

АЛГОРИТМ ВЫБОРА ОПТИМАЛЬНОЙ СИНХРОНИЗАЦИИ ПРОЦЕССОВ В РАСПРЕДЕЛЕННОЙ ИМИТАЦИОННОЙ СИСТЕМЕ

Благодаря интенсивному развитию вычислительной техники моделирование приобретает общенаучный характер и применяется в исследованиях объектов и процессов, происходящих в природе, в науках о человеке и обществе.

Одним из наиболее важных применений моделей в практическом, и в историческом аспектах является прогнозирование поведения моделируемых объектов.

Моделирование применяется сегодня в самых различных областях: экологии и геофизике (анализ распространения загрязняющих веществ в атмосфере), транспорте (конструирование транспортных средств, полетные имитаторы для тренировки пилотов), электронике и электротехнике (эмуляция работы электронных устройств), экономике и финансах (прогнозирование цен на финансовых рынках), архитектуре и строительстве (исследование поведения зданий, конструкций и деталей под механической нагрузкой), управлении и бизнесе (моделирование рынков сбыта и рынков сырья), промышленности (моделирование роботов и автоматических манипуляторов), медицине и биологии (моделирование пандемий и эпидемий), политике и военном деле (моделирование развития межгосударственных отношений, моделирование театра военных действий).

Применение моделей позволяет проводить контролируемые эксперименты в тех ситуациях, где экспериментирование на реальных объектах было бы практически невозможным или экономически нецелесообразным. [1] В таких случаях может быть построена модель, на которой необходимые эксперименты могут быть проведены с относительной легкостью и недорого.

Выделяют три основных вида моделирования: аналитическое, численное и имитационное [2]. В данной работе речь пойдет о имитационном моделировании. Имитационное моделирование — моделирование, при котором моделирующий алгоритм с той или иной степенью точности воспроизводит функционирование исходной системы. Имитационная модель воспроизводит поведение моделируемой системы во времени. Имитационная модель может быть выполнена на распределенной вычислительной системе, где выигрыш по времени выполнения модели достигается за счет параллельного выполнения событий, запланированных на разные моменты модельного времени.

Существует несколько технологий на уровне архитектур, которые позволяют задать основные правила создания распределенных имитационных моделей: DDS, ARIS, HLA и др.

HLA (High-level architecture) — архитектура высокого уровня, представляет собой архитектуру общего назначения для распределенных компьютерных имитационных систем. Является стандартом IEEE 1516. В соответствии с правилами построения имитационных моделей по архитектуре HLA были реализованы, например, такие проекты как объединение центров управления космическими аппаратами в рамках программы МКС в единую распределенную систему (РКК Энергия/Королев — ATVCC/Тулуза — NASA JSC/Хьюстон); предшественник HLA — сеть SIMNET реально использовалась для тренировок и поддержки военных операций (а также их последующего анализа), например, в 1992 году в рамках кампании «Буря в пустыне»; 333-й Центр боевой подготовки сухопутных войск в п. Мулино и др.

В имитационном моделировании принято различать три понятия времени: физическое, модельное, процессорное [3]. Сущностью имитационного моделирования является продвижение модельного времени при выполнении модели и выполнение событий, связанных с определенными значениями модельного времени. В распределенной модели первичной единицей является логический процесс. Каждый логический процесс выполняется в своем модельном времени как самостоятельная последовательная модель. Логический процесс взаимодействует с другими процессами, передавая им сообщения. При распределенной реализации модельное время в разных логических процессах движется с разными скоростями и в некоторый произвольный момент времени оказывается разным.

Проиллюстрируем сказанное на примере исторического события — битвы при Ватерлоо.

