.т.н., професор, завідувач кафедри «Природоохоронна діяльність», ДВНЗ «ДонНТУ».

Тетяна КОСТЕНКО – д.т.н., професор, заступник начальника кафедри безпеки об'єктів будівництва та охорони праці, ЧПБ «Черкаський інститут пожежної безпеки імені Героїв Чорнобиля

Організаційний комітет

Ляшок Ярослав Олександрович – д-р екон. наук, професор, в.о. ректора ДонНТУ, голова оргкомітету.

Богомаз Ольга Петрівна – Ph.D., доцент, доцент кафедри природоохоронної діяльності, голова Ради молодих вчених ДонНТУ, заступник голови оргкомітету.

Подкопаєв Сергій Вікторович – д-р техн. наук, професор, в.о. проректора з наукової роботи ДонНТУ.

Кутняшенко Олексій Ігорович – к.т.н., доцент, доцент кафедри природоохоронної діяльності.

Панкова Марія Вікторівна – Ph.D., доцент, доцент кафедри економіки підприємства.

Фоміна Олена Олександрівна – к.е.н., доцент, доцент кафедри економіки підприємства.

Чепіга Дар'я Анатоліївна – к.т.н., доцент, доцент кафедри управління гірничим виробництвом і охорони праці.

2007. URL: http://www.ittk.hu/netis/doc/ISCB_eng/02_ZKL_final.pdf (дата звернення 24.04.24)

2. Vacek, J. et al. Society, science and technology. Plzeň : University of West Bohemia in Plzeň, Faculty of Economics, KIP - Department of Management, Innovation and Projects, 1998. URL: <http://www.kip.zcu.cz/kursy/svt/> (дата звернення 24.04.24)

3. Herring, J. E. Information literacy and Science Festival. In Jems Herring's Blog. April 23, 2011. URL: <https://jherring.wordpress.com/2011/04/23/information-literacy-and-science-festival/> (дата звернення 24.04.24)

4. Klausegger, C., Sinkovics, R., Zou, H. Information overload : a cross-national investigation of influence factors and effects. In Marketing Intelligence & Planning. 2007, vol. 5. S. 691-718.

5. Britz, J. J. To Know or not to Know: A Moral Reflection on Information Poverty. In Journal of Information Science June 2004 vol. 30 no. 3. 192-204.

ОСОБЛИВОСТІ КОНТЕЙНЕРИЗАЦІЇ У СИСТЕМАХ КОМП'ЮТЕРНОЇ ГРАФІКИ

Романюк О. Н., д. т. н., проф., Завальнюк Є. К., асп., Стахов О. Я., к. т. н., ст. викл., Романюк О. В., к. т. н., доц.

Вінницький національний технічний університет

rom8591@gmail.com, qq9272627@gmail.com, aleksey.stakhov@gmail.com,
oroman@vntu.edu.ua

Вступ. Системи комп'ютерної графіки [1] характеризуються підвищеними вимогами до продуктивності візуалізації сцен [2] та до ефективності використання обчислювальних ресурсів. Для ефективного використання графічних даних і застосунків доцільним відповідно є упакування різних мультимедіа в один файл та упакування програм разом з налаштуваннями.

Аналіз особливостей контейнеризації у системах комп'ютерної графіки. Контейнеризація в комп'ютерній графіці може застосовуватись для різних об'єктів. Виокремлюються контейнеризація графічних даних, графічного програмного забезпечення, графічних обчислень, процесу рендерингу.

Контейнеризація даних [3] полягає у зберіганні зображень, відео, тривимірних моделей, що використовуються у комп'ютерній графіці, у одному файлі або структурі.

Формати файлів JPEG, PNG, TIFF дозволяють зберігати графічні дані разом з метаданими та іншою інформацією, як розмір, колірна гамма, та інше. Формати файлів AVI, MP4, MKV, дозволяють зберігати відео дані разом із аудіо, субтитрами та іншими відомостями. Вони можуть також підтримувати різні відео кодеки та параметри компресії. Зокрема, контейнер AVI [4] вміщує аудіо

та відео, які можуть синхронно відтворюватись. При цьому, підтримується значна кількість кодеків: DivX, XviD, MJPEG.

Формати 3D моделей OBJ, FBX, glTF можуть включати геометричні дані, текстури, скелетні анімації та інші властивості моделі. Зокрема, файл формату OBJ [5] зберігає інформацію про вершини полігонів поверхні об'єкта, нормалі полігонів, довільні криві поверхні. Використовуючи додаткову бібліотеку Material Template Library, також можливе збереження даних про прозорість, розсіяну, дифузну та спекулярну складові кольору, а також даних текстурних карт. Контейнери текстур дозволяють зберігати різні типи текстур, такі як зображення з різними форматами компресії та параметрами.

