

## Состав и структура подсистемы проведения имитационных экспериментов программно-технологического комплекса имитации для исследования ВП в МВС

В. Д. ЛЕВЧУК, С. Ф. МАСЛОВИЧ, В. В. СТАРЧЕНКО

### Введение

Для исследования вычислительного процесса (ВП) в МВС был разработан программно-технологический комплекс исследований (ПТКИ МВС), использующий базовое программное обеспечение (ПО): монитор измерения производительности Windows; система автоматизации имитационного моделирования MICIC4 [1]; набор процедур обработки данных в среде MS Excel. ПТКИ МВС состоит из следующих блоков: «подготовки данных», «моделирования», «обработки результатов моделирования и принятие решения». В таблице 1 представлена состав и структура ПТКИ МВС. Каждый блок использует библиотеку, содержащую программные модули выполняющие специфические функции по организации процесса исследования ВП в МВС.

Таблица 1 – Структура программно-технологического комплекса исследований ВП в МВС

Блок	Используемое базовое программное обеспечение	Библиотеки	
		Описание библиотеки	Название библиотеки
Получение и подготовка данных для моделирования	Монитор производительности и стабильности Windows	Мониторинг ВП в МВС с целью получения исходных данных для построения моделей РН	LIB. SITEMON
		Формирование моделей РН ВП в МВС	LIB. INVARIANT
Моделирование	СМ MICIC4	Моделирование узла МВС	LIB.UZEL
		Моделирование сети МВС	LIB.NET
		Моделирование сети МВС с РОИ	LIB.ROI
		Моделирование сети с МВС репликацией	LIB.NETREPL
Обработка результатов и принятие решений	MS Excel	Обработка результатов моделирования	LIB.REZ
		Принятие решений	LIB.SOL

Блок «Моделирование» создан с помощью базовой системы моделирования (СМ) MICIC4 и организует имитационное моделирование ВП в МВС. Он компонуется на основе четырех библиотек имитационных моделей (ИМ): «LIB.UZEL» (для разработки ИМ «Узел МВС»), «LIB.NET» (для формирования ИМ «Сеть МВС»), «LIB.ROI» (для построения ИМ «МВС с РОИ»), «LIB.NETREPL» (для формирования ИМ «Сеть МВС с репликацией»). Каждая из этих библиотек состоит из множества программных модулей, являющихся заготовками

ми для формирования ИМ. Помимо модулей организации имитации, каждая из перечисленных библиотек включает модули анализа результатов моделирования, позволяющие разработчику верифицировать и проверять адекватность создаваемой ИМ. Все библиотеки этого блока являются открытыми и доступны для изменений.

## 1 Состав и структура библиотеки имитационного моделирования ВП в узле МВС «LIB.UZEL»

Библиотека «LIB.UZEL» используется для постановки имитационных экспериментов ВП многопроцессорного узла ЛВС. Она состоит из набора модулей, каждый из которых реализует уникальную функцию. Модули библиотеки реализованы в среде СМ MICIC 4, за счет расширения базовых объектов этой СМ. Каждый из модулей представляет собой активность, содержащую алгоритм, имитирующий работу физического или программного устройств многопроцессорного узла ЛВС. Обработка всех запросов реализована на основе принципа поквантового выделения им ресурсов.

Модули «GEN\_DIAL», «GEN\_OTL» и «GEN\_ROI» предназначены для имитации создания различных типов запросов на ресурсы устройств узла: диалоговых, отложенного счета и распределенной обработки. Интенсивности поступления этих запросов ( $\lambda_{\text{диалог}}$ ,  $\lambda_{\text{отл.счет}}$ ,  $\lambda_{\text{рои}}$ ) могут изменяться исследователем. Модуль «GEN\_DIAL» формирует один или несколько запросов на ресурсы CPU<sub>i</sub> и HDD<sub>i</sub> узла. Причем, запросы могут подаваться либо из внешнего файла, или формироваться непосредственно в среде СМ MICIC4. «GEN\_OTL» и «GEN\_ROI» формируют запросы только на основании данных, хранящихся во внешних файлах, и представляют собой длительные требования на ресурсы устройств узла. Состав и структура этих запросов остаются неизменными и представляют собой временные диаграммы {ВД<sub>i</sub>} использования ресурсов CPU<sub>i</sub> и HDD<sub>i</sub>, преобразованных в квантовое представление. Причем запросы диалогового типа имеют больший приоритет по сравнению с запросами последних двух типов. Для предотвращения ситуации монопольного захвата ресурсов узла диалоговых задач в модуле «GEN\_DIAL» имитируется задержка поступления очередного запроса на один или несколько квантов. В случае программного формирования запросов диалогового типа в узле этот модуль функционирует до тех пор, пока в системе находятся запросы отложенного счета или распределенного типа.