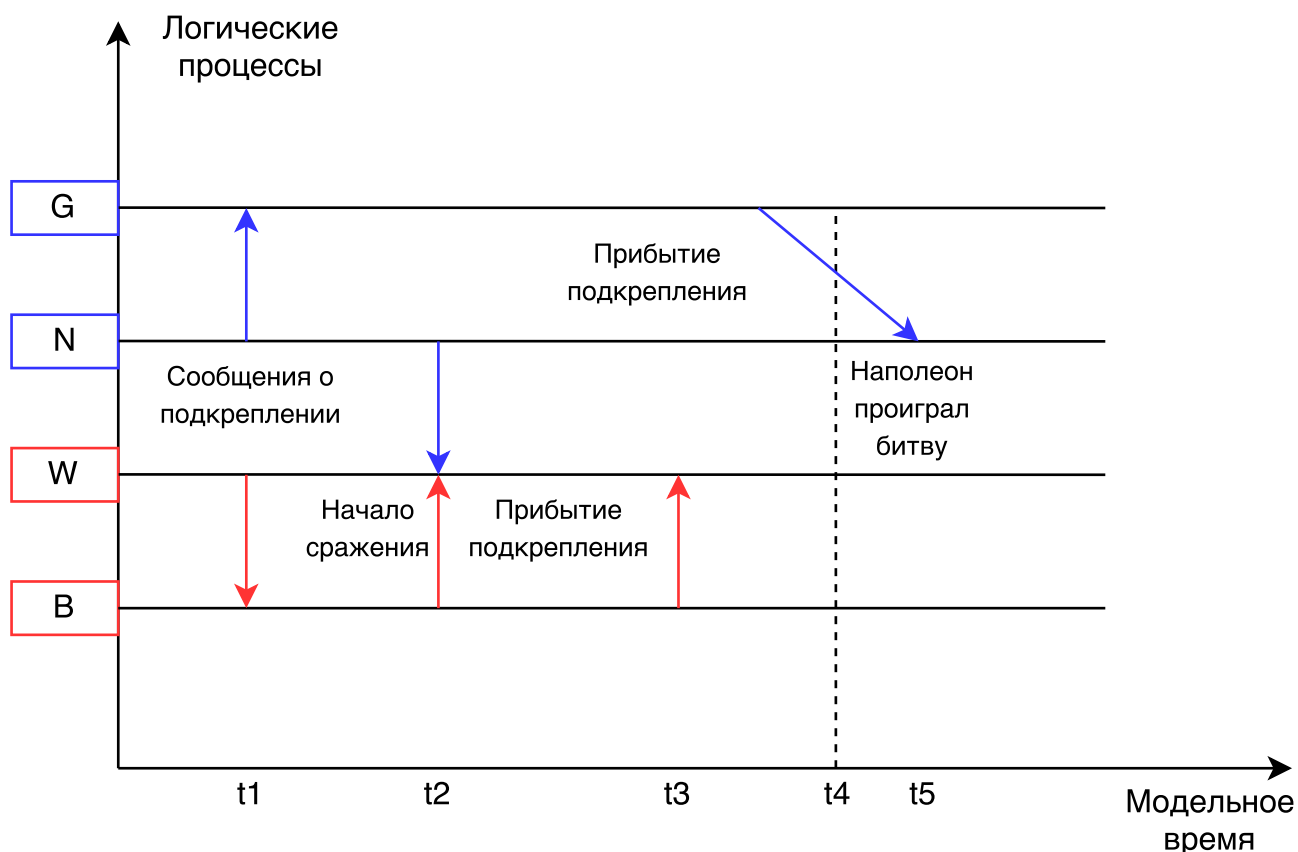


Рисунок 0.1 – Корректная временная диаграмма исполнения модели

Наполеон Бонапарт (процесс N) должен был вступить в бой с герцогом Веллингтоном (процесс W), расположившемся со своей английской армией на пути к Брюсселю. Но оба полководца ждали подкрепления: герцог Веллингтон — прусского князя Блюхера (процесс B), Наполеон — своего маршала Груши (процесс G). И Наполеон и Веллингтон отправили командующим своими подкреплениями сообщения с категорическим требованием скорейшего прибытия под Ватерлоо для усиления основной группировки войск. Тем временем, армия Бонапарта в ожидании прибытия подкрепления маршала Груши начала атаку неприятеля. Веллингтону ничего не оставалось, как принять бой. Не смотря на более выгодное расположение войск Веллингтона и примерно равные силы с Наполеоном, французы развивали успешное наступление на англичан и едва не опрокинули их — положение Веллингтона становилось критическим. И только вовремя подоспевшее подкрепление князя Блюхера помогло герцогу сдержать натиск Наполеона, а затем и перейти в контрнаступление на французов. После чего, так и не дождавшийся своего маршала с подкреплением, французский император был разбит.

Предположим, что эти процессы выполняются на распределенной системе. И логический процесс B в силу определенных причин (например, резко возросла

нагрузка на процессор) выполняется медленнее. Следовательно возникли задержки. Для логического процесса W получение сообщения в момент времени t_5 означает изменение прошлого в момент времени t_4 , что нарушает корректную последовательность развития моделирования.

