

МОДЕЛІ УЗГОДЖЕНОСТІ РОЗПОДІЛЕНИХ СИСТЕМ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ

Ця робота фокусується на інтерактивних розподілених програмах із спільним віртуальним середовищем, що також відомі як розподілені системи віртуальної реальності. Такі системи ефективно використовують у застосунках для навчальних, імітаційних і розважальних цілей. Системи розподіленої віртуальної реальності характеризуються високим рівнем взаємодії між користувачами у режимі реального часу у спільному віртуальному середовищі. В статті розглянуто основні вимоги до таких систем, їхні функції та властивості. Автори аналізують наявні підходи до досягнення узгодженості в розподілених системах, а також пропонують власну модель узгодженості розподілених систем віртуальної реальності, що дає змогу досягти ефективного балансу між узгодженістю та швидкістю відтворення станів.

Ключові слова: розподілені системи, віртуальна реальність, узгодженість, мультиплеєрні ігри, інтерактивність.

Вступ

Предметом особливого інтересу є розроблення розподілених систем віртуальної реальності, що довели свою ефективність у багатьох сферах освіти, науки, техніки, розваг та медицини. Для взаємодії великої кількості користувачів потрібно розробляти гнучкі й масштабовані розподілені системи віртуальної реальності. Такі системи дають змогу декільком користувачам одночасно занурюватися у віртуальну реальність у режимі реального часу та взаємодіяти один з одним.

Основна мета роботи – розроблення ефективної моделі узгодженості розподілених систем віртуальної реальності. Автори аналізують основні характеристики таких систем, відомі підходи до досягнення узгодженості в розподілених системах, а також пропонують власну модель оптимальної узгодженості розподілених систем віртуальної реальності.

Характеристика розподілених систем віртуальної реальності

Віртуальна реальність – це комп'ютерне реалістичне моделювання світу, яке часто називають віртуальним середовищем, яке дає змогу користувачам взаємодіяти з навколишнім середовищем у реальному часі. Віртуальне середовище – це сукупність віртуальних об'єктів із певними наборами атрибутів, що визначають властивості та поведінку кожного об'єкта, утворюючи стан об'єкта. Стан віртуального середовища – це сукупність станів усіх віртуальних об'єктів, що містяться у віртуальному середовищі [5].

Розподілена система віртуальної реальності – це система віртуальної реальності, яка дає змогу віддаленим користувачам взаємодіяти у спільному віртуальному середовищі. Зазвичай користувачі таких систем приєднуються з різних географічних місць за допомогою широкопasmової мережі, але це також можливо і з використанням локальних мереж.

Розподілені системи віртуальної реальності – це підклас розподілених систем, тому вони успадковують усі їхні властивості [6]:

- Немає глобальних фізичних годинників. Системні процеси асинхронні, що вводить у систему розподільний елемент.
- Немає спільної пам'яті. Процеси мають спілкуватися, використовуючи обмін повідомленнями.
- Географічне відокремлення. Процеси можуть працювати як у локальній мережі (LAN), так і в широкій мережі (WAN).
- Автономність та неоднорідність. Процесори слабо зв'язані, це означає, що вони можуть працювати з різною швидкістю та мати різні операційні системи.

Розподілені системи віртуальної реальності також мають свої унікальні характеристики [4]:

- Обчислення у режимі реального часу, тому обчислювальні вузли повинні мати можливість обробляти велику кількість операцій над інформацією за короткий час.
- Специфічний характер розподілених обчислень. Розрахунки відбуваються здебільшого навколо розрахунку глобального стану середовища.
- Вимога синхронізації годинника. Процеси не мають загального фізичного годинника, але вони мають синхронізувати свої локальні годинники між собою.

Моделі узгодженості розподілених систем

Важливою проблемою у розподілених системах є реплікація даних. Дані, як правило, реплікують для підвищення надійності або підвищення продуктивності. Однією з найважливіших проблем є узгодженість реплік. Коли одна копія оновлюється, ми повинні забезпечити оновлення інших, інакше репліки вже не будуть однаковими.

Модель узгодженості – це контракт між процесами та сховищем даних. У ньому йдеться про те, що якщо процеси погоджуються виконувати деякі правила, то сховище даних гарантує, що працюватиме належним чином [8].

