

СНИЖЕНИЕ СЕТЕВОГО ТРАФИКА В ФИЗИЧЕСКИХ СИМУЛЯЦИЯХ НА БАЗЕ UNREAL ENGINE 5

Згурский Е.О., магистрант,

Матвеев С.П., магистрант,

Ефромеев Н.М., ст. преподаватель,

ФГАОУ ВО "МГТУ "СТАНКИН", г. Москва, Россия

Аннотация. В статье исследуется проблема высокого потребления сетевого трафика в многопользовательских играх с интенсивным физическим взаимодействием на движке Unreal Engine 5. Стандартная модель репликации состояния генерирует накладные расходы, линейно растущие с числом динамических объектов. В качестве решения предложена архитектура репликации входов, основанная на детерминизме физического движка Chaos Physics. Показаны результаты проведенного сравнительного эксперимента.

Ключевые слова: Unreal Engine 5, сетевая физика, детерминизм, репликация состояния, сетевой трафик.

Современный человек живет в мире высоких информационных технологий [6]. Первоначально компьютеры предназначались главным образом для выполнения сложных математических расчетов [1, с. 5]. Но в домашних условиях компьютер давно используется для развлечения, причем не только детьми, но и взрослыми. Индустрия программного обеспечения ежедневно поставляет на рынок программных продуктов десятки, если не сотни новых программ, приложений, систем либо очередных модификаций, версий уже существующих программных средств [2, с. 111].

Игры занимают свое важное место и для отдыха, и для обучения. Однако написание игры с самого нуля – это сложная задача, так как требует большого количества времени на программирование. При этом значительная доля всей работы производится вручную [3]. Для облегчения данной работы были

созданы так называемые игровые движки, в которых реализована основная часть игры [4]. Игровой движок – базовое программное обеспечение компьютерной игры, которое пригодно для повторного использования и расширения при создании последующих игр [5].

Сетевая синхронизация физически корректных взаимодействий представляет одну из наиболее ресурсоёмких задач при разработке многопользовательских игр. Стандартная клиент-серверная модель Unreal Engine – репликация состояния – требует регулярной рассылки обновлений положения, вращения и скоростей каждого динамического объекта, причём стоимость этих операций линейно растёт с числом объектов.

Физический движок Chaos Physics, введённый в Unreal Engine 5 на смену PhysX, предоставляет асинхронный режим симуляции с фиксированным шагом времени. Это открывает возможность детерминированных вычислений – состояния, при котором идентичные начальные условия и последовательности входов производят побитово совпадающие результаты на разных машинах. Достижение детерминизма позволяет перейти к репликации входов: сервер транслирует только пользовательские команды и дискретные события, а клиенты воспроизводят идентичную симуляцию локально, без постоянного получения корректирующих обновлений состояния.

Были рассмотрены две основные архитектуры синхронизации физического состояния объектов в сетевой среде: базовая архитектура, основанная на передаче состояния объектов, и альтернативный подход, основанный на передаче пользовательского ввода.

Базовая архитектура предполагает, что сервер выполняет авторитетный расчёт физического взаимодействия объектов и периодически передаёт клиентским приложениям полное состояние каждого объекта, включая его положение, ориентацию, линейную и угловую скорости. Клиентская часть в такой схеме играет пассивную роль: она принимает полученные данные и выполняет интерполяцию между последовательными сетевыми пакетами для сглаживания движения. Основным недостатком данного подхода является

избыточность передаваемой информации, поскольку обновления отправляются даже для объектов, движение которых происходит по инерции и может быть достаточно точно предсказано на стороне клиента.

В качестве альтернативы рассмотрена архитектура, основанная на передаче пользовательского ввода. В данном случае как сервер, так и клиентские приложения выполняют идентичную модель физического моделирования. Сервер передаёт только управляющие воздействия пользователя – например, нажатия клавиш или применение сил – а также дискретные события, такие как взрыв или создание нового объекта. Клиент, применяя полученные данные в том же кадре моделирования, воспроизводит результат вычислений локально, что позволяет отказаться от прямой передачи координат и скоростей объектов.