Модули «QUEUE\_DIAL», «QUEUE\_OTL», «QUEUE\_ROI», «QUEUE\_HDD», «QUEUE\_IO» имитируют обслуживание очередей соответственно к устройствам CPU<sub>i</sub> и HDD<sub>i</sub> узла МВС. Исходя из того предположения, что все задачи отложенного счета и задачи РОИ должны быть обработаны, очереди могут содержать неограниченное количество запросов.

Модули «SERVER\_CPU» «SERVER\_HDD» реализуют алгоритмы имитации обработки запросов на *i*-ых устройствах CPU и HDD узла МВС. Запросы всех трех типов, поступающие в узел МВС для обработки, представляются набором квантов по использованию устройств его ресурсов (CPU<sub>i</sub> и HDD<sub>i</sub>). Поэтому эти запросы можно представить последовательностью циклов использования (CPU–HDD), с различным количеством квантов использования CPU и HDD в каждом цикле. Модуль «SERVER\_IO» имитирует функции устройства связанного с вводом/выводом данных (на экран, печать и т.п.). Время вывода данных  $\tau_{\text{IO}}$  также задается исследователем. Данный модуль реализует имитирует запроса задач различных типов на время  $\tau_{\text{IO}}$ .

Модуль «OUTSIDE» уничтожает запросы, у которых закончилась имитация циклов CPU-HDD. По умолчанию СМ MICIC4 реализует широко известный алгоритм FIFO обслуживания очередей.

В модуле «DISP» реализуются алгоритмы имитаций стратегий выбора поступающих на устройства CPU и HDD запросов.

Модули «STAT» и «OTKL» фиксируют статистики и отклики ИМ узла МВС. Функции этих модулей расширяют существующие в СМ MICIC 4 функции получения статистической информации. Вызов этих функций осуществляется в активностях устройств.

В модуле «*TRACE*» формируются трассы реализации имитационного эксперимента. Полученная с помощью этого модуля информация также сохраняется во внешний файл и в дальнейшем может быть использована, например, для верификации создаваемой ИМ.

Модуль «*ANALIZ\_TRACE*» анализирует содержание файла трассы, созданного модулем «*TRACE*». В нем реализованы функции анализа графического представления трассы ИМ. С помощью этого модуля можно проследить динамику взаимодействия запросов, поступающих на узел МВС.

