

Д.А. Гавриш, С.Н. Саранча

Харьковский национальный университет радиотехники, Харьков

МЕТОД РАСПРЕДЕЛЕНИЯ ЗАДАЧ ПРИ ПАРАЛЛЕЛЬНОМ МОДЕЛИРОВАНИИ СЛОЖНЫХ ЦИФРОВЫХ СИСТЕМ В ГЕТЕРОГЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ

Для оптимизации распределенного моделирования необходимо добиться равномерной загрузки каждой рабочей станции в сети. Для этого необходимо составить алгоритм распределения задач по рабочим станциям, который учитывает производительность каждой отдельно взятой рабочей станции, параметры текущей загрузки рабочей станции, трудоемкость каждого параллельного процесса в описании цифрового компонента и потребление памяти каждого процесса.

Ключевые слова: процесс, вычислительная сложность, ресурсоемкость.

Введение

С ростом сложности проектируемых цифровых систем вычислительных ресурсов одной рабочей станции в некоторых случаях становится недостаточно для проведения процесса моделирования. Для решения такой проблемы предлагается использовать набор рабочих станций, соединенных коммуникационной сетью для выполнения распределенного моделирования. В общем случае вычислительная система является гетерогенной по причине разнородности используемого аппаратного и программного обеспечения на каждой рабочей станции. Также неоднородность вычислительных ресурсов объясняется разной степенью загрузки рабочих станций в разные моменты времени. На момент запуска имитационного моделирования отдельные рабочие станции в сети могут выполнять другие пользовательские процессы, либо выполнять моделирование другого компонента, которое было запущено ранее.

Возможность проведения распределенного моделирования электронного компонента основывается на его высокой степени параллелизма. Эта особенность дает возможность моделировать отдельные параллельные блоки в описания компонента на отдельных рабочих станциях с выполнением синхронизации данных между ними. Как правило, вычислительная сложность и необходимый объем памяти для моделирования различных блоков цифровой системы разные, что дает возможность использовать гетерогенную вычислительную систему.

При организации распределенного моделирования возникает проблема выделения параллельных блоков описания модели и оптимального распределения этих блоков между рабочими станциями.

Анализ литературных данных. В настоящее время существуют такие известные технологии распределенного имитационного моделирования, как

SPEEDES (Synchronous Parallel Environment for Emulation and Discrete Event Simulation), PARASOL (Parallel Solution) и HLA (High Level Architecture) [1 – 3].

Для обеспечения процесса распределенного моделирования также существует ряд программных продуктов, таких как Aldec Riviera-PRO. Этот программный продукт позволяет не только проводить распределенное моделирование цифровой системы, а также управлять сразу большим количеством запущенных процессов моделирования на сети компьютеров. Также эта система позволяет централизованно формировать результаты процессов моделирования, предоставлять статистическую информацию, а также результаты функционального покрытия. Использование данного программного продукта позволяет проводить моделирование цифровой системы различной степени сложности, описание которой может быть реализовано с использованием различных языков описания аппаратуры, а также эта система предоставляет большие возможности для масштабирования – в процессе моделирования может использоваться различное количество рабочих станций с различными характеристиками.

В качестве альтернативных средств для запуска множества процессов моделирования на наборе рабочих станций могут использоваться различные GRID системы. В этом случае можно запустить множество процессов моделирования одновременно и централизованно анализировать результаты моделирования, но с другой стороны выполнение распределенного моделирования не представляется возможным.

Постановка задачи. Для выполнения распределенного моделирования используется некое множество рабочих станций под управлением сервера, который осуществляет раздачу заданий на моделирование, где задание представляет собой отдельный процесс.

Каждый процесс характеризуется следующим набором параметров:

- трудоемкость процесса;
- требуемый минимальный и максимальный объем памяти.

Каждая рабочая станция моделирования обладает следующими характеристиками:

- пиковая производительность центрального процессора и подсистемы памяти;
- объем установленной физической памяти и объем доступной памяти файла подкачки;
- текущий объем свободной динамической памяти;
- текущая нагрузка центрального процессора.

Задача равномерного распределения заданий на моделирование решается по-разному в зависимости от следующих условий:

- гомогенная или гетерогенная вычислительная среда;
- статическая или динамически изменяющаяся трудоемкость отдельных заданий.