Реализация такого подхода требует обеспечения строгого детерминизма вычислений физического движка Chaos. Для этого физическая симуляция была отвязана от частоты кадров рендеринга и вынесена в отдельный вычислительный цикл с фиксированным шагом моделирования. В проведённом эксперименте частота симуляции составляла 60 Гц, что соответствует шагу интегрирования $1/60$ секунды. Такой режим гарантирует одинаковое количество и последовательность шагов симуляции на всех вычислительных узлах независимо от их производительности.

Дополнительной проблемой является детерминизм операций с плавающей запятой. Некоторые инструкции процессора, например Fused Multiply-Add (FMA), могут давать минимально отличающиеся результаты на разных архитектурах центральных процессоров. Для устранения этого эффекта использовался строгий режим вычислений с плавающей запятой, отключающий нестабильные оптимизации компилятора. Кроме того, вычисления внутри решателя физического движка были приведены к явным и воспроизводимым последовательностям операций.

Особое внимание также уделялось порядку обработки объектов и вопросам многопоточности. В стандартной реализации Chaos порядок обхода

объектов и применения ограничений может изменяться с целью повышения производительности. В детерминированной версии алгоритма была введена явная сортировка объектов по уникальному идентификатору, а параллельные вычисления, способные приводить к состояниям гонки данных, синхронизированы или перенесены в основной поток выполнения.

Несмотря на принятые меры, вероятность рассинхронизации полностью исключить невозможно. Для её обнаружения сервер периодически, с интервалом около 800 миллисекунд, передаёт клиентским приложениям контрольную сумму состояния ключевых объектов. Контрольная сумма формируется как хеш позиций и скоростей приблизительно 10% физических тел сцены. В случае обнаружения расхождения сервер принудительно отправляет клиенту полный снимок состояния рассинхронизированных объектов. Низкая частота подобных коррекций используется в качестве интегрального показателя качества достигнутого детерминизма вычислений.

Тестирование проводилось в контролируемой среде: сервер и клиент развёрнуты на отдельных машинах в локальной сети. Конфигурация стенда: Intel Core i9-7980Xe, 64 ГБ DDR4, Windows 10, MSVC 19.34, Unreal Engine 5.5.4, сборка Shipping. Сбор метрик выполнялся инструментами NetProfile/Stat Net и внешними пакетными анализаторами.

Отслеживались четыре метрики: средний исходящий трафик сервера (байт/с), среднее число пакетов в секунду, частота коррекций рассинхронизации (корр./мин) и средняя позиционная ошибка (юниты). Каждый сценарий выполнялся 5 минут с прогревом 30 с; фиксированный шаг физики – 60 Гц.

Сценарий А (высокая плотность). 100 физических объектов (кубы) с одновременным включением физики и коллизий, создающим интенсивные столкновения (рис. 1). Данный сценарий моделирует пиковую нагрузку на физическую систему, характеризующуюся высокой плотностью столкновений и значительным количеством вычислений, связанных с разрешением контактов.

