

Параллельный алгоритм имитационного моделирования динамики объема груза на складе хранения

О.Н. Яркова, Л.П. Москаленко

Санкт-Петербургский государственный архитектурно-строительный университет

Аннотация: Применение имитационного и статистического моделирования требует проведения большого количества имитаций и времени вычислений. Сократить время расчета в сложных системах имитационного и статистического моделирования позволяет внедрение технологий параллельного программирования в реализуемые модели. В настоящей работе ставится задача распараллеливания алгоритма имитационного моделирования динамики некоторого показателя (на примере модели динамики объема груза на складе хранения), описываемого моделью, представленной в виде линейной комбинации входных и выходных потоков, заданных в виде: моделей авторегрессии скользящего среднего с трендовыми составляющими; потоков описываемых процессами, заданными по принципу суммы произведений интенсивности на объем (размер) исследуемого параметра, при условии стационарности каждого из них. Предложена схема распараллеливания алгоритма с применением технологии OpenMP. Проведены вычислительные эксперименты, оценены показатели эффективности параллельного алгоритма в зависимости от числа имитационных экспериментов.

Ключевые слова: имитационное моделирование, параллельное программирование, эффективность параллельного алгоритма, модель загрузки склада, технология OpenMP.

Введение

В настоящее время для проведения анализа поведения сложных стохастических процессов используется имитационное и статистическое моделирование. Однако, зачастую, исследователи ограничиваются небольшим количеством имитационных экспериментов, что обусловлено временными ограничениями и/или сложностью реализации повторов модель. Однако реализация небольшого количества экспериментов приводит к снижению точности моделирования, так как рассматриваются только единичные траектории случайных процессов, протекающих в исследуемой системе. Сократить время расчета в сложных системах имитационного и статистического моделирования позволяет внедрение технологий параллельного программирования в реализуемые модели.

Исследованию применения параллельных алгоритмов для решения различных задач посвящено большое количество работ. V.V. Zosimov, O.S.

Vulgakova [1] провели сравнение производительности параллельных вычислений в разных реализациях и языках программирования, сформулировали достоинства и недостатки OpenMP и TPL при реализации на языках C++ и C#. А.А. Аль-Хулайди и Ю.О. Чернышев [2] анализировали эффективность параллельного алгоритма решения транспортной задачи на вычислительном кластере, оценили размерность задачи, при которой целесообразно применение параллельных технологий. J. Huang и L. Chen [3] предложили программную реализацию параллельного алгоритма решения задачи затвердевания изделий литейной промышленности на основе технологии OpenMP на двухядерном микрокомпьютере. Вопросы параллельной обработки паттернов в больших данных в системах имитационного моделирования занимались А.И. Миков и Н.В. Моспан [4]. Б.М. Глинский с соавторами [5] проводили анализ масштабируемости алгоритмов для суперкомпьютеров при реализации агентно-ориентированных моделей в пакете AGNES (AGent NEtwork Simulator). Авторы исследовали параллельные реализации алгоритма моделирования методом Монте-Карло для задачи динамики разреженного газа, разностного метода распространения сейсмических полей, имитационного моделирования сеточного метода. Д.А. Мигов и Д.В. Винс [6] провели исследование задачи моделирования надежности сети методом статистического моделирования, подобрали оптимальные значения параметров масштабируемости параллельного алгоритма при расчете на суперЭВМ большой производительности. Авторы исследовали зависимость ускорения времени вычислений при распараллеливании алгоритма от количества вычислительных ядер. А.В. Калюжный с соавторами [7] рассматривали модель для ресурсоемкости стохастических параллельных процессов и показали возможность улучшения энергоэффективности вычислительных элементов, выполняющих параллельные расчеты. Н.П. Блинова и

П.Е. Ломовцева [8] анализировали возможности распараллеливания алгоритма имитационного моделирования процессов материально-технического обеспечения войск (сил) на примере графа взаимосвязей событий модели с целью сокращения времени проведения экспериментов. В.В. Симонов [9] оценивал эффективность параллельных алгоритмов моделирования многослойного персептрона. Отметим, что, несмотря на большое количество работ по реализации параллельных вычислительных алгоритмов в системах имитационного моделирования и разнообразие задач, решаемых исследователями, нет универсального подхода к распараллеливанию, и каждая задача требует выработки своего подхода.