Для упрощения из дальнейшего рассмотрения исключены вопросы определения производительности подсистемы передачи данных (компьютерной сети), топологической связности процессов, времени, требуемого сервером для выполнения синхронизации процессов, и так далее. Это является предметом дальнейших исследований.

Таким образом, необходимо решить задачу равномерного распределения заданий со статическими характеристиками трудоемкости TN_i , минимального и максимального объема памяти ($RMIN_i$ и $RMAX_i$ соответственно) на разнородные рабочие станции, обладающие различными значениями производительности процессоров PR и объемов доступной физической памяти RAM и файла подкачки SW , текущая нагрузка центрального процессора u . Примем U – коэффициент простоя центрального процессора ($U=1-u$).

Таким образом, все множество процессов требуется разбить на отдельные подмножества с соблюдением указанных условий:

$$\begin{aligned}
 P &= \{P_i\}; \\
 P_j \cap P_k &= \emptyset \quad \text{для всех } j \neq k; \\
 \sum_i TN_{ij} &\approx a \cdot PR_j \cdot U_j; \\
 \sum_i RMIN_{ij} &\leq RAM_j; \\
 \sum_i RMAX_{ij} &\leq RAM_j + SW_j.
 \end{aligned}$$

Метод оценивания производительности рабочей станции

Оценивание производительности вычислительной системы используется для определения сравни-

тельных характеристик рабочих станций в гетерогенной вычислительной среде. Под производительностью рабочей станции будет подразумеваться время выполнения программного обеспечения, предназначенного для тестирования.

При оценке производительности рабочей станции необходимо учитывать много факторов, таких как:

- операционная система;
- установленные драйвера аппаратного обеспечения;
- установленное программное обеспечение;
- оптимизация работы программного обеспечения компилятором;
- характеристики центрального процессора;
- характеристики динамической памяти;
- характеристики дисковой подсистемы;
- характеристики сетевого интерфейса.

Практика показывает, что производительность вычислительной системы однозначно определить невозможно, так как эта оценка зависит от особенностей используемого тестового программного обеспечения. Для получения более достоверной оценки выполняется комплексное тестирование с использованием множества тестов, которые оценивают различные параметры вычислительной системы, влияющие на производительность.

Для выполнения алгоритма распределения заданий по рабочим станциям необходимо единожды выполнить оценивание производительности рабочей станции (далее эти характеристики останутся неизменными), и при каждом запуске этого алгоритма вычислить степень использования центрального процессора и динамической памяти. Это позволяет ускорить скорость работы алгоритма распределения заданий, так как при каждом его запуске нет необходимости запускать бенчмарк.

Описание бенчмарка

Для оценивания производительности рабочей станции использовался набор тестов “Scimark2”. Этот набор тестов использует численные методы, которые чаще всего используются в научных исследованиях. В состав тестов входит:

- Быстрое преобразование Фурье (Fast Fourier Transform – FFT) – выполняет последовательное преобразование 4K комплексных чисел. Этот тест выполняет сложные арифметические операции, перемещение элементов, использует тригонометрические функции. Первая часть алгоритма выполняет инвертирование части битовой информации (не выполняя каких-либо вычислений), вторая часть алгоритма – выполняет вычисления. Вычислительная сложность алгоритма $N \log(N)$.

- Преобразование Якоби для массива размером 100×100 . Эти операции часто используются в чис-

ленных методах решения математических задач, например, решение системы линейных алгебраических уравнений методом Лапласа. В этом тесте элементу двумерного массива $A(i,j)$ присваивается среднее значение 4-х ближайших соседних элементов, умноженное на определенный коэффициент. При выполнении этого теста преобладают операции доступа к памяти.

```

for (int i=1; i < Mm1; i++)
{
    double[] Gi = G[i];
    double[] Gim1 = G[i-1];
    double[] Gip1 = G[i+1];
    for (int j=1; j < Nm1; j++)
        Gi[j] = omega_over_four * (Gim1[j] +
        Gip1[j] + Gi[j-1] +
        Gi[j+1]) + one_minus_omega *
        Gi[j];
}

```

– Метод Монте-Карло для вычисления числа π . Этот метод вычисляет значения интеграла для четвертой круга, уравнение которого описывается следующим образом: $y = \sqrt{1-x^2}, x \in [0,1]$. Алгоритм генерирует случайные координаты, которые входят в квадрат с гранью, равной 1, и вычисляет процент попаданий в круг. Данный тест используется для оценки производительности генератора случайных чисел и времени синхронного вызова функции.