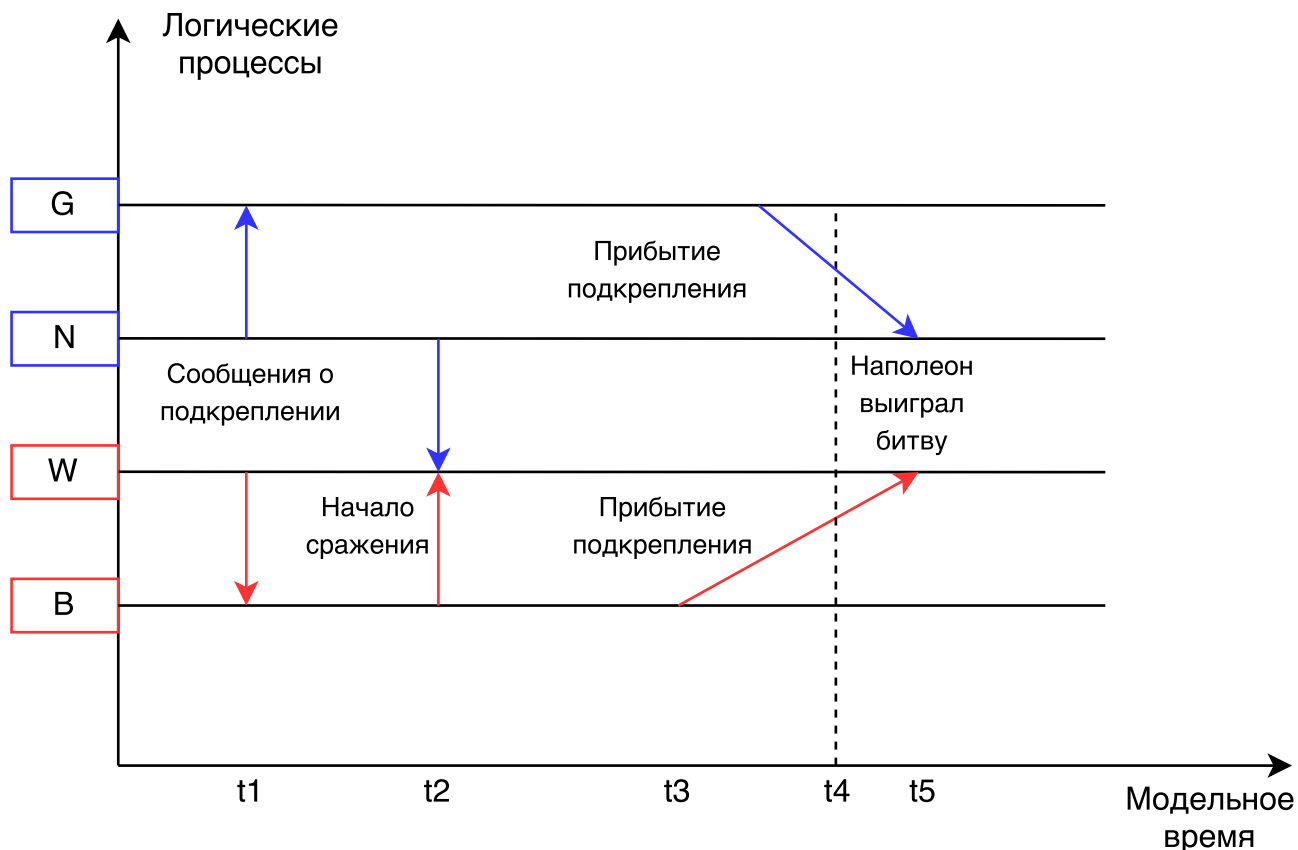


Рисунок 0.2 – Некорректная временная диаграмма исполнения модели

Для того, чтобы модель правильно воспроизводила последовательность событий в моделируемой системе, необходимо чтобы не возникало подобных парадоксов времени. Тот процесс продвигает свое время вперед, который получил сообщение от процесса с бóльшим временем. Если процесс получает сообщение от процесса с меньшим временем, т.е. $t_i^m < t_i^n$, $m \rightarrow n$, получаем парадокс, т.е. говорят, что процесс приславший сообщение «из прошлого» отстает во времени [3].

Для того, чтобы парадоксов времени в системе не возникало, необходимо предусмотреть специальные программы, которые бы синхронизировали по времени процессы в моделируемой системе. Такие программы получили название алгоритмов синхронизации модельного времени. Алгоритмы синхронизации разнообразны, но все их как правило можно разделить на два основных класса: консервативные и оптимистические.

Если синхронизация модельного времени процессов происходит под управлением консервативного алгоритма, это предполагает, что процесс получает сообщения в том же порядке, в котором их посылает ему отправитель. Иначе, консервативные алгоритмы блокируют продвижение модельного времени в том случае, если процесс получил сообщение от процесса с меньшим временем.

Оптимистические алгоритмы в аналогичном случае осуществляют откат времени процесса с большим временем до времени процесса, приславшего сообщение, обрабатывает его, а так же заново обрабатывают все сообщения от этого времени до текущего в правильной временной последовательности.

Математическая модель консервативного алгоритма синхронизации может быть представлена в виде [4]:

$$\left\{ \begin{array}{l} x_{i+1}^n = x_i^n + \xi_i^n F^n(i) \\ F^n(i) = \begin{cases} 1, & \text{если } \forall m \in M^n : x_i^n + \xi_i^n \leq x_i^m + \xi_i^m \\ 0, & \text{иначе} \end{cases} \\ x_0^n = 0 \\ n = \overline{1, N} \end{array} \right. \quad (0.1)$$