В настоящей работе ставится задача распараллеливания алгоритма имитационного моделирования динамики некоторого показателя, описываемого моделью (далее – Модель (1)), представленной в виде линейной комбинации входных и выходных потоков, заданных в виде: моделей авторегрессии скользящего среднего с трендовыми составляющими; потоков описываемых процессами, заданными по принципу суммы произведений интенсивности на объем (размер) исследуемого параметра, при условии стационарности каждого из них [10, 11].

Моделирование на основе метода Монте-Карло предусматривает выполнение большого количества имитаций для достижения заданной степени точности расчетов, и это количество существенно возрастает при увеличении количества моделируемых параметров. Так, например, в рассматриваемом при апробации алгоритма примере решения задачи моделирования финансовых ресурсов [10], потребовалось 60000 имитаций для достижения точности 0.05 при использовании девяти входных потоков поступления/оттока финансовых средств; в работе [12] при пяти входных параметрах, заданных стандартными законами распределения, авторы применяли 10000 имитаций при моделировании методом Монте-Карло

задачи оценки показателей эффективности при производстве зерна и зернобобовых культур. Таким образом, даже при небольшом количестве входных параметров расчеты методом имитационного моделирования требуют продолжительного времени.

Цель и задачи исследования

Цель исследования – разработка схемы распараллеливания и реализация параллельного алгоритма решения задачи имитационного моделирования динамики показателя, описываемого Моделью (1).

Задачи исследования: разработка алгоритма решения задачи имитационного моделирования динамики загрузки склада - объема груза на складе хранения; разработка параллельного алгоритма моделирования показателя динамики загрузки склада; реализация параллельного алгоритма моделирования показателя динамики загрузки склада; анализ эффективности распараллеливания.

Методы исследования

Формализуем модель на примере модели загрузки склада. Рассмотрим один из показателей загрузки склада – объем складированного груза, как функцию времени t ($t=0, 1...T$), где T – период исследования. Объем складированного груза зависит от объема накопленных ранее грузов, поступлений груза на склад и оттока (выдачи) груза со склада. Имитационная модель (Модель (1)) для оценки динамики объема складированного груза имеет вид:

$$Q_t = Q_{t-1} + \sum_{i=1}^m \sum_{j=0}^{M_t^i} D_t^{i,j} + \sum_{i=1}^{s1} \eta_t^i - \sum_{i=1}^p \sum_{j=0}^{N_t^i} R_t^{i,j} - \sum_{i=s1+1}^{s1+s2} \eta_t^i, \quad t=1,2...T, \quad Q_0 = Q^0, \quad (1)$$

где Q_t – объем складированного груза в момент времени t ; Q^0 – объем груза на складе при $t=0$; M_t^i – стационарный случайный процесс, характеризующий количество поступающих грузов на склад в момент времени t по i -му виду

потока поступлений ($i=1,2..m$); $D_t^{i,j}$ – стационарный случайный процесс, отражающий объем j -го ($j=0,1..M_t^i$) груза, поступившего в момент времени t , по i -му ($i=1,2..m$) виду потока поступления; p – количество однородных видов груза, убывающих со склада для тех видов потоков, по которым количество единиц груза убывающего со склада являются стационарным случайным процессом; N_t^i – стационарный случайный процесс, характеризующий количество однородных грузов i -го вида ($i=1,2..p$), убывающих со склада в момент времени t ; $R_t^{i,j}$ – стационарный случайный процесс, отражающий объем j -ого ($j=0,1..N_t^i$) груза i -го вида ($i=1,2..p$), убывающего со склада в момент времени t ; η_t^i – временной ряд, характеризующий поступления ($i=1,2..s1$) и отток ($i=s1+1,s1+2..s1+s2$) грузов, описываемые моделями АРСС с трендовой компонентой [10].

Далее представлен алгоритм имитационного моделирования.

Входные параметры: $Lsim$ – количество имитаций; T – период моделирования; Q^0 – объем груза на складе в начальный момент времени; коэффициенты трендов и авторегрессионных моделей η_t^i ($i=1,2,\dots,s1+s2$) и исходные статистические данные к ним, для прогнозирования параметров; статистические данные, характеризующие количество и объем поступления и оттока грузов M_t^i ($i=1,2..m$); $D_t^{i,j}$ ($j=0,1..M_t^i, i=1,2..m$); N_t^i ($i=1,2..p$); $R_t^{i,j}$ ($j=0,1..N_t^i, i=1,2..p$).