– Умножение разреженных матриц. Для этого метода матрица хранится в сжатом виде. В этом тесте для обработки генерируется разреженная матрица A размером $1,000 \times 1,000$ с 5,000 ненулевых элементов по следующему шаблону:

x	x																					
x	x	x																				
x		x	x																			
x	x		x			x																
x	x	x	x					x														
x	x		x	x						x												
x		x			x	x							x									
x			x		x			x							x						x	
x			x			x							x									x

В каждой строке находится приблизительно 5 ненулевых элементов, которые равномерно распределены между первым столбцом и главной диагональю.

– LU-разложение. Представление матрицы A в виде произведения двух матриц, $A=LU$, где L – нижняя треугольная матрица, а U – верхняя треугольная матрица.

Бенчмарк выполняет 5 различных тестов и измеряет время выполнения каждого теста. Показателем производительности рабочей станции является время выполнения каждого теста. Чем меньше время – тем выше показатель производительности. Результирующий показатель продуктивности является взвешенной суммой полученных оценок, полученных при выполнении каждого теста.

Метод определения трудоемкости процесса

Трудоемкость алгоритма – количество вычислительной работы, требуемой для реализации алгоритма. Так как используемые алгоритмы в параллельных блоках, как правило, имеют много параметров для фиксированного размера входных данных, необходимо оценить его трудоемкость в худшем, среднестатистическом и лучшем случае. В дальнейшем при расчетах будет использоваться среднестатистическое значение трудоемкости. Для оценки использования памяти необходимо проанализировать область декларации каждого блока процесса.

Методы распределения вычислительных заданий при распределено моделировании

Входными данными для алгоритма распределения вычислительных заданий являются:

- оценка производительности каждой рабочей станции PR, полученная в результате проведения тестов (бенчмарка);
 - характеристики динамической памяти (объем физической памяти и файла подкачки RAM и SW) – эта информация позволяет учесть текущую степень загрузки рабочей станции;
 - текущее состояние нагрузки на центральный процессор;
 - статическая оценка трудоемкости выполнения каждого параллельного процесса TN_i ;
 - характеристики использования памяти параллельным процессом $RMIN_i$ и $RMAX_i$.
- Требования к алгоритмам распределения вычислительных заданий:

- $\sum_i RMIN_{ij} \leq RAM_j$ – объема физической памяти должно быть достаточно для минимальных требований памяти параллельного процесса;
- $\sum_i RMAX_{ij} \leq RAM_j + SW_j$ – объема динамической памяти должно быть достаточно для работоспособности системы в самом худшем случае;

– $\sum_i TN_{ij} \approx a \cdot PR_j \cdot U_j$ – суммарная вычислительная трудоемкость параллельных процессов на

каждой рабочей станции пропорционально зависит от производительности центрального процессора умноженного на коэффициент его простоя.

Для упрощения вычислений предлагается отсортировать параллельные процессы $P = \{P_i\}$ в порядке увеличения вычислительной сложности и вычислить коэффициент сложности каждого процесса $C_i \in [0:1]$, разделив его вычислительную сложность на значение максимальной трудоемкости $P_{MAX} = MAX(P_i)$.

Критерии оценивания методов распределения

Так как вычислительные задания имеют различные требования к динамической памяти и различную вычислительную сложность, равномерно распределить эти задания в большинстве случаев невозможно. В качестве основного критерия оценки метода распределения выбирается превышение суммарного значения коэффициентов заданий C значения 1:

$$K = \sum_{i, WU_i > AVG} WU_i,$$

где $WU = \sum_j C_j$ – степень нагрузки рабочей станции, K – коэффициент использования системы,

$$AVG = \frac{\sum_i C_i}{N}.$$

Также критерием оценивания эффективности метода распределения является количество итераций алгоритма. Рассмотрим методы распределения заданий с учетом использования гетерогенной вычислительной системы.