## 2 Состав и структура библиотеки имитационного моделирования ВП в сети МВС «*LIB.NET*»

Библиотека «*LIB.NET*» используется при постановке имитационных экспериментов ВП в сети МВС и состоит из программных модулей, реализующих алгоритмы аппаратной и программной составляющей ИМ сети МВС. Алгоритмы их функционирования во многом схожи с алгоритмами многими модулями предыдущей библиотеки. Модули «*GEN\_DIAL<sub>i</sub>*», «*GEN\_OTL<sub>i</sub>*» формируют запросы различных типов на ресурсы узлов сети МВС. В отличие от модулей предыдущей библиотеки, они создают запросы, содержащие набор временных диаграмм {ВД<sub>*i*</sub>} использования ресурсов узлов *UZ<sub>i</sub>*. В модуле «*QUEUE\_OTL*» реализован алгоритм организации абсолютного приоритета диалоговых запросов перед другими типами запросов. Имитация обработки запросов на вычислительном устройстве (ВУ<sub>*i*</sub>) *i*-го узла сети МВС реализована в модуле «*SERVER\_UZ<sub>i</sub>*». Запрос, поступивший на ВУ<sub>*i*</sub>, задерживается на нем на необходимое этому запросу время  $\tau_{calc}$ . В случае поступления в очередь к ВУ запроса с абсолютным приоритетом обработка запроса (отложенного счета или РОИ) прерывается. В его тело записывается остаток времени, необходимого для дообслуживания, и он возвращается в очередь к этому же ВУ<sub>*i*</sub>. После обслуживания ВУ<sub>*i*</sub> диалогового запроса прерванный им запрос другого типа возвращается на дообслуживание. Причем такая ситуация может повторяться с одним и тем же запросом. Модуль «*NETOS*» имитирует дисциплины выбора запросов из очередей для перенаправления их на различные узлы сети МВС. В нем учитывается скорость меж узловых обменов. Модуль «*NET*» имитирует алгоритм работы коммутатора сети. Запрос задерживается на нем на время  $t_{NET} = v_k / v_{NET}$ , где  $v_k$  и  $v_{NET}$  объем запроса и пропускная способность сети МВС соответственно. Модули «*STAT*», «*OTKL*», «*TRACE*», «*OUTSIDE*», «*ANALIZ\_TRACE*» по функционированию аналогичны модулям предыдущей библиотеки. При запуске ИМ УПМ создает и инициализирует объекты моделирования: узлы, сетевое устройство, генераторы запросов, устройства. При активизации генераторов-имитаторов запросов различных типов реализуется параллельное формирование запросов диалогового типа *DIAL<sub>i</sub>* (*шаг 1*) и отложенного счета *OTL<sub>i</sub>* (*шаг 2*). Эти операции реализуются параллельно на всех узлах сети МВС. После формирования запросов, в конце модулей «*GEN\_DIAL<sub>i</sub>*» и «*GEN\_OTL<sub>i</sub>*», производится вызов функций блока *S*, содержащего модули: сбора статистики «*STAT*» (*S1*) и формирования трассы «*TRACE*» (*S2*). Затем запросы поступают в свои очереди к устройствам *UZ* узла (*шаг 3* и *шаг 4*). Если в очереди находится запрос диалогового типа и на *UZ* нет иных диалоговых запросов, то он поступает на обработку на *UZ* (*шаг 13*), иначе ожидает возможности активизации. После обработки на *UZ* запрос *DIAL<sub>i</sub>* уничтожается (*OUTSIDE*) проверяется наличие в системе запросов отложенного счета (*шаг 17*). Если таковых не имеется, генератор диалоговых запросов останавливается и формируются соответствующие отклики имитации (*шаг 18*), и моделирование завершается. В противном случае происходит опять обращение к модулю «*GEN\_DIAL<sub>i</sub>*» узла. На *шаге 7* проверяется наличие прерывания. При этом осуществляется обработка запроса модулем «*NETOS*» (*шаг 8*), который обращается к библиотеке алгоритмов выбора запросов (*шаг 9*). На *шаге 10* реализуется проверка необходимости пересылки запроса отложенного счета на другие узлы сети МВС. Если условие истинно, то запрос отправляется на сетевое устройство, которое перенаправляет его (*шаг 11*) на другой узел сети. В случае, когда пересылки нет, запрос отправляется на *UZ* собственного узла для обработки (*шаг 12*). Если условие *шага 6* истинно, то выполнение запроса прерывается и в его тело перезаписывается информация об

остатке времени, необходимого для дообслуживания  $t_{\text{дообр}ij} = t_{\text{обр}ij} - t_{\text{тек\_обр}ij}$ , где  $t_{\text{дообр}ij}$ ,  $t_{\text{обр}ij}$  и  $t_{\text{тек\_обр}ij}$  – времена дообслуживания, первоначальное время обслуживания, время, уже обслуженное устройством соответственно (шаг 15). На шаге 16 проверяется условие остатка времени дообслуживания. Если оно равно нулю, то запрос уничтожается из модели (шаг 14). В противном случае отправляется в очередь (модуль «*QUEUE\_OTL<sub>i</sub>*») с присвоением более высокого приоритета перед другими запросами отложенного счета, а диалоговый запрос занимает его место обслуживания на устройстве UZ (шаг 13). Если в очереди *QUEUE\_OTL* находится ранее прерванный запрос (условие шага 7 истинно), то он сразу отправляется на обработку на устройство UZ (шаг 12) и на другие узлы сети МВС не направляется. Данный факт объясняется тем, что узел, на котором ранее обрабатывался прерванный запрос, хранит промежуточные данные, которые не пересылаются на другие узлы. После каждой операции по обработке запросов вызываются модули блока S, который реализуют функции сбора статистик имитации и формирование трассы работы ИМ.

### 3 Состав и структура библиотеки имитационного моделирования ВП в сети МВС с распределенной обработкой информации «*LIB.ROI*»

Библиотека «*LIB.ROI*» организует постановку ИЭ с ВП в сети МВС, в которой присутствуют запросы РОИ [2]. Модель состоит из набора модулей, реализующих определенную функцию. Из-за случайного характера времен счета запросов РОИ реализуется метод Монте-Карло. Он требует множественных прогонов ИМ с вычислением характеристик для каждой l-ой реализации сетевого графика задачи РОИ.

Модули «*GEN\_DIAL<sub>i</sub>*», «*GEN\_OTL<sub>i</sub>*», «*GEN\_ROI<sub>i</sub>*», «*QUEUE\_DIAL<sub>i</sub>*», «*QUEUE\_OTL<sub>i</sub>*», «*QUEUE\_ROI<sub>i</sub>*», «*SERVER\_UZ<sub>i</sub>*», «*NET*», «*STAT*», «*OTKL*», «*TRACE*», «*OUTSIDE*» по функционированию аналогичны модулям предыдущих библиотек.