где x_{i+1}^n — время процесса n на следующем шаге, ξ_i^n — локальное приращение времени процесса n (время, затрачиваемое процессом на внутреннюю работу), n — процесс, который продвигает время (получает сообщение от процесса m), m — процесс, посылающий сообщение процессу n . Поскольку общее модельное время T_I — случайная величина, будем искать ее математическое ожидание. Модельное время всей системы на i -ом шаге равно минимальному времени процесса на этом шаге: $\forall i : M(T_i) = M(x_i)$. **Доказать оценку матожидания** С учетом оценки математических ожиданий величин $\{x_i^n\}$, получаем:

$$K^n(i) = \left\{ \begin{array}{l} \overline{M}(x_{i+1}^n) = \overline{M}(x_i^n) + M(\xi_i^n) \overline{F}^n(i) \\ \overline{F}^n(i) = \begin{cases} 1, & \text{если } \overline{M}(x_i^n) + M(\xi_i^n) \leq \overline{M}(x_i^m) + M(\xi_i^m) \\ 0, & \text{иначе} \end{cases} \\ M(x_0^n) = 0 \\ n = \overline{1, N} \end{array} \right. \quad (0.2)$$

Математическая модель оптимистического алгоритма может быть представлена в виде:

$$\begin{cases} x_{i+1}^n = x_i^n + \xi_i^n - \max_{m=1, \overline{N}} ((x_i^n - x_i^m) I_i^{m \rightarrow n} I_i^{x_i^m < x_i^n}) \\ x_0^n = 0 \\ n = \overline{1, N} \end{cases} \quad (0.3)$$

где $I_i^{m \rightarrow n}$ — событие, означающее посылку сообщения процессом m процессу n , $I_i^{x_i^m < x_i^n}$ — событие, означающее, что время процесса m меньше времени процесса n .

Если обозначить $\rho_i^{mn} = (x_i^n - x_i^m) I_i^{x_i^m < x_i^n}$, $\alpha_i^{mn} = I_i^{m \rightarrow n}$ получим следующую форму записи:

$$\begin{cases} x_{i+1}^n = x_i^n + \xi_i^n - \max_{m=1, \overline{N}} (\alpha_i^{mn} \rho_i^{mn}) \\ x_0^n = 0 \\ n = \overline{1, N} \end{cases} \quad (0.4)$$

С учетом оценки математических ожиданий величин $\{x_i^n\}$, получим:

$$O^n(i) = \begin{cases} \overline{M}(x_{i+1}^n) = \overline{M}(x_i^n) + M(\xi_i^n) - \max_{m=1, \overline{N}} (\alpha_i^{mn} \overline{M}(\rho_i^{mn})) \\ \overline{M}(\rho_i^{mn}) = \begin{cases} \overline{M}(x_i^n) - \overline{M}(x_i^m), & \text{если } \overline{M}(x_i^n) > \overline{M}(x_i^m) \\ 0, & \text{если } \overline{M}(x_i^n) \leq \overline{M}(x_i^m) \end{cases} \\ M(x_0^n) = 0 \\ n = \overline{1, N} \end{cases} \quad (0.5)$$

Оба приведенных алгоритма синхронизации обладают определенными недостатками: консервативный алгоритм допускает «простой» модельного времени процессов, оптимистический — «откаты» модельного времени.

Для уменьшения количества простоев и откатов модельного времени разработан алгоритм выбора оптимальной синхронизации процессов, основанный на анализе продвижения модельного времени каждым процессом.

Начало моделирования происходит под управлением консервативного алгоритма. Продвижение времени каждого процесса, участвующего в моделировании, оценивается на предмет изменения приращения его локального

времени, а так же интенсивности посылки этим процессом сообщений другим процессам, участвующим в моделировании.

Ознакомиться с источниками, по типам моделей, включить их в список литературы

Согласно [4] процессы можно условно разделить на три типа:

1. Тип 1. Математические ожидания приращений локальных часов имеют небольшой разброс $\leq 50\%$, процессы активно обмениваются сообщениями $\alpha^{mn} \rightarrow 1$. К данному типу могут быть отнесены имитационные модели сетей, многопроцессорных систем обработки сигналов и любого коммуникационного оборудования.

2. Тип 2. Математические ожидания приращений локального времени имеют неограниченный разброс, процессы довольно редко обмениваются сообщениями $\alpha^{mn} < 0.5$. Данному типу соответствуют имитационные модели работы сложных параллельных алгоритмов, где процессы в основном заняты своими внутренними вычислениями и обмениваются только входными данными и результатами. К этому типу могут относиться, например, имитационные модели алгоритмов бортового вычислительного комплекса летательного аппарата.

3. Тип 3. Разброс математических ожиданий неограничен, процессы обмениваются сообщениями с разной интенсивностью. К такому типу можно отнести полные модели распределенных вычислительных систем, например имитационная модель бортового навигационного комплекса.

Там же, в ходе экспериментов было установлено, что в зависимости от распределения приращений локальных часов можно говорить о наибольшей эффективности какого-либо алгоритма синхронизации для определенного типа процесса.

В общем виде алгоритм выбора оптимальной синхронизации следующий.

Шаг 1. Ввод исходных данных: число процессов, участвующих в моделировании, число шагов моделирования, вероятность посылки сообщений процессами лежит в интервале от 0 до 1, разброс приращений локальных часов — неограничен.

Шаг 2. Старт моделирования. На данном шаге генерируются последовательности посылки сообщений каждым процессом и приращения локального времени для процесса.