Этап 1.

1.1 Чтение исходных данных из файлов, оценка функций распределения параметров для моделирования методом обратной функции (на этапах 3.5, 3.7). Данные сохраняются в виде многомерных массивов, где первый индекс определяет номер потока поступления/оттока средств, и по

этому индексу далее в программе выбирается нужная информация для моделирования случайной величины.

Этап 2.

2.1 Формирование и инициализация структур данных: матриц коэффициентов моделей АРСС с трендовой компонентой для моделирования потоков поступлений грузов η^i ($i=1, 2\dots s1$);

2.2 Формирование и инициализация структур данных: матриц коэффициентов моделей АРСС с трендовой компонентой для моделирования оттока грузов со склада η^i ($i=s1, s1+1\dots s2$).

Этап 3.

3.1. $L = 1$.

3.2. $Q_0^l = Q^0$.

3.3. $t = 1$.

3.4. $Sum_t = 0$.

3.5. Генерация методом обратной функции реализаций M_t^i ($i = 1, 2..m$); $D_t^{i,j}$ ($j = 0, 1..M_t^i, i = 1, 2..m$); N_t^i ($i = 1, 2..p$); $R_t^{i,j}$ ($j = 0, 1..N_t^i, i = 1, 2..p$).

3.6. Вычисление: $Sum_t = Sum_t + \sum_{i=1}^m \sum_{j=0}^{M_t^i} D_t^{i,j} - \sum_{i=1}^p \sum_{j=0}^{N_t^i} R_t^{i,j}$.

3.7. Генерация методом обратной функции реализации остатков ε_t^i ($i=1, 2..s1+s2$) модели η^i . Формирование η_t^i , где τ – модельное время.

3.8. Вычисление: $Sum_t = Sum_t + \sum_{i=1}^{s1} \eta_t^i - \sum_{i=s1+1}^{s1+s2} \eta_t^i$.

3.9. Вычисление: $Q_t^l = Q_{t-1}^l + Sum_t$.

3.10. Если ($t < T$), то $t = t + 1$ и идти на шаг 3.4, иначе шаг 3.11.

3.11. Если $l < Lsim$, то $l = l + 1$ и идти на шаг 3.2, иначе конец алгоритма.

Этап 4.

4.1 Визуализация результатов моделирования в виде таблиц, графиков; статистический анализ; сохранение в файл.

Конец алгоритма.

Выходные данные: реализации $\{Q_t^l\}$, $t = 1, 2..T$, $l = 1, 2..Lsim$.

Проанализируем этапы алгоритма.

Этап 1 предусматривает чтение исходной информации из файлов и последующую инициализацию структур данных для последующего моделирования случайных компонент модели. Указанный этап не может быть распараллелен, так как требует непосредственного участия пользователя для выбора нужных файлов с исходной информацией. Хранение же большого объема статистических данных, представляющих исходную информацию в файлах, для распараллеливания этапов оценки функций распределения нецелесообразно.

Этап 2 может быть распараллелен с использованием критических секций (`#pragma omp sections`), так как шаги 2.1, 2.2 не связаны между собой по данным. Максимальное число задействованных процессоров и потоков на этом этапе равно двум.

Этап 3 целесообразно распараллелить по параметру l (`#pragma omp for`). Каждым потоком в очередной момент времени будет рассчитываться одна из имитаций алгоритма, что позволит реализовать равномерную загрузку процессоров. Максимальное число задействованных процессоров и потоков на этом этапе будет определяться по числу доступных процессоров. Шаг 3.11 предусматривает работу потоков с общей переменной, он должен выполняться только одним потоком в определенный момент времени, следовательно, его будем реализовывать, как критическую секцию (`#pragma omp critical`).

Распараллеливание производится с использованием OpenMP. Схема распараллеливания выглядит следующим образом (рис. 1).

