

Параллельная симуляция. Параллельная дискретная симуляция

Курс «Программное моделирование вычислительных
систем»

Григорий Речистов
grigory.rechistov@phystech.edu

27 апреля 2015 г.

1 Идея PDES

2 Консервативные схемы

3 Оптимистичные схемы

На прошлых лекциях

- Моделирование многих агентов, работающих асинхронно — DES в один поток
- Параллельное моделирование исполняющих устройств — процессоров

Вопросы

- Что такое атомарная инструкция?

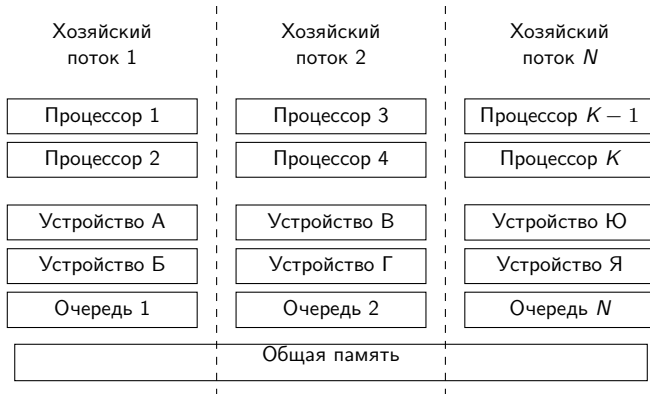
Вопросы

- Что такое атомарная инструкция?
- Что такое инструкция-барьер?

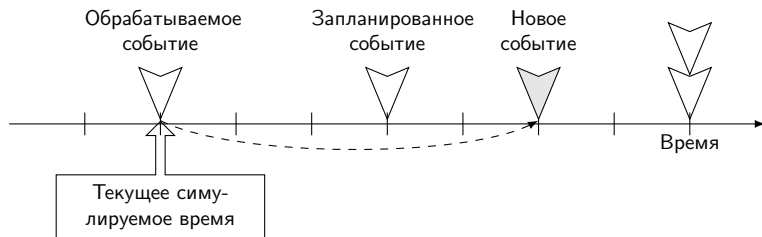
Вопросы

- Что такое атомарная инструкция?
- Что такое инструкция-барьер?
- Верно ли, что любая атомарная инструкция является барьером?

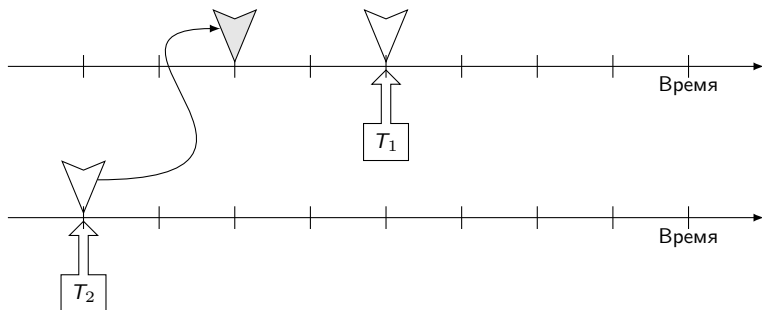
Общая схема моделируемой системы



DES



Наивный PDES



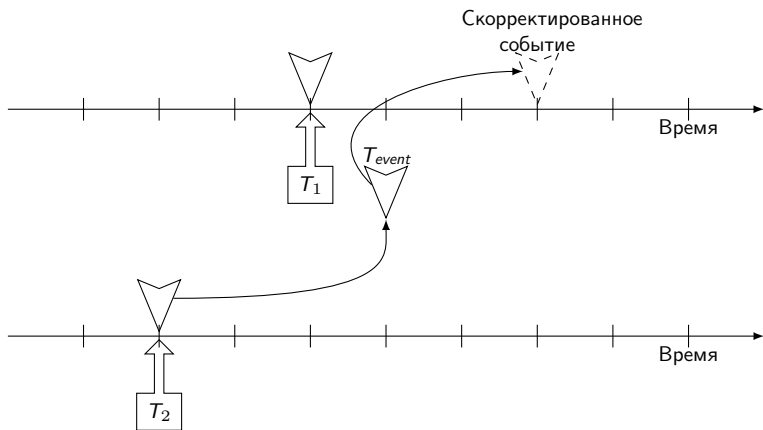
Проблемы

- Нарушение каузальности (причинно-следственной связи)
- Недетерминизм модели
- Эффект ускорения от параллелизации не гарантирован

Как детектировать нарушения

- При пересылке сообщения добавлять к нему метку локального симулируемого времени
- По получении проверять метку и корректировать точку создания события
- При отрицательном значении корректировки — бить тревогу

Метки времени



Консервативно или оптимистично?