Шаг 3. Процессы в соответствии с последовательностями обмениваются сообщениями. Продвижение времени происходит по консервативному алгоритму.

Шаг 4. Если заключительный шаг моделирования не достигнут, то в течение (следующих) 10 шагов анализируются интенсивности посылки сообщений каждым процессом и величины локального приращения времени. Иначе — моделирование заканчивается. Если пара «*вероятность посылки сообщений — локальное приращение времени процесса*» удовлетворяет какому-либо заданному типу, то следующие 10 шагов процесс осуществляет продвижение времени, синхронизируя его с помощью наиболее оптимального для этой пары алгоритма синхронизации. Например, процесс посылает сообщения с интенсивностью 6 сообщений за 10 шагов. Это означает, что вероятность посылки составляет 0.6. Если приращение локального времени процесса в результате получения сообщений от других процессов $\leq 50\%$, то процесс принадлежит к типу 1. Для данного типа процессов наиболее эффективен консервативный алгоритм синхронизации. Это означает, что следующие 10 шагов процесс будет продвигать свое время с помощью консервативного алгоритма. Если один из параметров процесса указывает на его принадлежность к типу 2, то наиболее оптимальным алгоритмом синхронизации для процесса будет оптимистический и следующие 10 шагов процесс будет продвигать свое время с помощью этого алгоритма.

Шаг 5. Переход на Шаг 4.

Обозначим $\Theta^n(i)$ функцию, меняющую свое значение в зависимости от типа процесса

$$\bar{\Theta}^n(i) = \begin{cases} 1, & \text{если } \alpha^{mn} \rightarrow 1, \Delta T \leq 50\% \\ 0, & \text{если } \alpha^{mn} < 0.5, \Delta T \rightarrow \infty \\ \text{если } \alpha^{mn} \in [0 \dots 1], \Delta T \in [0 \dots \infty] \\ m, n & = \overline{1, N} \end{cases} \quad (0.6)$$

где ΔT — относительное приращение локального времени процессом.

$$\Delta T_{\text{отн.}} = \frac{\Delta T_{\text{абс.}}}{\min M(\xi_i^n)} 100\%, \quad (0.7)$$

$$\Delta T_{\text{абс.}} = \max M(\xi_i^n) - \min M(\xi_i^n).$$

Математическая модель выбора оптимального алгоритма синхронизации, с учетом оценок математического ожидания, будет иметь вид:

$$\left\{ \begin{array}{l} \overline{M}(x_{i+1}^n) = \overline{K}^n(i) \overline{\Theta}^n(i) + \overline{O}^n(i) \overline{\Theta}^n(i) \\ \overline{K}_i^n(i) = \overline{M}(x_i^n) + M(\xi_i^n) \overline{F}^n(i) \\ \overline{F}^n(i) = \begin{cases} 1, & \text{если } \overline{M}(x_i^n) + M(\xi_i^n) \leq \overline{M}(x_i^m) + M(\xi_i^m) \\ 0, & \text{иначе} \end{cases} \\ \overline{O}_i^n(i) = \overline{M}(x_i^n) + M(\xi_i^n) - \max_{m=\overline{1}, \overline{N}} (\alpha_i^{mn} \overline{M}(\rho_i^{mn})) \\ \overline{M}(\rho_i^{mn}) = \begin{cases} \overline{M}(x_i^n) - \overline{M}(x_i^m), & \text{если } \overline{M}(x_i^n) > \overline{M}(x_i^m) \\ 0, & \text{если } \overline{M}(x_i^n) \leq \overline{M}(x_i^m) \end{cases} \\ \overline{\Theta}^n(i) = \begin{cases} 1, & \text{если } \alpha^{mn} \rightarrow 1, \Delta T \leq 50\% \\ 0, & \text{если } \alpha^{mn} < 0.5, \Delta T \rightarrow \infty \\ \alpha^{mn} \in [0 \dots 1], \Delta T \in [0 \dots \infty] \\ m, n = \overline{1}, \overline{N} \end{cases} \\ \overline{M}(x_0^n) = 0 \\ m, n = \overline{1}, \overline{N} \end{array} \right. \quad (0.8)$$

Работа алгоритма была смоделирована с применением приложения Wolfram Mathematica и показала следующие результаты (на рисунке продвижение времени оптимальным алгоритмам обозначено как «КАС +»).

1. Все процессы принадлежат 1 типу.