Реплікація даних створює проблеми узгодженості, які не можна ефективно вирішити загальним способом. Тільки якщо ми послабимо узгодженість, то можливо буде сподіватися на досягнення ефективних рішень. На жаль, також немає загальних правил послаблення узгодженості: те, що можна послабити, сильно залежить від конкретних застосувань.

У застосунках є різні способи вказати, який рівень невідповідності вони можуть допустити. Г. Ю і А. Вахдат застосовують загальний підхід, розрізняючи три незалежні підходи для визначення невідповідностей: відхилення числових значень між репліками, відхилення усталеності між репліками та відхилення щодо впорядкування операцій оновлення [9].

Вимірювання невідповідності з погляду числових відхилень можуть використовувати програми, для яких дані мають числову семантику. Одним із можливих прикладів є реплікація даних, що відповідають за позицію об'єктів у віртуальному середовищі. У цьому випадку програма може вказати, що дві копії не повинні відхилитися позицією більше ніж на 1 метр, що було б абсолютним числовим відхиленням. Крім того, можна зазначити відносне числове відхилення, вказуючи, що дві копії повинні відрізнятись не більш ніж, наприклад, 1 %. В обох випадках ми побачимо, що якщо позиція зміниться і репліки негайно оновляться без порушення вказаних числових відхилень, репліки все одно вважатимуться узгодженими.

Відхилення стійкості стосуються останнього разу, коли репліку було оновлено. Для деяких застосунків можна допустити, що репліка надає старі дані доти, доки вони не занадто старі. Наприклад, звіти про погоду зазвичай залишаються досить точними протягом певного часу, скажімо, декількох годин. У таких випадках основний сервер може отримувати своєчасні

оновлення, але може вирішити поширювати оновлення до реплік лише раз у годину [8].

Також існують класи застосунків, у яких впорядкування оновлень може відрізнятись у різних репліках, допоки відмінності залишаються обмеженими. Оновлення застосовують орієнтовно до локальної копії, очікуючи глобальної згоди від усіх реплік. Як наслідок, деякі оновлення, можливо, доведеться скасувати та застосувати в іншому порядку, перш ніж вони стануть постійними.

Послідовна узгодженість – важлива модель узгодженості, орієнтована на дані. Вперше її визначив Л. Лампорт у контексті спільної пам'яті для багатопроцесорних систем. За такого підходу сховище даних є послідовно узгодженим, якщо воно відповідає такій умові: «Результат будь-якого виконання такої самий, якби операції (читання і запис) усіма процесами в сховищі даних були виконані в певному послідовному порядку, і операції кожного окремого процесу з'являються в цій послідовності в порядку, що визначений його програмою» [7].

Модель причинно-наслідкової послідовності являє собою послаблення послідовної послідовності, оскільки вона розрізняє події, які потенційно можуть бути причинно-наслідковими, і ті, які не є такими. Якщо подія event2 викликана або впливає на попередню подію event1, причинність вимагає, щоб усі інші процеси спочатку бачили event1, а потім event2 [3].

Якщо два процеси одночасно записують два різні елементи даних, вони не є причинно-наслідковими. Кажуть, що операції, які не є причинно-наслідковими, є паралельними. Щоб сховище даних вважали причинно-узгодженим, воно має відповідати такій умові: «Записи, які є потенційно причинно-наслідковими, повинні розглядатися всіма процесами в одному порядку. Паралельне записування може спостерігатись в різному порядку на різних машинах» [8].

Остаточна узгодженість – це модель, за якої узгодженість досягається, якщо тривалий час не відбувається оновлення даних. Отже, сховища даних у певний момент часу стають узгодженими, тобто мають властивість, що за відсутності оновлень всі репліки сходяться до однакових копій одна одної. По суті ця модель потребує лише того, щоб оновлення гарантовано поширювались у всіх репліках.

Клієнто-орієнтована узгодженість забезпечує гарантію для одного клієнта щодо узгодженості доступу до сховища даних цим клієнтом. Жодних гарантій щодо одночасного доступу різних клієнтів не дається.

Перша клієнто-орієнтована модель узгодженості – це монотонне зчитування. Кажуть, що сховище даних забезпечує монотонне зчитування, якщо виконується така умова: «Якщо процес зчитує значення елемента даних x , будь-яка послідовна операція читання x цим процесом завжди повертає те саме значення або більш нове значення» [8]. Монотонне зчитування послідовності гарантує, що якщо процес побачив значення X у час T , він ніколи не побачить старішу версію X пізніше.