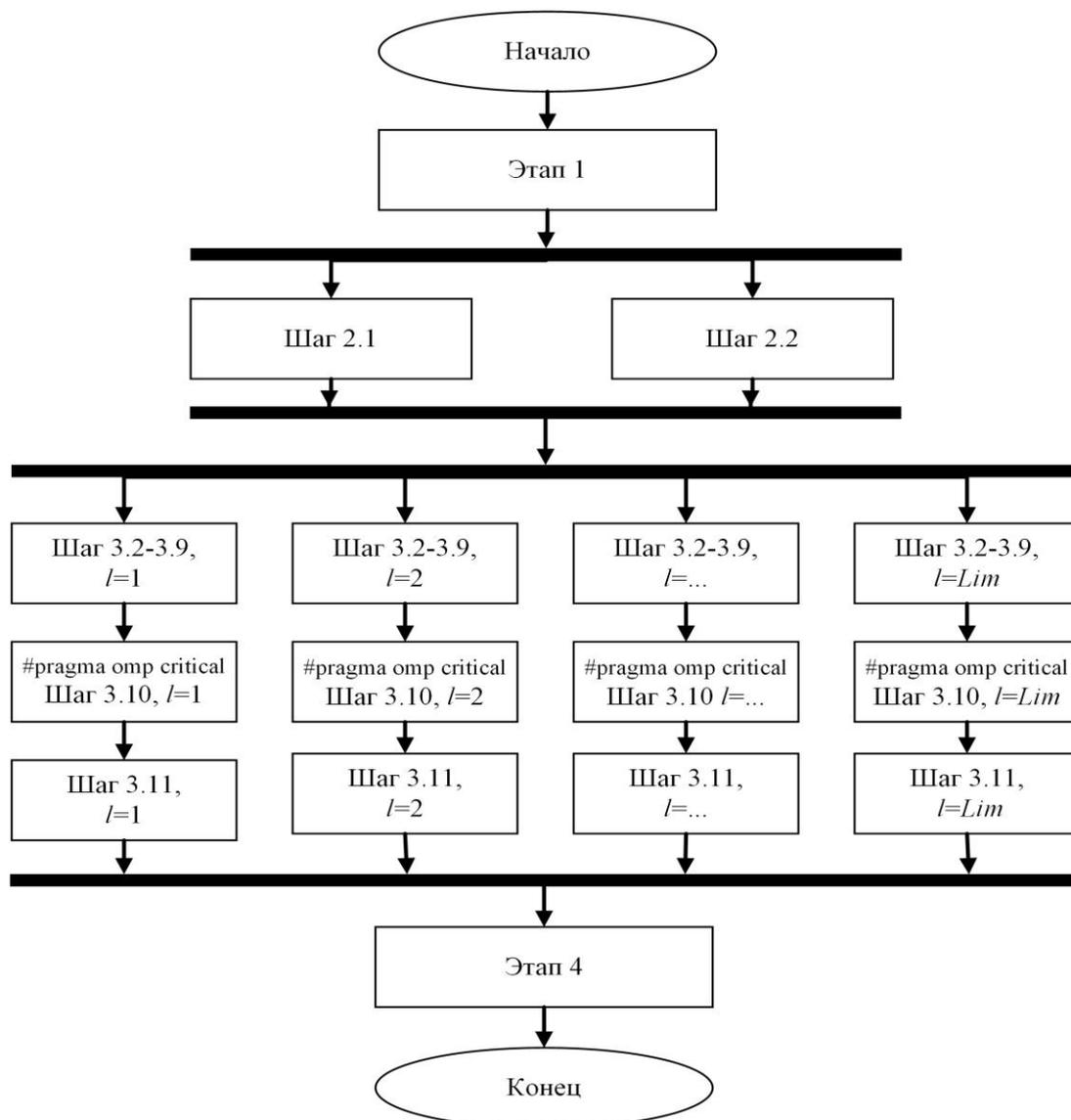


Рис. 1. – Схема распараллеливания алгоритма

Вычислительный эксперимент

Проанализируем эффективность распараллеливания. Для этого фиксировалось время вычислений на этапах 2 и 3 алгоритма. Результаты расчетов показателей времени работы алгоритма в однопоточном и многопоточном (при количестве процессоров 6) режимах приведены в таблице №1.

Расчеты проводились на персональной ЭВМ на процессоре AMD FX (tm)-6300 Six-Core Processor 3.5GHz, оперативная память 24 Гб.