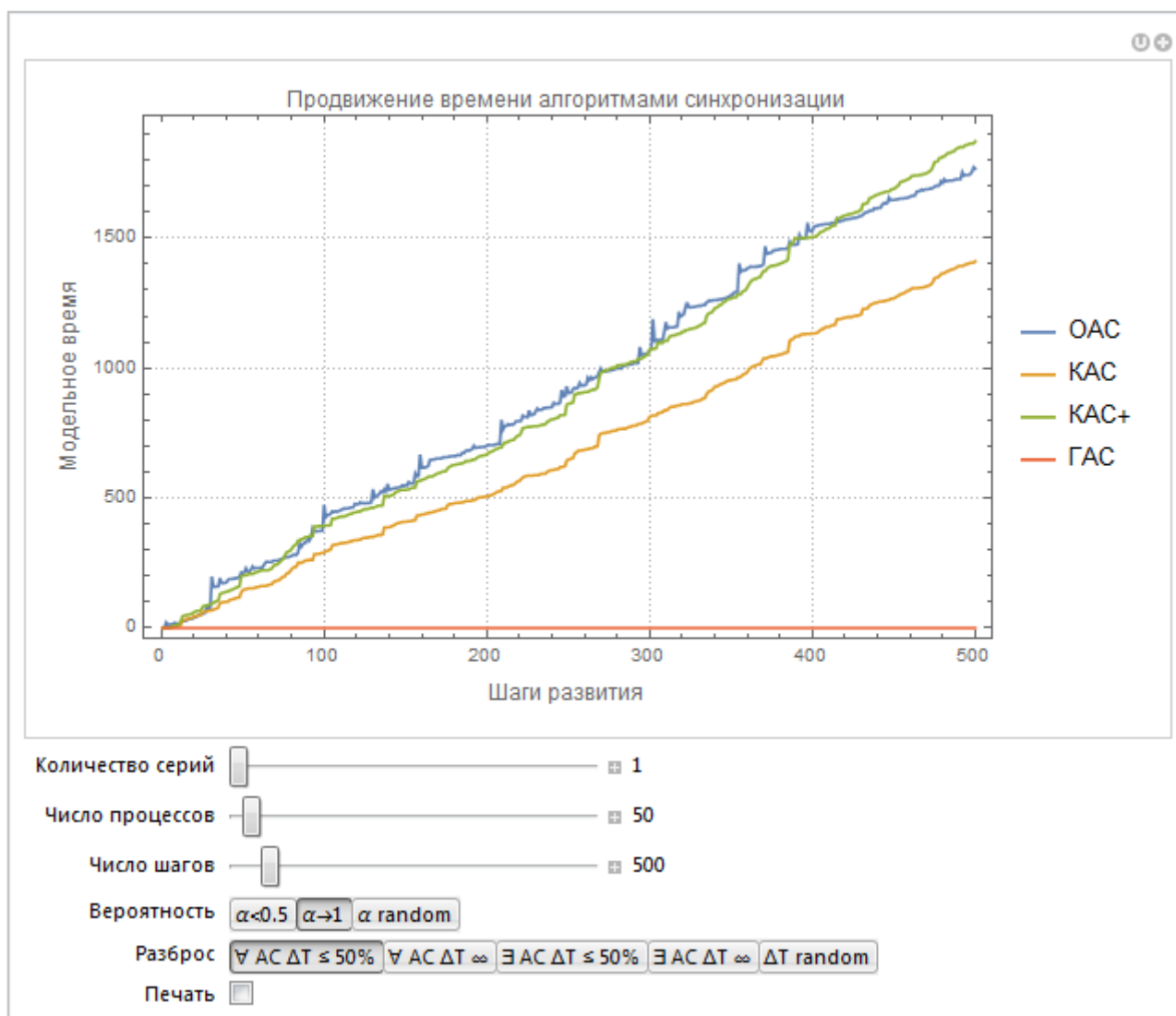


Рисунок 0.3 – Тип 1. Продвижение времени алгоритмами синхронизации

Характеристики / AC	KAC	KAC+	OAC	ГАС
Количество процессов	50			
К-во серий / К-во шагов	1 / 500			
ΔT	$\forall AC \Delta T \leq 50\%$			
P отсылки сообщений	$\rightarrow 1$			
Продвижение времени	1411	1872	1768	0
Кол-во простоев	103 248	138 011		0
Кол-во откатов		0	70 980	0
Эффективность	-25 %	100 %	-6 %	100 %
Соотношение	0.0136642	0.0135657	0.0249044	0.

Рисунок 0.4 – Тип 1. Сравнительная эффективность алгоритмов синхронизации

2. Все процессы принадлежат 2 типу.