У багатьох ситуаціях важливо, щоб операції запису поширювались у правильному порядку на всі копії сховища даних. Ця властивість виражається в узгодженості монотонних записів. У сховищі даних, що відповідає монотонному запису, виконується умова: операція запису процесом на елементі даних X завершується перед будь-якою послідовною операцією запису на X тим самим процесом. Іншими словами, операція запису на копії елемента X виконується лише в тому випадку, якщо ця копія була оновлена за допомогою будь-якої попередньої операції запису, яка могла відбуватися на інших копіях X . Якщо потрібно, нове записування даних *повинно* дочекатися закінчення старих.

Кажуть, що сховище даних забезпечує узгодженість читання-запису, якщо ефект операції запису процесом на елемент даних X завжди буде видно послідовною операцією зчитування X тим самим процесом. Іншими словами, операція запису завжди завершується перед послідовною операцією читання тим самим процесом, незалежно від того, де відбувається ця операція зчитування [8].

Остання модель клієнто-орієнтованої узгодженості полягає в тому, що оновлення поширюються у результаті попередніх операцій зчитування. Кажуть, що сховище даних забезпечує узгодженість запису-читання, якщо виконується таке: операція запису процесом на елементі даних X після попередньої операції зчитування на X тим самим процесом гарантовано відбуватиметься за тим самим чи більш новим значенням X , яке було прочитано. Іншими словами, будь-яка послідовна операція запису процесом на елементі даних X буде виконуватися на копії X , яка є актуальною зі значенням, яке було останнім прочитане процесом.

Модель оптимальної узгодженості

Важливою характеристикою розподілених систем віртуальної реальності є високий рівень інтерактивності між користувачами в реальному

часі, що потребує швидкого відображення віртуального середовища.

Для досягнення повної узгодженості можна було б передавати всю інформацію про середовище для кожного відображення кадру, але хоч це і можливо було б зробити в умовах використання LAN у деяких випадках, в більшості випадків користувачі географічно віддалені один від одного, а отже відправка даних займатиме забагато часу. Наприклад, для військових симуляцій є необхідність мати частоту оновлення кадрів щонайменше 60 Hz. Отже, розподілені системи віртуальної реальності мають досягати послабленої узгодженості з урахуванням вимог до швидкості отримання та оброблення даних.

Віртуальне середовище являє собою сукупність віртуальних об'єктів. Кожен об'єкт має певний набір атрибутів, що визначають властивості та поведінку об'єкта, утворюючи стан об'єкта. Стан віртуального середовища – це сукупність станів усіх віртуальних об'єктів, що містяться у віртуальному середовищі. Тобто для досягнення узгодженості стану віртуального середовища необхідно досягти узгодженості станів об'єктів цього середовища.

Основна ідея підходу полягає в тому, щоб мінімізувати кількість атрибутів, які необхідно узгоджувати між процесами. Якщо існує така функція, що для деякого набору атрибутів об'єкта може дати результатом наближене значення з певною точністю до іншого атрибута, то такий атрибут може бути ігнорованим при узгодженні. Наприклад, отримання позиції та повороту об'єкта, що рухається, потребує великої кількості відправлених даних кожного кадру, що, зважаючи на можливі затримки мережі, призведе до не найкращих результатів симуляції. Можливим варіантом буде використати атрибути швидкості та прискорення для прорахування позицій та поворотів із певною точністю. Потрібно зазначити, що цей підхід потребуватиме використання методів прогнозування станів, як-от числення координат тощо.

Далі розробник може виявити, що швидкість і прискорення об'єкта залежать від станів натискання клавіш клавіатури чи мишки користувача. Таким способом можливо буде уникнути вимоги узгодженості атрибутів швидкості та прискорення, і необхідно буде узгоджувати лише стани пристроїв введення користувачів.