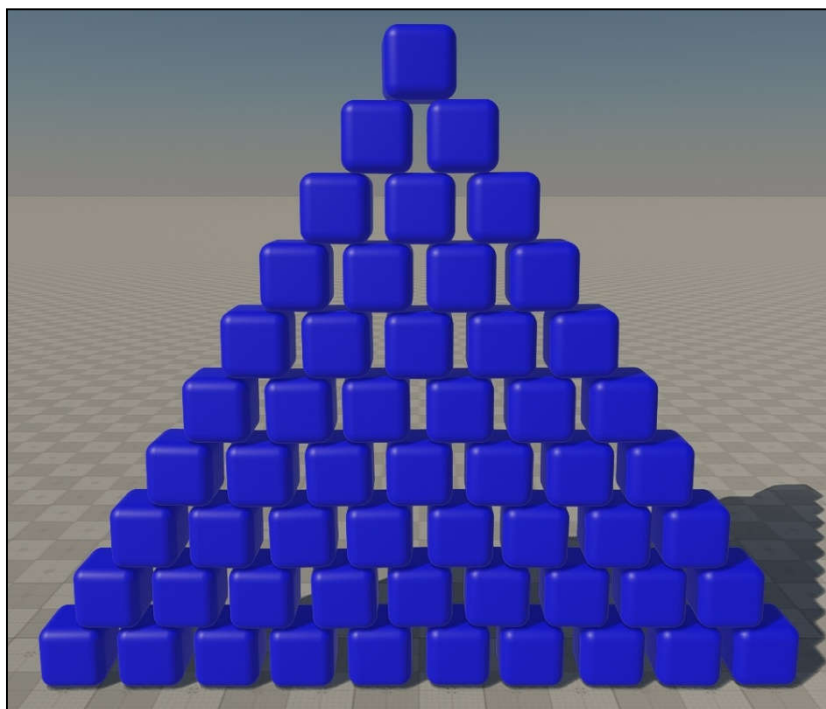


Рис. 1. Стартовое состояние: в объектах отключена физика

Этап активации коллизий физических объектов показан на рис. 2. На иллюстрации красным цветом выделены объекты, для которых выполняется обновление положения с учётом столкновений с ближайшими телами, а жёлтым цветом обозначены точки контакта между взаимодействующими объектами.

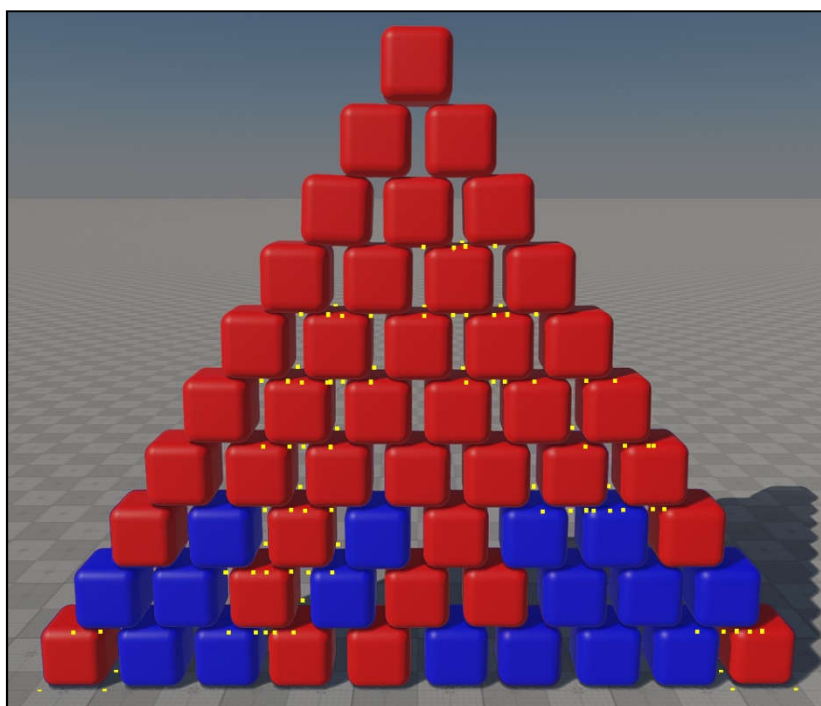


Рис. 2. Состояние сцены с объектами при включении физики

Сценарий В (умеренная плотность). 20 объектов (кубы), падающих с различных высот; менее интенсивное взаимодействие.

В таблице приведено сравнение базовой архитектуры и предлагаемого подхода по ключевым сетевым и точностным метрикам (значения усреднены за 5 минут симуляции).

Таблица

Сравнение архитектур синхронизации

Метрика	Базовая	Предлагаемая	Улучшение
Сценарий А – высокая плотность (100 объектов)			
Исходящий трафик сервера, байт/с	118 400	48 600	58,9%
Пакеты в секунду	340	145	57,4%
Частота коррекций, корр./мин	постоянные	1,8	—
Средняя позиционная ошибка, юниты	2,3	0,08	96,5%
Сценарий В – умеренная плотность (20 объектов)			
Исходящий трафик сервера, байт/с	28 600	16 900	40,9%
Пакеты в секунду	120	65	45,8%
Частота коррекций, корр./мин	постоянные	0,4	—

Средняя позиционная ошибка, юниты	1,2	0,03	97,5%
-----------------------------------	-----	------	-------

В сценарии А (100 объектов) снижение исходящего трафика составило 58,9%, число пакетов сократилось на 57,4%. Высокая экономия обусловлена тем, что при максимальной плотности столкновений базовый алгоритм вынужден реплицировать большой объём промежуточных состояний, тогда как в детерминированной архитектуре все взаимодействия воспроизводятся локально без передачи по сети.