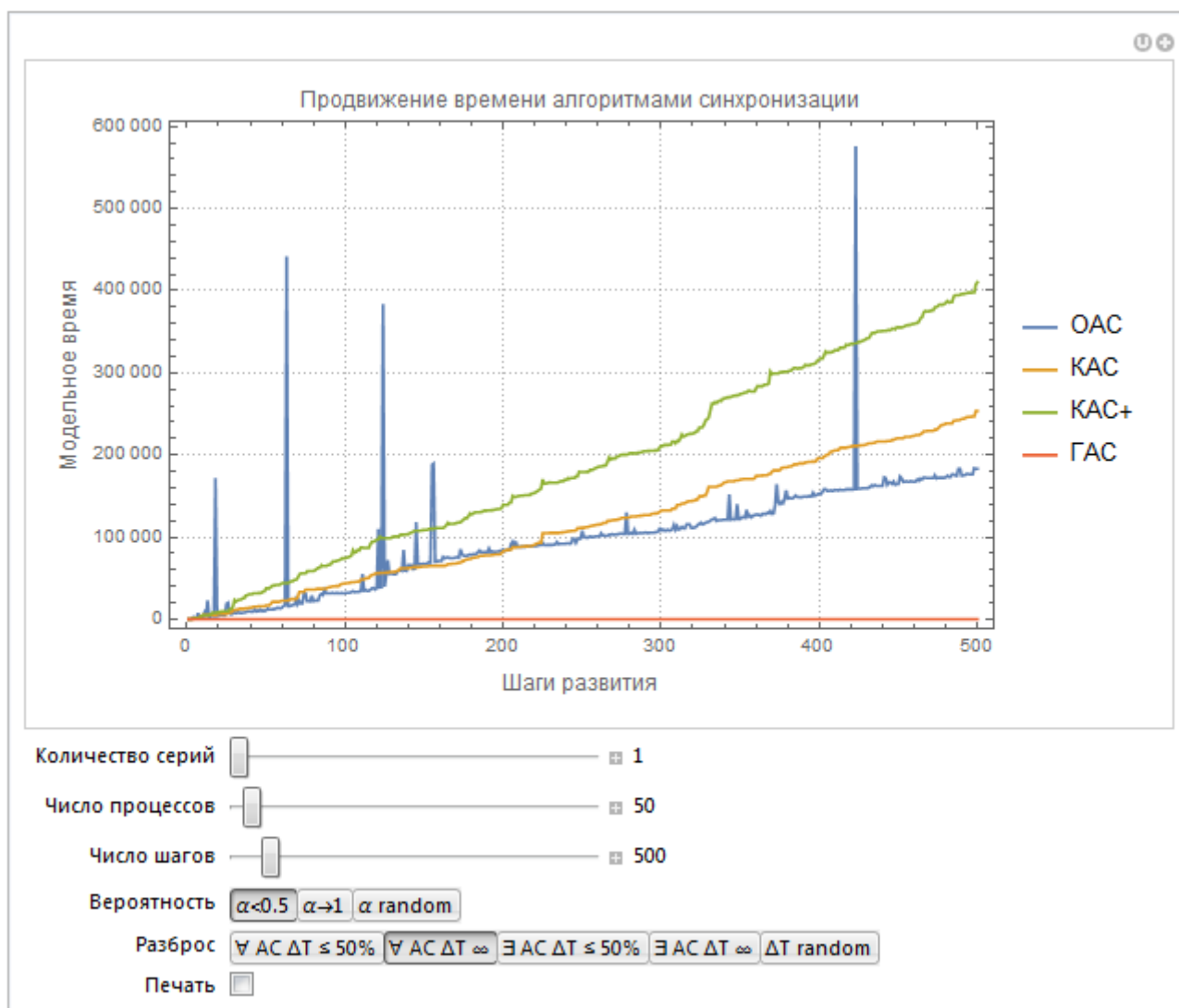


Рисунок 0.5 – Тип 2. Продвижение времени алгоритмами синхронизации

Характеристики / АС	КАС	КАС+	ОАС	ГАС
Количество процессов	50			
К-во серий / К-во шагов	1 / 500			
ΔT	$\forall AC \Delta T \infty$			
P послыки сообщений	<0.5			
Продвижение времени	253 521	410 540	183 017	0
Кол-во простоев	32 576	88 572		0
Кол-во откатов		49 226	25 794	0
Эффективность	-38 %	100 %	-55 %	100 %
Соотношение	7.78246	4.6351	7.09534	0.

Рисунок 0.6 – Тип 2. Сравнительная эффективность алгоритмов синхронизации

3. Процессы смешанного типа.