Для наших цілей зручно організувати дані в ієрархічній структурі, до якої матимуть доступ користувачі у віртуальному середовищі, а саме у формі дерева – ациклічного орієнтованого графу із випадковою кількістю вершин. Коренем

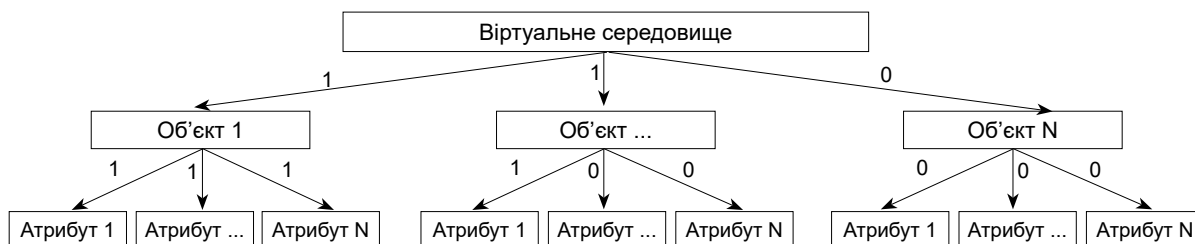


Рисунок. Дерево узгодженості об'єктів та їхніх атрибутів

дерева є рівень, що охоплює весь віртуальний світ. На наступному рівні світ розбивається на ієрархію вузлів, що відповідають за конкретні об'єкти віртуального середовища. Далі, об'єкти розбиваються на вузли, що відповідають за їхні атрибути. Кожному ребру приписується значення 1, якщо об'єкт або атрибут вимагають узгодженості, та 0, якщо ні.

Для синхронізації годинників процесів використовують алгоритм Крістіана та Network Time Protocol. Усі події у віртуальному середовищі пов'язані з певним моментом часу. Глобальний стан віртуального середовища розраховують на основі конкретного моменту часу.

Ця модель дає змогу зменшити кількість атрибутів стану об'єкта, необхідних для реплікації та підтримки узгодженості, що допомагає досягти компромісу між узгодженістю станів і швидкістю відтворення станів.

Висновки

У цій роботі було проаналізовано основні підходи до досягнення узгодженості в розподілених системах. Також наведено основні характеристики розподілених систем віртуальної реальності та їх вплив на задачу досягнення узгодженості станів. Запропонований авторами підхід дає змогу мінімізувати кількість даних, які необхідно передати іншим процесам для коректної реплікації з погляду узгодженості станів.

Модель узгодженості, запропоновану в цій статті, нині застосовують здебільшого в розподілених інтерактивних застосунках, що були розроблені одним із авторів та широко використовуються у світовій глобальній мережі: Dragon Sim Online [2] та Cat Sim Online [1].

Список літератури

1. Cat Sim Online [Electronic resource]. – 2019. – Mode of access: <https://play.google.com/store/apps/details?id=com.turborocketgames.catsim>.
2. Dragon Sim Online [Electronic resource]. – 2019. – Mode of access: <https://play.google.com/store/apps/details?id=com.turborocketgames.dragonsim>.
3. Hutto P. W. Slow Memory: Weakening Consistency to Enhance Concurrency in Distributed Shared Memories / P. W. Hutto, M. Ahmad // The 10th International Conference on Distributed Computing Systems: Paris, France, May 28 – June 1, 1990. – P. 302–311.
4. Kharitonov V. Y. A consistency model for distributed virtual reality systems / V. Y. Kharitonov // Forth International Conference on Dependability of Computer Systems DepCoS-RELCOMEX. – Brunow, Poland, 2009, June 30 – July 2, 2009. – Pp. 271–278.
5. Kharitonov V. Y. An Approach to Consistent Displaying of Virtual Reality Moving Objects / V. Y. Kharitonov // Third International Conference on Dependability of Computer Systems DepCoS-RELCOMEX. Brunow, Poland, 2008, June 26 – June 28. – Pp. 390–397).
6. Kshemkalyani A. Distributed Computing: Principles, Algorithms, and Systems / A. Kshemkalyani, M. Singhal. – New York : Cambridge University Press, 2011. – 756 p.
7. Lamport L. How to Make a Multiprocessor Computer That Correctly Executes Multiprocess Programs / Leslie Lamport // IEEE Transactions on Computers C-28. – 1979. – No. 9. – P. 690–691.
8. Tanenbaum A. Distributed Systems: Principles and Paradigms / A. Tanenbaum, S. Maarten. – Scotts Valley, California : CreateSpace Independent Publishing Platform, 2016. – 702 p.
9. Yu H. Design and Evaluation of a Conit-Based Continuous Consistency Model for Replicated Services / H. Yu, A. Vahdat // ACM Transactions on Computer Systems (TOCS). – 2002. – No. 20 (3). – P. 239–282.