Таблица № 1

Показатели эффективности расчетов

L_{sim}	Однопоточный расчет, время, сек.		Многопоточный расчет ($p = 6$), время, сек.		Ускорение		Эффективность	
	Этап 2-4	Этап 3	Этап 2-4	Этап 3	Этап 2-4	Этап 3	Этап 2-4	Этап 3
1	0,006	0,002	0,006	0,002	1,021	0,145	0,170	0,145
500	0,564	0,556	0,266	0,254	2,116	0,365	0,353	0,365
1000	1,775	1,766	0,418	0,402	4,242	0,731	0,707	0,731
2000	2,345	2,332	0,682	0,669	3,439	0,581	0,573	0,581
5000	5,678	5,609	1,529	1,471	3,714	0,636	0,619	0,636
10000	11,244	11,151	2,756	2,677	4,079	0,694	0,680	0,694
20000	23,380	23,200	5,964	5,370	3,920	0,720	0,653	0,720
	23,444	23,193	6,364	5,247	3,684	0,737	0,614	0,737
	24,254	23,647	5,499	5,250	4,410	0,751	0,735	0,751
30000	35,644	34,819	8,954	7,848	3,981	0,739	0,663	0,739
	35,770	34,936	9,111	7,987	3,926	0,729	0,654	0,729
	36,218	34,705	9,457	8,416	3,830	0,687	0,638	0,687
40000	48,589	46,772	12,483	10,988	3,892	0,709	0,649	0,709
	48,459	46,531	12,931	10,808	3,747	0,718	0,625	0,718
	48,775	46,557	12,719	10,844	3,835	0,716	0,639	0,716
50000	60,645	58,155	15,110	12,950	4,013	0,748	0,669	0,748
	60,882	58,305	15,014	12,854	4,055	0,756	0,676	0,756
	61,198	58,440	15,392	12,969	3,976	0,751	0,663	0,751
60000	72,932	69,869	18,559	15,921	3,930	0,731	0,655	0,731
	72,975	69,877	18,251	15,430	3,998	0,755	0,666	0,755
	73,758	70,037	19,533	16,085	3,776	0,726	0,629	0,726
70000	85,565	81,845	21,506	18,016	3,979	0,757	0,663	0,757
	85,713	81,825	22,402	18,014	3,826	0,757	0,638	0,757
	85,860	81,749	22,142	18,056	3,878	0,755	0,646	0,755
80000	97,589	93,233	24,949	20,558	3,912	0,756	0,652	0,756
	97,642	93,200	24,989	20,559	3,907	0,756	0,651	0,756
	97,590	92,852	25,388	20,659	3,844	0,749	0,641	0,749
90000	109,802	104,719	28,168	23,143	3,898	0,754	0,650	0,754
	109,595	104,482	28,290	23,171	3,874	0,752	0,646	0,752
	109,848	104,333	28,684	23,225	3,830	0,749	0,638	0,749
100000	121,413	115,711	31,172	25,781	3,895	0,748	0,649	0,748
	122,431	116,693	31,526	25,759	3,883	0,755	0,647	0,755
	122,660	116,555	31,803	25,737	3,857	0,755	0,643	0,755
200000	225,728	224,813	53,092	50,993	4,252	0,735	0,709	0,735
500000	564,551	562,827	130,994	128,329	4,310	0,731	0,718	0,731
800000	901,768	898,692	210,347	208,084	4,287	0,720	0,715	0,720

Вычисления проводились при следующих количествах потоков в модели: $s_1=1$, $s_2=2$, $m=4$, $M[N_1]=0.9$, $M[N_2]=414.6$, $M[N_3]=0.9$, $M[N_4]=6.8$,

$p=7$, $M[M1]=16.0$, $M[M2]=29.4$, $M[M3]=23.9$, $M[M4]=156.8$, $M[M5]=92.6$, $M[M3]=8.8$, $M[M4]=19.6$, где $M[*]$ -математическое ожидание параметра. Вид моделей APCC и трендов не приводится, так как цель работы не связана с анализом итогового показателя, пример задания моделей можно посмотреть в работах [8, 9].

Ускорение определяется величиной $S_p = T_1/T_p$, где T_1 – время выполнения последовательного алгоритма, T_p – время выполнения параллельного алгоритма с использованием p процессоров.

Эффективность распараллеливания характеризует долю времени, которую вычислительные потоки тратят на вычисления и определяется величиной: $E_p = T_1/(p * T_p)$.

На рис. 2 приведена зависимость времени выполнения алгоритма от количества имитаций для однопоточного и многопоточного режима, при количестве процессоров 6.