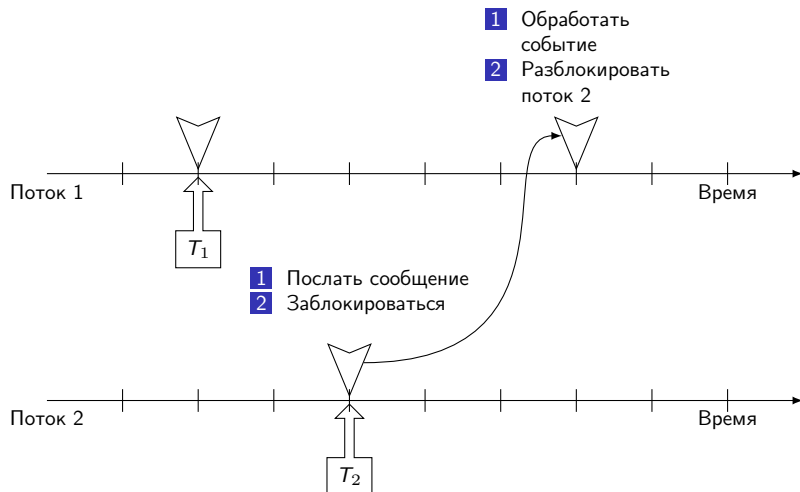
Как избавиться от появления/последствий нарушений?

- 1 Предотвращать их возникновение, или
- 2 Подавлять их вредное проявление

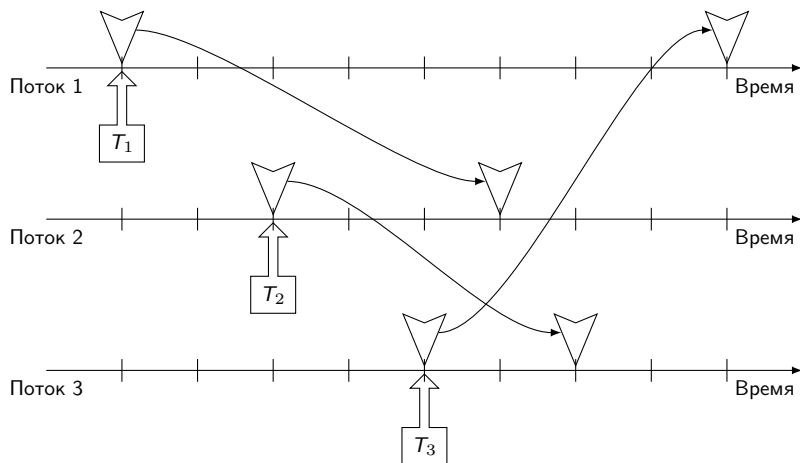
Консервативная схема

- При посылке сообщения блокировать отправителя до тех пор, пока получатель не обработает связанного события
- Не даём «быстрым» потокам продвигаться через этапы коммуникации

Консервативная схема



Взаимоблокировка



Разрушение взаимоблокировок

При детектировании ситуации дедлока безопасно освободить один поток

- Лучший выбор — очередь с наименьшим значением симулируемого времени
- Система может оказаться в ситуации, когда большую часть времени $K - 1$ потоков стоят \rightarrow выигрыша в скорости нет