Метод полного перебора

Этот алгоритм имеет самое большое время выполнения, но он гарантировано дает минимальное значение коэффициента K . При этом количество итераций алгоритма фиксированное, и описывается следующим выражением: $O=M^N$, где M – количество заданий, N – количество рабочих станций. Исходя из этого выражения, количество итераций имеет степенную сложность и при большом количестве заданий обладает малой эффективностью. Алгоритм работы данного метода следующий:

1. Вычислить коэффициент использования вычислительной системы K для текущего распределения работ.
2. Сравнить этот коэффициент с полученным ранее более оптимальным коэффициентом, если таковой имеется.
3. Вычислить следующую комбинацию распределения работ по рабочим станциям.
4. Если все комбинации проанализированы – вернуть результат, иначе вернуться к пункту 1.

Модифицированный метод полного перебора

Данный метод позволяет получить такой же результат, как и предыдущий, но с меньшими временными затратами.

В этом алгоритме из рассмотрения отбрасывается большое количество комбинаций, которые заведомо являются неэффективными. Распределение работ считается заведомо неэффективным, если $K \geq MIN(C_i)$. При вычислении K алгоритм анализирует задания от четвертого к первому и пошагово вычисляет нагрузку на каждую рабочую станцию, суммируя значения вычислительной сложности процессов, которые присвоены этой рабочей станции.

Если эта нагрузка превышает значение $MIN(C_i)$, итерация прекращается, все последующие задания присваиваются WS_0 , а текущее анализируемое задание отправляется следующей рабочей станции.

Таким образом, отбрасывается M^j заведомо неэффективных комбинаций, где j – индекс текущего анализируемого задания.

Ускоренный метод распределения

Этот алгоритм предоставляет результат за одну итерацию, при этом значение K не является самым оптимальным, но приемлемым. Этот метод пошагово заполняет рабочие станции, в первую очередь, распределяя самые сложные задачи, а потом, по мере возможности, – более простые. Алгоритм работы этого метода следующий:

1. Отсортировать рабочие станции по мере уменьшения производительности.
2. Отсортировать задания по мере уменьшения трудоемкости.
3. Выбрать рабочую станцию. Пошагово в порядке убывания трудоемкости выбирать задания до тех пор, пока нагрузка на рабочую станцию не превысит вычисленное ранее среднее значение нагрузки.
4. Перейти к следующей рабочей станции и повторить п. 3.
5. Если все задачи распределены по рабочим станциям – выход из метода.
6. На текущей стадии распределение любого задания на произвольную рабочую станцию к превышению приведет к превышению среднего значения нагрузки на нее. Поэтому необходимо отсортировать рабочие станции в порядке уменьшения текущего значения нагрузки.
7. Распределить оставшиеся задачи по рабочим станциям: на менее загруженную станцию более сложную задачу.

Результаты сравнения методов распределения задач по рабочим станциям

Для проведения экспериментов была написана программа, которая автоматизировано генерирует набор рабочих станций с различными характеристиками производительности и текущей занятости и набор задач с различными требованиями к динамической памяти вычислительной сложности. После генерации входных данных последовательно запускается 3 различных метода по распределению задач по рабочим станциям. Так как в экспериментах использовался метод полного перебора, для уменьшения времени работы алгоритма количество рабочих станций и задач ограничено. В эксперименте было проанализировано 64 комбинации. Максимальное количество рабочих станций 8, максимальное количество заданий 8. В худшем случае метод полного перебора проанализировал $8^8=16\ 777\ 216$ комбинаций. В ходе эксперимента были проанализированы 3 различных соотношения между количеством рабочих станций N и количеством задач M:

– $M < N$ – количество рабочих станций больше количества задач. В этом случае, как правило, на каждую рабочую станцию спланировано по одному заданию, некоторая часть вычислительной системы не задействована в процессе моделирования.

– $M \approx N$ – в случае, когда характеристики рабочих станций практически одинаковые, а также при однородных параллельных процессах в описании цифровой системы, ожидается получить самый оптимальный случай для проведения распределенного моделирования.

– $M > N$ – наиболее часто встречаемый случай на практике.

В качестве оценки быстродействия метода предлагается рассматривать количество проанализированных комбинаций I в процессе работы алгоритма. Это поможет различить эффективность работы методов при малой размерности M и N.

Ускоренный метод распределения задач позволяет получить результат за одну итерацию и генерирует только одну выходную комбинацию. В этом случае $I = 1$.