В сценарии В (20 объектов) экономия составила 40,9% по трафику и 45,8% по числу пакетов. Меньший выигрыш закономерен: при небольшом числе объектов накладные расходы репликации состояния ниже, а фиксированные издержки синхронизации входов составляют большую долю от общего потока.

Частота коррекций в предложенной архитектуре крайне низка (1,8 корр./мин в сценарии А, 0,4 корр./мин в сценарии В), что подтверждает высокую надёжность достигнутого детерминизма. Позиционная ошибка до момента коррекции не превышает 0,08 юнита, что визуально незаметно для пользователей. Эффективность подхода масштабируется нелинейно: прирост числа объектов с 20 до 100 увеличивает экономию трафика с 40,9 до 58,9%, указывая на растущее относительное преимущество детерминированной модели в высоконагруженных сценах.

Проведённое исследование показывает, что детерминированная модель симуляции Chaos Physics может быть более эффективным инструментом снижения сетевого трафика в многопользовательских играх. Переход от репликации состояния к репликации входов обеспечил сокращение исходящего трафика сервера на 58,9% в сценариях с высокой плотностью объектов и на 40,9% в условиях умеренной нагрузки. Надёжность детерминизма подтверждается низкой частотой рассинхронизаций (не более 1,8 корр./мин в пиковых условиях) и пренебрежимо малой позиционной ошибкой.

Вместе с тем следует отметить ряд ограничений предложенного подхода. Реализация требует глубокого вмешательства в исходный код физического движка и значительно сложнее стандартной репликации. Детерминизм чувствителен к платформе и настройкам компилятора, что усложняет кроссплатформенную разработку. Наконец, данная модель не устраняет проблему сетевой задержки ввода, поскольку она предполагает получение серверных данных перед выполнением очередного шага моделирования.

Литература

1. Ефромеев, Н. М. Информатика и информационно-телекоммуникационные технологии / Н. М. Ефромеев, Е. В. Ефромеева. – Москва : Московский государственный технологический университет "СТАНКИН", 2015. – 175 с. – ISBN 978-5-7028-0673-0. – EDN STTOAP.

2. Ефромеев, Н. М. Информатика и информационно-телекоммуникационные системы / Н. М. Ефромеев, Е. В. Ефромеева. – Москва : Московский государственный технологический университет "СТАНКИН", 2018. – 167 с. – ISBN 978-5-7028-0530-6. – EDN AYZAXQ.

3. Ефромеева, Е. В. Автоматизированная система для управления работой по поиску информации на основе анализа обработанных обращений / Е. В. Ефромеева, И. А. Смирнова // Вестник МГТУ "Станкин". – 2019. – № 1(48). – С. 118-121. – EDN ECATEQ.

4. Кривошеин, А. Е. Дополнительные модули базового программного обеспечения для разработки компьютерных игр / А. Е. Кривошеин, Н. М. Ефромеев // Образование России и актуальные вопросы современной науки : Сборник статей VI Всероссийской научно-практической конференции, Пенза, 22–23 мая 2023 года / Под научной редакцией П.А. Гагаева, Е.П. Белозерцева. – Пенза: Пензенский государственный аграрный университет, 2023. – С. 257-262. – EDN WLCEFB.

5. Кривошеин, А. Е. Минимальный состав базового программного обеспечения для разработки компьютерных игр / А. Е. Кривошеин, Н. М.

Ефромеев // Современные технологии в российской и зарубежных системах образования : Сборник статей XII Международной научно-практической конференции, Пенза, 11–12 апреля 2023 года / Под редакцией Ф.Е. Удалова, В.В. Бондаренко, В.В. Полукарова. – Пенза: Пензенский государственный аграрный университет, 2023. – С. 180-183. – EDN JHLVNV.

6. Сабинская, Е. А. Исследование сетевого периметра московских университетов на наличие угроз информационной безопасности / Е. А. Сабинская, Н. М. Ефромеев, Е. В. Ефромеева // Информационные технологии и автоматизация управления : Материалы XI Всероссийской научно-практической конференции студентов, аспирантов, работников образования и промышленности, Омск, 29–30 мая 2020 года. – Омск: Омский государственный технический университет, 2020. – С. 174-180. – EDN AMAIEG.