Пустые сообщения

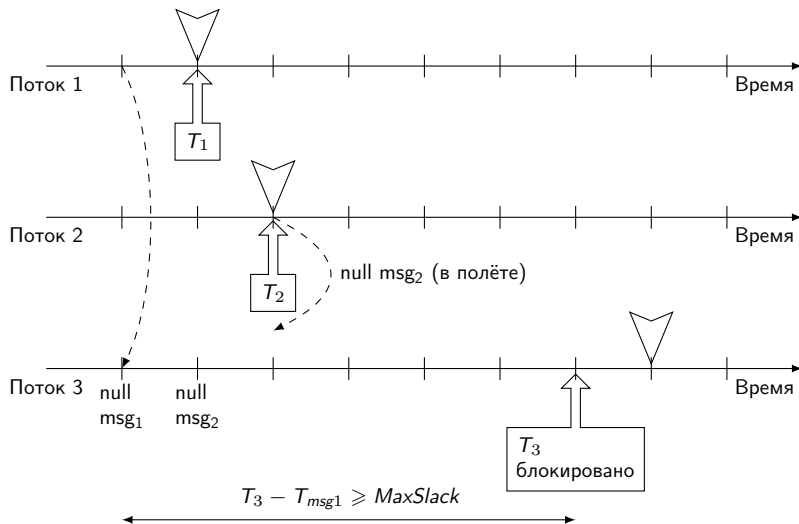
Можно ли избежать блокировок?

- Необходимость в них возникает из-за того, что отдельные потоки не знают, в какой стадии находятся остальные
- Как поток A может узнать локальное время потока B?
Через временную метку, хранящуюся в каждом событии от B

Пустые сообщения

- Периодическая рассылка пустых (null) сообщений, не связанных с архитектурными событиями, но несущими метку времени
- Каждый поток гарантированно имеет *приближённое* представление о том, не слишком ли он далеко убежал в будущее, и может сам (раз)блокировать своё исполнение

Slack



Пустые сообщения

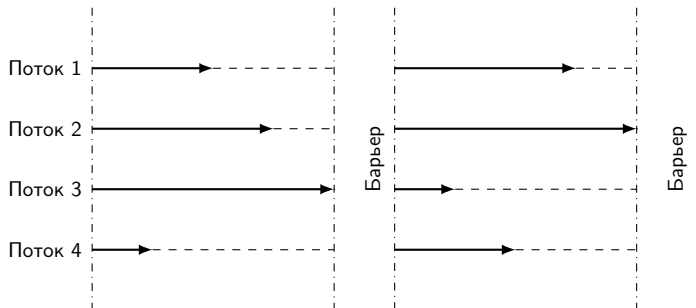
Как часто рассылать сообщения?

- Часто → потоки могут бежать свободнее, но большой трафик
- Редко → потоки не имеют актуальной информации о состояниях остальных и простаивают зря

Кому рассылать?

- Всем остальным — большой трафик
- Не всем — дедлоки вероятны
- Случайным адресатам — можно балансировать

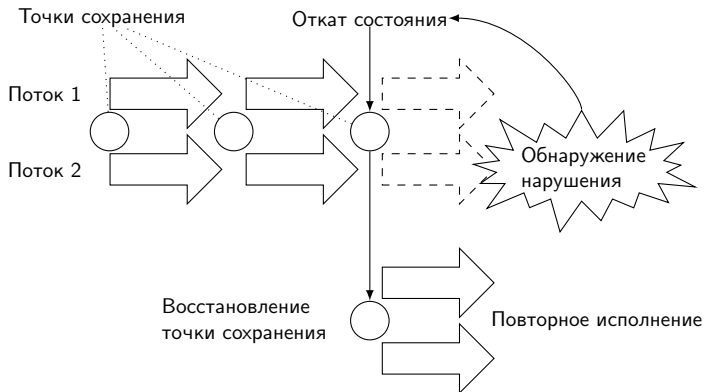
Частный случай: домены синхронизации



Оптимистичные схемы

- Параллельная программа работает без блокировок
- Периодически сохраняем (корректное) состояние системы
- При обнаружении каузальных ошибок откатываемся до ближайшего сохранённого состояния
- Проходим проблемный участок аккуратными методами (напр. консервативно)

Точки сохранения



Time Warp

- Сообщение — набор данных, описывающих событие, которое должно быть добавлено в одну из очередей событий. Оно характеризуется, кроме своего непосредственного содержимого, виртуальными временами отправки t_{send} и обработки $t_{receive}$.
- LVT (local virtual time) — значение симулируемого времени отдельного потока, участвующего в симуляции. Для создаваемых событий их время отправки t_{send} равно значению LVT отправителя. В отличие от консервативных схем, эта величина может как расти в процессе симуляции, так и убывать в случае отката процесса.