В методе полного перебора количество анализируемых комбинаций определяются следующим выражением: $I=M^N$. Этот метод не является эффективным. Количество итераций при работе этого метода приведено в табл. 1.

При использовании модифицированного метода полного перебора количество итераций заранее определить невозможно, но оно определяется следующим выражением $I \leq M^N$. Количество итераций при использовании этого метода приведено в табл. 2.

Таблица 1

Количество итераций при распределении заданий методом полного перебора

N\M	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8
2	1	4	9	16	25	36	49	64
3	1	8	27	64	125	216	343	512
4	1	16	81	256	625	1296	2401	4096
5	1	32	243	1024	3125	7776	16807	32768
6	1	64	729	4096	15625	46656	117649	262144
7	1	128	2187	16384	78125	279936	823543	2097152
8	1	256	6561	65536	390625	1679616	5764801	16777216

Таблица 2

Количество итераций при использовании ускоренного метода полного перебора

N\M	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8
2	1	4	8	13	21	11	13	1
3	1	4	1	7	41	3	4	160
4	1	3	20	74	51	9	79	155
5	1	17	56	156	107	54	61	460
6	1	12	49	1	101	155	7181	41
7	1	20	113	21	3347	340	2041	1045
8	1	5	38	13	109	40968	76722	2073

В качестве критерия эффективности работы метода используется коэффициент K. При использовании метода полного перебора и ускоренного метода полного перебора, полученные коэффициенты эффективности имеют одинаковое значение. Полученные коэффициенты приведены в табл. 3.

Таблица 3

Коэффициенты эффективности распределения задач K при использовании методов полного перебора и ускоренного метода полного перебора

N\M	1	2	3	4	5	6	7	8
1	0	0,267	0,02811	0,6323	0,4562	0,5417	0,6954	0,632
2	0	0,0911	0,32	0,54865	0,8224	0,2613	0,8122	0,7171
3	0	0,104	0,34042	0,347583	0,15375	0,049969	0,14682	0,08492
4	0	0,1027	0,09629	0,2453	0,1977	0,18086	0,21338	0,120613
5	0	0,058269	1,14285	0,2378	0,40478	0,178	0,37314	0,216494
6	0	0,02618	0,138	0,2842	0,10687	0,11	0,5907	0,2895
7	0	0,0015	0,0415	0,189	0,1457	0,3067	0,2865	0,3096
8	0	0,009898	0,0234	0,1516	0,0967	0,3028	0,2738	0,3654

При использовании ускоренного метода распределения задач полученные коэффициенты эффективности являются приемлемыми и близкими к оптимальным значениям. Полученные коэффициенты приведены в табл. 4.

Таблиця 4
Коефіцієнти ефективності
распределения задач К при использовании
ускоренного метода распределения задач

N/M	1	2	3	4	5	6	7	8
1	0	0,267	0,028	0,6323	0,4562	0,5417	0,6954	0,632
2	0	0,0911	0,32	0,548	0,8224	0,3493	0,8122	0,7171
3	0	0,104	0,34	0,6444	0,519	0,05	0,14682	0,085
4	0	0,152	0,096	0,2453	0,2845	0,36045	0,3362	0,127
5	0	0,0582	0,143	0,45	0,4048	0,178	0,37314	0,5029
6	0	0,0444	0,149	0,576	0,236	0,3995	0,6033	0,364
7	0	0,0224	0,167	0,399	0,2121	0,344	0,6067	0,544
8	0	0,050	0,083	0,337	0,2271	0,644	0,42018	0,4335

Выводы

Для оптимизации процесса распределенного моделирования сложной цифровой системы необходимо оптимально распределить выполнение параллельных блоков по рабочим станциям. Для этого необходимо учитывать много различных факторов, таких как характеристики каждого параллельного процесса (ресурсоемкость, которая определяется вычислительной сложностью и потребностями памяти), а также характеристики каждой рабочей станции (производительность центрального процессора, характеристики динамической памяти, текущая занятость). Рассмотренные методы позволяют распределить задания по вычислительным системам при различных конфигурациях рабочих систем и характеристик процессов, а также при различном соотношении количества процессов и имеющихся рабочих станций. В результате проведенных экспериментов были подтверждены полученные теоретически данные.