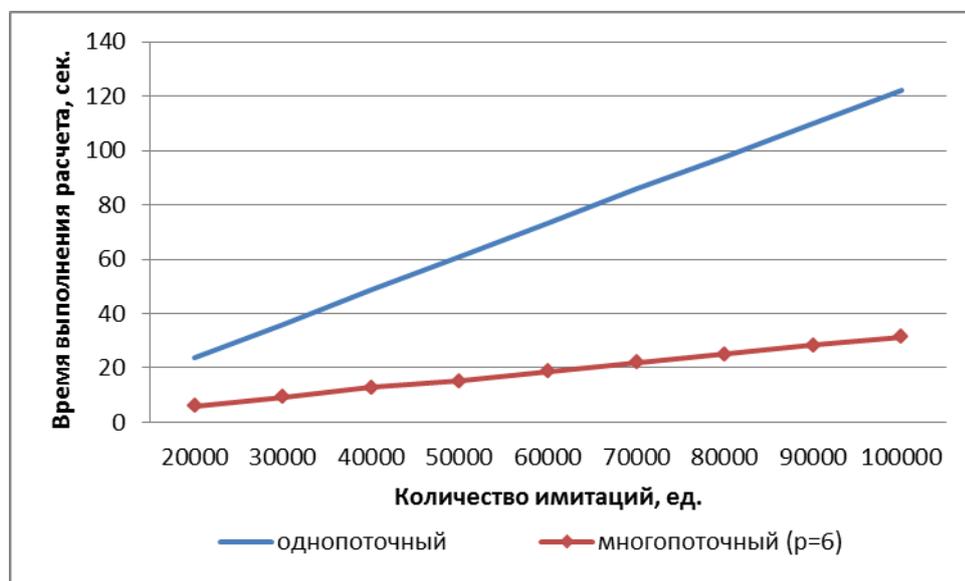


Рис. 2 – Зависимость времени выполнения алгоритма от количества имитаций для однопоточного и многопоточного режима, при $p = 6$

В среднем показатели эффективности имеют следующие значения:

- ускорение для этапов 2-4 равно 3,94;
- эффективность для этапов 2-4 равна 0,66;
- ускорение для этапа 3 равно 4,38;
- эффективность для этапа 3 равна 0,73.

Расчеты показали линейную зависимость времени выполнения алгоритма от количества имитаций. При небольших значениях числа имитаций не целесообразно использовать распараллеливания, к примеру, при количестве имитаций 1 – 500. При значении числа имитаций от 1000 наблюдаются стабильные показатели ускорения, приблизительно равные 4 и эффективности 0,66 для всего алгоритма и 4,38 и 0,73 соответственно для ресурсоемкого по времени этапа 3. Таким образом, проведенный анализ показал высокие показатели эффективности распараллеливания.

Заключение

В работе проведено исследование эффективности распараллеливания алгоритма имитационного моделирования динамики показателя (на примере модели динамики объема груза на складе хранения), описываемого моделью, представленной в виде линейной комбинации входных и выходных потоков, заданных в виде: моделей АРСС с трендовыми составляющими; потоков описываемых процессами, заданными по принципу суммы произведений интенсивности на объем (размер) исследуемого параметра, при условии стационарности каждого из них. Предложена схема распараллеливания алгоритма с применением технологии OpenMP. Оценены показатели эффективности параллельного алгоритма. Построена зависимость времени выполнения последовательного и параллельного алгоритма от количества имитаций. Эффективность основного этапа реализации имитаций 73%, ускорение вычисления 4,38 при количестве процессоров 6. Вычислительные

эксперименты показали достаточно высокие значения эффективности предложенного параллельного алгоритма.

Работа выполнена в рамках гранта на выполнение научно-исследовательских работ научно-педагогическими работниками Санкт-Петербургского государственного архитектурно-строительного университета.

Литература

1. Zosimov V.V., Bulgakova O.S. Optimizing Computational Performance with OpenMP Parallel Programming Techniques // Control Systems and Computers. 2023. No. 3 (303). pp. 61-68. DOI: 10.15407/csc.2023.03.061.

2. Аль-Хулайди А.А., Чернышев Ю.О. Разработка параллельного алгоритма нахождения оптимального решения транспортной задачи на кластере // Инженерный вестник Дона. 2011. №2. URL: ivdon.ru/ru/magazine/archive/n2y2011/445.

3. Huang J., Chen L. Parallel Numerical Simulation of Solidification Based on OpenMP // Foundry. 2006. No. 55. pp. 704-706.

4. Миков А.И., Моспан Н.В. Параллельная обработка паттернов в больших данных имитационного моделирования // Информатизация и связь. 2013. №2. С. 141-143.