GVT

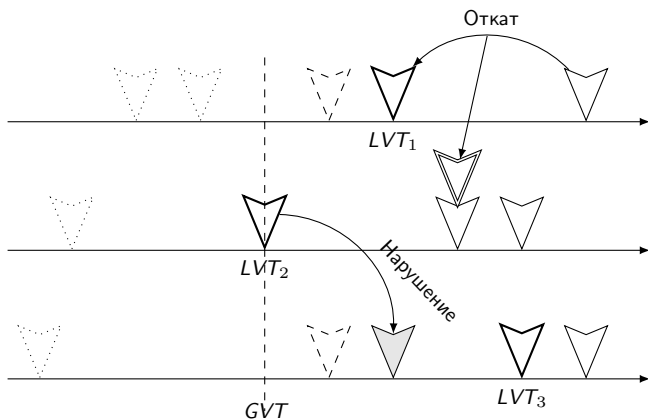
GVT (global virtual time) — глобальное время для всей симуляции, определяющее, до какой степени возможно её откатывать. Глобальное время всегда монотонно растёт, всегда оставаясь позади локального времени самого медленного потока, а также оно меньше времени отправки самого раннего ещё не доставленного события:

$$GVT \leq \min \left(\min_i LVT_i, \min_k t_k^{send} \right).$$

Straggler, antimessage

- Отставшее сообщение (straggler) — событие, пришедшее в очередь с меткой времени $t_{straggler}$, меньшей, чем LVT получателя. Его обнаружение вызывает *откат* текущего состояния, при этом LVT уменьшается, пока не станет меньше, чем $t_{straggler}$, после чего оно может быть обработано. После этого возобновляется прямая симуляция.
- Антисообщение (antimessage). Каждое антисообщение соответствует одному ранее созданному сообщению, порождённому в интервале симулируемого времени $[t_{straggler}, LVT_i]$ и вызывает эффект, обратный его обработке (т.е. отменяет его эффекты).

Работа Time Warp



Fossil Collection

Освобождение места, занятого сообщениями, расположенными левее GVT

Дополнительные вопросы параллельной симуляции

- Параллельная двоичная трансляция
- Распределённая общая память




Дополнительные вопросы параллельной симуляции

- Параллельная двоичная трансляция
- Распределённая общая память
- Почему параллельная симуляция настолько сложна?

Дополнительные вопросы параллельной симуляции

- Параллельная двоичная трансляция
- Распределённая общая память
- Почему параллельная симуляция настолько сложна?
«...необходимо определить, может или нет сообщение E_1 быть обработано одновременно с E_2 . Но каким образом узнать, влияет или нет E_1 на E_2 , без его симуляции?»

Рекомендуемая литература I

-  Fujimoto Richard M. Parallel discrete event simulation // Commun. ACM. — 1990. — Окт. — Т. 33, No 10. — С. 30–53. <http://doi.acm.org/10.1145/84537.84545>
-  Liu Jason. Parallel Discrete-Event Simulation. — 2009. <http://www.cis.fiu.edu/~liux/research/papers/pdes-eorms09.pdf>
-  Jayadev Misra. Distributed discrete-event simulation //ACM Computing Surveys 18 1986 www.cis.udel.edu/~cshen/861/notes/p39-misra.pdf

На следующей лекции

Эффективная современная виртуализация

Спасибо за внимание!

Слайды и материалы курса доступны по адресу
<http://is.gd/ivuboc>

Замечание: все торговые марки и логотипы, использованные в данном материале, являются собственностью их владельцев. Представленная здесь точка зрения отражает личное мнение автора, не выступающего от лица какой-либо организации.

Параллельные симуляторы

- Simics
- Graphite
- SimOS
- Coremu
- Pqemu
- BigSim
- DynamoRIO