В качестве дальнейшего развития предлагается модифицировать данные методы для распределения задач в гетерогенной вычислительной среде с учетом коммуникационных зависимостей между параллельными процессами. Для этого необходимо учи-

тывать разные показатели производительности различных рабочих станций.

Список литературы

1. Миков А.И. Информационные процессы и нормативные процессы в ИТ [Текст] / А.И. Миков. – Книжный дом ЛИБРОКОМ. – 2013. – 256 с.
2. Аврамчук Е.Ф. Технология системного моделирования [Текст] / Е.Ф. Аврамчук, А.А.Вавилов, С.В. Емельянов и др.; под общ. ред. С.В. Емельянова и др. – М.: Машиностроение; Берлин: Техник, 1998. – 520 с.
3. Бигдан В.В. Становление и развитие имитационного моделирования в Украине [Текст] / В.В. Бигдан, В.В. Гусев, Т.П. Марьянович, М.А. Сахнюк // Пр. міжнар. симп. «Комп'ютери у Європі. Минуле, сучасне та майбутнє». – К., 1998. – С. 182-193.
4. Горбачев В.А. Технология моделирования систем [Текст]: учеб. пособие для студ. вузов / В.А. Горбачев. – Х.: Компания СМІТ, 2005. – 162 с.
5. Mikov A.I. Formal Method for Design of Dynamic Objects and Its Implementation in CAD Systems [Текст] / A.I. Mikov; Gero J.S. and F. Sudweeks F.(eds), Advances in Formal Design Methods for CAD, Preprints of the IFIPWG 5.2 Workshop on Formal Design Methods for Computer-Aided Design. – Mexico, 1995. – P. 105-127.
6. Миков А.И. Система оперирования распределёнными имитационными моделями сетей телекоммуникаций [Текст] / А.И. Миков, Е.Б. Замятина, А.Х. Фатыхов // Труды Первой Всероссийской научной конференции «Методы и средства обработки информации». – М.:Изд-во МГУ, 2003. – С. 437-443.
7. Fujimoto R.M. Parallel and Distributed Simulation [Текст] / R.M. Fujimoto // Proc. of the Winter Simulation Conf. – 1999. – P. 122-131.
8. High Level Architecture Object Model Template Specification. Version1.3. [Текст] U.S. Department of Defense. 5 February, 1998 (27 July 1998 Document Release).
9. Колмогоров А. Теория систем. Математические методы и моделирование [Текст] / А. Колмогоров, С. Новиков // сб. статей под общ. ред. А. Колмогорова. – М.: Мир, 1989. – 314 с.

Поступила в редколлегию 12.02.2014

Рецензент: д-р техн. наук, проф. О.Ф. Михаль, Харьковский национальный университет радиоэлектроники, Харьков.

МЕТОД РОЗПОДІЛУ ЗАВДАНЬ ПРИ ПАРАЛЕЛЬНОМУ МОДЕЛЮВАННІ СКЛАДНИХ ЦИФРОВИХ СИСТЕМ В ГЕТЕРОГЕННОМУ ОБЧИСЛЮВАЛЬНОМУ СЕРЕДОВИЩІ

Д.О. Гавриш, С.М. Саранча

Для оптимізації розподіленого моделювання необхідно добитися рівномірного завантаження кожної робочої станції в мережі. Для цього необхідно скласти алгоритм розподілу завдань по робочих станціях, який враховує продуктивність кожної окремо узятій робочої станції, параметри поточної завантаженості робочої станції, трудомісткість кожного паралельного процесу в описі цифрового компоненту і споживання пам'яті кожного процесу.

Ключові слова: процес, обчислювальна складність, ресурсоемність.

TASK MANAGEMENT METHOD FOR PARALLEL SIMULATION OF COMPLEX DIGITAL SYSTEM IN HETEROGENEOUS COMPUTER ENVIRONMENT

D.A. Havrysh, S.N. Sarancha

The method of task management in computer system was proposed to provide distributed simulation of complex model. The criterion of efficiency for methods was established. The considered algorithms take into account the computation complexity of each parallel block in the model. The set of experiments was provided for these methods to compare efficiency. These methods can be used to accelerate simulation process and increase efficiency of simulation with using of distributed simulation protocol.

Keywords: task, computational complexity, resource consumption.