Контейнеризація даних у графіці допомагає управляти різними типами графічних ресурсів у зручний спосіб, сприяючи ефективнішому використанню та обробці цих даних у різних програмах та середовищах. Основними перевагами контейнеризації даних є висока ефективність збереження та передачі контенту, інтеграція різних типів даних. Недоліком є те, що надається лише спільна структура для збереження даних, що не впливає на якість, яка визначається форматами кодування окремих типів даних.

Контейнеризація програмного забезпечення полягає в упакуванні графічних застосунків разом з усіма їх залежностями та середовищем виконання у контейнери, такі як Docker або Kubernetes. Це спрощує розгортання застосунків у різних областях, включаючи хмарні сервіси, віртуалізовані середовища та сервери. Надання ізольованого середовища для графічних застосунків допомагає уникнути конфліктів між програмами та забезпечити безпеку їх виконання. Це особливо важливо, оскільки застосунки, як відеоігри чи візуалізаційні програми, можуть характеризуватися великими розривами в потужності та обсязі ресурсів. Забезпечується ефективне масштабування застосунків та оптимальне керування їх ресурсами. Також, забезпечуються стандартизація середовища виконання програм і відтворюваність результатів між різними пристроями.

Наприклад, можна запакувати засіб тривимірної графіки Blender [6] у контейнер Docker. Для цього потрібно встановити Docker на комп'ютер, у терміналі витягнути образ Blender Docker із репозиторію Docker Hub, запустити Blender у контейнері Docker, після цього можна взаємодіяти із Blender та використовувати його функціонал. В результаті, забезпечується сумісність засобу із різними конфігураціями систем. Це дозволяє групі розробників використовувати Blender із однаковим набором залежностей.

Контейнеризація обчислень полягає у ізоляції та управлінні ресурсами обчислювальних середовищ, що базуються на різних техніках паралельного обчислення (CUDA у випадку NVIDIA GPU або OpenCL для різних пристроїв) для виконання завдань комп'ютерної графіки. Тому забезпечуються паралельний запуск багатьох процесів обробки даних без взаємного впливу, управління доступом до обчислювальних ресурсів, розподілення обчислювальних завдань між різними вузлами та ресурсами, легке розгортання обчислювальних середовищ. Це спрощує розробку та тестування графічних програм.

Наприклад, можливе упакування програми CUDA [7] у контейнері Docker. Для цього потрібно встановити Docker, налаштувати драйвер Nvidia та середовище виконання, встановити середовище виконання Docker, перезапустити Docker, вибрати необхідний образ для додатка, запустити програму CUDA у Docker.

Контейнеризація процесу рендерингу відіграє важливу роль у візуалізаційних проєктах та рендерингових платформах. Зокрема, можливим є створення ізольованих середовищ для рендерингу, в яких можуть бути налаштовані конкретні версії програмного забезпечення, бібліотеки та драйвери. Це допомагає уникнути конфліктів між різними версіями програм та забезпечити відтворюваність результатів. Також, забезпечується легке розгортання рендерингових рішень у хмарних сервісах, таких як AWS ECS, Google Kubernetes Engine або Azure Kubernetes Service. Це значно полегшує масштабування рішень рендерингу для великих проєктів. Також, використання контейнерів дозволяє спростити управління обчислювальними ресурсами рендерингу. Наприклад, спрощується розподіл задач рендерингу між різними вузлами для оптимізації швидкості та ефективності. Окрім цього, контейнери дозволяють відокремлювати різні етапи процесу рендерингу, наприклад, підготовку сцени, рендеринг та пост-обробку. Це полегшує керування складними рендеринговими потоками та дозволяє швидше вносити зміни.

Висновок. Контейнеризація програмного забезпечення комп'ютерної графіки сприяє швидкому розгортанню, масштабуванню та зручному управлінню графічними застосунками. Забезпечується підвищення ефективності роботи з візуалізаційними проєктами.

ЛІТЕРАТУРА

1. Романюк О. Н., Романюк О. В., Чехместрук Р. Ю. Комп'ютерна графіка: електронний навчальний посібник. Вінниця : ВНТУ, 2023. 147 с. URL: <https://ir.lib.vntu.edu.ua/handle/123456789/37689> (дата звернення: 28.04.2024).
2. Романюк О. Н., Чорний А. В. Високопродуктивні методи та засоби зафарбовування тривимірних графічних об'єктів. Вінниця : УНІВЕСУМ-Вінниця, 2006. 190 с.
3. What Are Container File Formats (Media Containers)? *Cloudinary*. URL: <https://cloudinary.com/guides/video-formats/what-are-container-file-formats-media-containers>. (date of access: 30.04.2024).
4. AVI Format: Should You Still Use AVI? *Cloudinary*. URL: <https://cloudinary.com/guides/video-formats/avi-format-should-you-still-use-avi#:~:text=AVI%20stands%20for%20Audio%20Video,opened%20using%20Windows%20Media%20Player>. (date of access: 30.04.2024).
5. Chakravorty D. The OBJ File Format – Simply Explained. *All3DP*. URL: <https://all3dp.com/2/obj-file-format-simply-explained/>. (date of access: 30.04.2024).