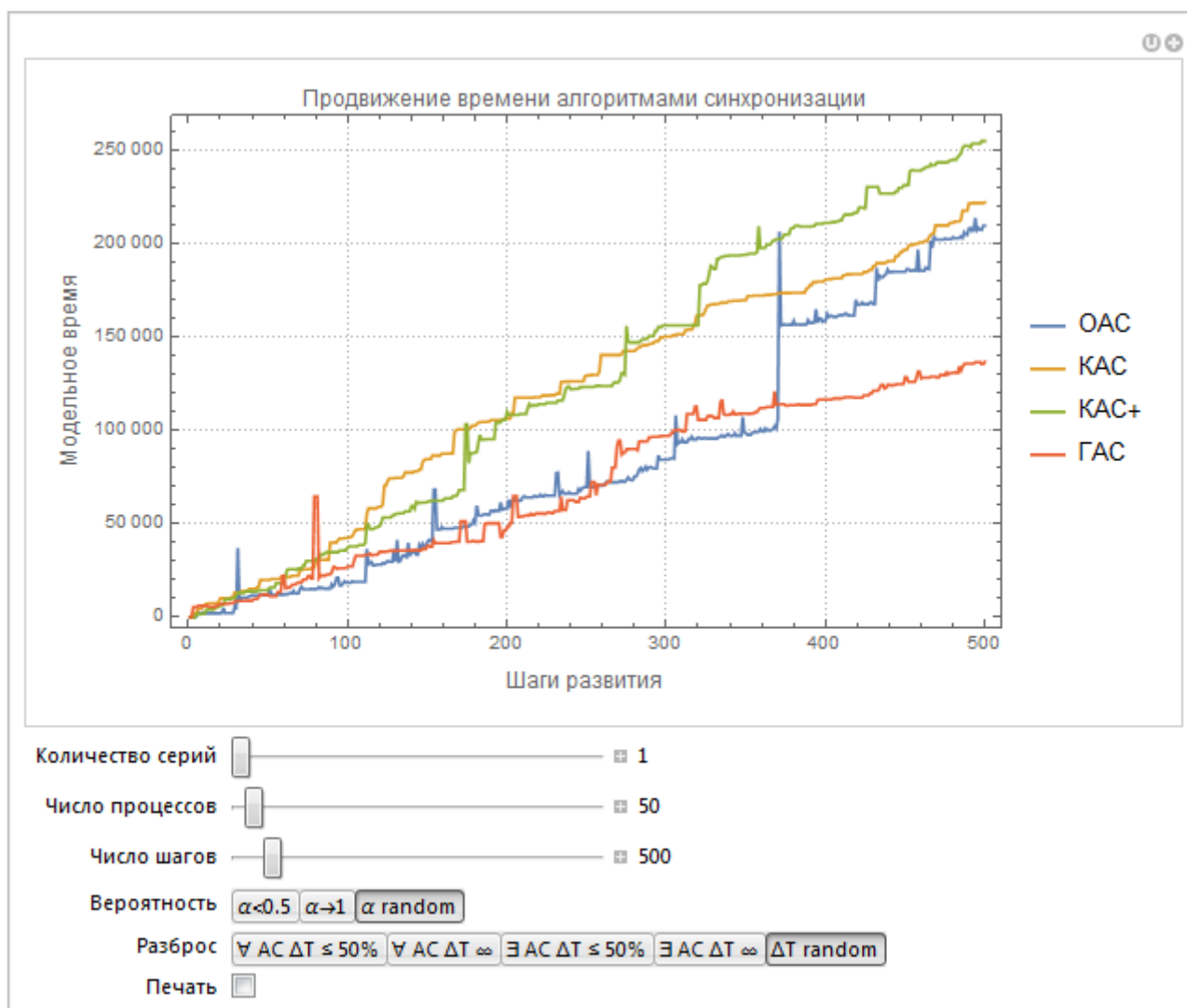


Рисунок 0.7 – Тип 3. Продвижение времени алгоритмами синхронизации

Характеристики / АС	КАС	КАС+	ОАС	ГАС
Количество процессов	50			
К-во серий / К-во шагов	1 / 500			
ΔT	$\leq 50\%$ 51 % / ∞ 49 %			
P послыки сообщений	< 0.5 50 % / $\rightarrow 1$ 50 %			
Продвижение времени	222 586	255 464	210 090	137 079
Кол-во простоев	62 534	119 939		52 520
Кол-во откатов		12 339	47 979	7245
Эффективность	-13 %	100 %	-18 %	-46 %
Соотношение	3.55945	2.12995	4.37878	2.61004

Рисунок 0.8 – Тип 3. Сравнительная эффективность алгоритмов синхронизации

Выводы.

Эксперименты показали, что продвижение времени процессами с помощью алгоритма выбора оптимальной синхронизации для всех типов процессов дает лучший результат, чем в случае, если бы продвижение времени происходило

с помощью только консервативного алгоритма или только оптимистического алгоритма синхронизации. В случае, когда синхронизация времени осуществляется для процессов типа 1, алгоритм выбора оптимальной синхронизации дает результаты едва лучшие алгоритма оптимистического. В случае, когда продвижение времени процессов осуществляется для процессов типа 2, алгоритм выбора оптимальной синхронизации показывает результаты лучшие, чем консервативный и значительно лучшие, чем оптимистический. При использовании смешанного типа процессов, алгоритм выбора оптимальной синхронизации так же способен показывать результаты, лучшие, чем алгоритмы консервативной и оптимистической синхронизации.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Шеннон Р. Имитационное моделирование систем — искусство и наука. — Москва : Издательство «Мир», 1978. — 411 с.
- 2 Дигрис А.В. Дискретно-событийное моделирование. — Минск : БГУ, 2011. — 175 с.
- 3 Окольнішников В. В. Представление времени в имитационном моделировании // Вычислительные технологии. — 2005. — № 5. — С. 23.
- 4 Вознесенская Тамара Васильевна. Исследование эффективности алгоритмов синхронизации времени для систем распределенного имитационного моделирования: дис. ...канд. физ.-мат. наук: 05.13.11 / Вознесенская Тамара Васильевна. — Москва : МГУ, 2001. — 95 с.