References

- Cat Sim online: Play with cats – Apps on Google Play. (2019). Google Play. Retrieved from <https://play.google.com/store/apps/details?id=com.turborocketgames.catsim>.
- Dragon Sim online: Be a dragon – Apps on Google Play. (2019). Google Play. Retrieved from <https://play.google.com/store/apps/details?id=com.turborocketgames.dragonsim>
- Hutto, P. W., & Ahmad, M. (1990). Slow Memory: Weakening Consistency to Enhance Concurrency in Distributed Shared Memories. In *The 10th International Conference on Distributed Computing Systems*. Paris, France, May 28 – June 1, 1990 (pp. 302–311).
- Kharitonov, V. Y. (2009). A consistency model for distributed virtual reality systems. In *Forth International Conference on Dependability of Computer Systems DepCoS-RELCOMEX*. Brunow, Poland, 2009, June 30–July 2 (pp. 271–278).
- Kharitonov, V. Y. (2008). An Approach to Consistent Displaying of Virtual Reality Moving Objects. In *Third International*

- Conference on Dependability of Computer Systems Dep-CoS-RELCOMEX*. Brunow, Poland, 2008, June 26 – June 28 (pp. 390–397).
- Kshemkalyani, A. D., & Singhal, M. (2011). *Distributed computing principles, algorithms and systems*. Cambridge University Press.
- Lampert, L. (1979). How to make a multiprocessor computer that correctly executes Multiprocess programs. *IEEE Transactions on Computers*, C-28(9), 690–691. <https://doi.org/10.1109/tc.1979.1675439>
- Tanenbaum, A. S., & Steen, M. V. (2016). *Distributed systems: Principles and paradigms*. Createspace Independent Publishing Platform.
- Yu, H., & Vahdat, A. (2002). Design and evaluation of a conit-based continuous consistency model for replicated services. *ACM Transactions on Computer Systems (TOCS)*, 20 (3), 239–282. <https://doi.org/10.1145/566340.566342>

V. Duchenchuk, V. Boublik

CONSISTENCY MODELS FOR DISTRIBUTED VIRTUAL REALITY SYSTEMS

This paper focuses on interactive distributed software with a shared virtual environment, also known as distributed virtual reality systems, which allows remote users interaction within a shared virtual environment. Usually users of such systems are connected from different geographical places by means of broadband network, but it is also possible to communicate within local networks. Such systems are effectively used in applications for educational, simulation and entertainment purposes. Distributed virtual reality systems are characterized by a high level of interaction between users in real time in a shared virtual environment. They enable multiple users concurrently immersed in virtual reality to simultaneously interact with each other.

Distributed virtual reality systems are a subclass of distributed systems, so they inherit all of their properties such as the absence of global physical clock and shared memory, as well as autonomy and heterogeneity of a system. Distributed virtual reality systems also have their own unique characteristics such as real-time computations, specific nature of distributed computations and clock synchronization requirement. Extensive computer networks involving rigorous numbers of users in man-computer interaction demand high level of flexibility and scalability.

The paper considers the main requirements for such systems, their functions and properties. The authors analyze the existing approaches to achieving consistency in distributed systems such as spatial consistency, sequential consistency, causal sequence model and also eventual consistency model. Some models of client-centric consistency have also been analyzed.

Achievement of the necessary level of effectiveness depends on the policy of data replication during the whole performance. One of the major problems in this case is coherence of replicas. When one copy is updated, the others should be updated, too; otherwise replicas will not be identical anymore. Appropriate coherence model should provide a contract between processes and a storage of data. In other words, if processes agree to carry out some rules, then the storage of data guarantees consistent processing.

The authors offer their own model of consistency of distributed virtual reality systems, which allows to achieve an effective balance between consistency and the speed of states simulation.

Keywords: distributed systems, virtual reality, consistency, multiplayer games, interactive activity.



Creative Commons Attribution 4.0 International License (CC BY 4.0)

Матеріал надійшов 17.06.2020