5. Глинский Б.М., Родионов А.С., Марченко М.А., Караваев Д.А., Подкорытов Д.И., Винс Д.В. Использование имитационного моделирования для настройки параметров масштабируемых алгоритмов при высокопроизводительных вычислениях // Вестник Уфимского государственного авиационного технического университета. 2013. Т. 17. №5(58). С. 200-209.

6. Мигов Д.А., Винс Д.В. Параллельная реализация и имитационное моделирование оценки надёжности сети методом Монте-Карло // Вестник

Томского государственного университета. Управление, вычислительная техника и информатика. 2019. №47. С. 66-74. DOI: 10.17223/19988605/47/8.

7. Калюжный А.В., Шульгин А.Н., Шушаков А.О. Модель и алгоритм оптимального управления ресурсоемкостью стохастических параллельных вычислительных процессов // Труды Военно-космической академии имени А.Ф. Можайского. 2020. №675. С. 45-52.

8. Блинова Н.П., Ломовцева П.Е. Оценка возможности применения параллельных алгоритмов при имитационном моделировании процессов материально-технического обеспечения войск (сил) // Научные проблемы материально-технического обеспечения Вооружённых Сил Российской Федерации. 2021. № 2(20). С. 33-38.

9. Симонов В.В. Оценка эффективности параллельных алгоритмов для моделирования многослойного персептрона // Доклады ТУСУР. 2010. № 1-2 (21). С. 166-171.

10. Яркова О.Н., Чудинова О.С. Инструментарий для моделирования динамики капитала финансовых организаций // Прикладная информатика. 2021. Т. 16. №5. С. 48–65. DOI: 10.37791/2687-0649-2021-16-5-48-65.

11. Яркова О.Н., Реннер А.Г., Пивоварова К.В. Имитационное моделирование финансовых ресурсов коммерческого банка // Прикладная информатика. 2017. Том 12. №4(70). С. 5-14.

12. Краковский Ю.М., Гуляев А.С. Вычисление показателей эффективности при производстве зерна и зернобобовых культур на основе метода Монте-Карло // Инженерный вестник Дона. 2021. №2. URL: ivdon.ru/ru/magazine/archive/n2y2021/6820.

References

1. Zosimov V.V., Bulgakova O.S. Control Systems and Computers. 2023. No. 3 (303). p. 61-68. DOI: 10.15407/csc.2023.03.061.
2. Al'-Khulaydi A.A., Chernyshev Yu.O. Inzhenernyj vestnik Dona. 2011. №2. URL: ivdon.ru/ru/magazine/archive/n2y2011/445.
3. Huang J., Chen L. Foundry. 2006. No. 55. pp. 704-706.
4. Mikov A.I., Mospan N.V. Informatizatsiya i svyaz'. 2013. №2. pp. 141-143.
5. Glinskiy B.M., Rodionov A.S., Marchenko M.A., Karavaev D.A., Podkorytov D.I., Vins D.V. Vestnik Ufimskogo gosudarstvennogo aviatsionnogo tekhnicheskogo universiteta. 2013. T. 17. №5 (58). pp. 200-209.
6. Migov D.A., Vins D.V. Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika. 2019. №47. pp. 66-74. DOI: 10.17223/19988605/47/8.
7. Kalyuzhnyy A.V., Shul'gin A.N., Shushakov A.O. Trudy Voenno-kosmicheskoy akademii imeni A.F. Mozhayskogo. 2020. №675. pp. 45-52.
8. Blinova N.P., Lomovtseva P.E. Nauchnye problemy material'no-tekhnicheskogo obespecheniya Vooruzh. 2021. № 2(20). pp. 33-38.
9. Simonov V.V. Doklady TUSUR. 2010. № 1-2 (21). pp. 166-171.
10. Yarkova O.N., Chudinova O.S. Prikladnaya informatika. 2021. T. 16. №5. p. 48–65. DOI: 10.37791/2687-0649-2021-16-5-48-65.
11. Yarkova O.N., Renner A.G., Pivovarova K.V. Prikladnaya informatika. 2017. T. 12. №4(70). pp. 5-14.
12. Krakovskiy Yu.M., Gulyaev A.S. Inzhenernyj vestnik Dona. 2021. №2. URL: ivdon.ru/ru/magazine/archive/n2y2021/6820.

Дата поступления: 14.04.2024

Дата публикации: 28.05.2024