Модуль «*GR\_CONTROL*» выполняет функцию контроля последовательности формирования запросов на ресурсы сети МВС согласно структуре распределенной задачи, имеющей вид сетевого графика. Функции этого модуля вызываются вначале моделирования (в модуле «*GEN\_ROI<sub>i</sub>*») и каждый раз, когда заканчивается обслуживание одного из запросов задачи РОИ. Алгоритм модуля «*GR\_CONTROL*» реализуется последовательностью следующих шагов. На шаге 1 имитируется отправка запросов на ресурсы сети МВС, которые отражают все дуги графа, исходящие из начальной вершины. После обработки j-го запроса (шаг 2) проверяется ситуация по обработке запросов входящих в K-ю вершину (шаг 3). Если это условие не выполняется, то происходит дальнейшее ожидание обработки очередного запроса. В противном случае проверяется условие, что K-я вершина последняя в графе задачи (шаг 4). Если условие истинно, то модуль заканчивает свою работу. Иначе, из K-ой вершины отправляются запросы на обработку на устройства сети МВС. Модуль «*STATROI*» подсчитывает характеристики обработки запросов РОИ. При отправке запроса на обработку на узел фиксируется время его отправки ( $t_{\text{отпр}}$ ), а при окончании – время его окончания ( $t_{\text{окон}}$ ). Разница между ними составляет фактическое время реализации j-го запроса ( $t_{\text{реализ}}$ ). По окончании очередного прогона ИМ эти данные используются модулем «*OTKLROI*» для расчета параметров сетевого графика. По окончанию моделирования модулем «*OTKLROI*» определяются усредненные характеристики: времена выполнения запросов задачи РОИ и критические времена сетевого графика.

### 4 Состав и структура библиотеки имитационного моделирования ВП в сети МВС с репликацией «*LIB.NETREPL*»

Библиотека «*LIB.NETREPL*» обеспечивает постановку ИЭ ВП в ИМ сети МВС с репликацией данных. Особенностью рассматриваемой библиотеки является иерархическая организация представления устройств сети (УЗЕЛ→УСТРОЙСТВА УЗЛА), а также отображение запросов, поступающих для обработки, в виде кортежей. Алгоритмы функционирования модулей этой библиотеки во многом схожи с алгоритмами модулей библиотеки «*LIB.NET*». Отличительные особенности функционирования некоторых модулей состоят в следующем.

Модуль «*GEN\_KORT*» формирует кортежи запросов отложенного счета, поступающих в узлы сети МВС с частотой  $\lambda_{\text{KORT}_i}$ . В модуле «*SERVER\_CPU*» по окончании обработки кортежей запросов отложенного счета происходит формирование результатов их счета ( $Z_{\text{REZ}}$ ), которые в дальнейшем дублируются в базах данных всех узлов сети МВС путем их рассылки. Модули «*QUEUE\_CPU\_REZ*», «*QUEUE\_HDD\_REZ*», «*QUEUE\_NET\_REZ*» имитируют очереди запросов результатов ( $Z_{\text{REZ}}$ ) к устройствам CPU, HDD, NET соответственно. Имитация дисциплины выбора запросов из очередей для обработки их на узлах  $UZ_i$  сети реализуется модулем «*DISP*». Особенностью алгоритма является анализ значений времен выполнения кортежей запросов ( $\tau_{\text{KORT}_j}$ ) и принятие решения по их пересылке на другие узлы сети МВС. При поступлении очередного кортежа в узел сети МВС в этом модуле анализируются ситуация загрузки устройств узла сети (CPU, HDD) и загрузки сетевого устройства. Модули «*QUEUE\_NET\_KORT*» и «*QUEUE\_NET\_REZ*» имитируют создание очереди к сетевому устройству для запросов кортежей и результатов их счета соответственно. Модули «*OTKL*», «*TRACE*», «*OUTSIDE*» аналогичны модулям предыдущих библиотек. Из-за сложности структуры узла  $UZ$ , состоящей из набора устройств, а также наличия регулярных рассылок результатов счета кортежей по узлам сети МВС. Имитация иерархии сети осуществляется с использованием функции CM MICIC 4 Down(), активизирующей механизм обслуживания внутреннего устройства и направляющей запросы вниз по иерархии устройств сети ( $UZ \rightarrow \text{QUEUE\_CPU}$ ,  $UZ \rightarrow \text{CPU}$ ,  $UZ \rightarrow \text{QUEUE\_HDD}$ ,  $UZ \rightarrow \text{HDD}$ ). Технология использования ИМ «Сеть МВС с репликацией» аналогична технологии использования ИМ «Сеть МВС».

### Заключение

Таким образом, рассмотренная авторами подсистема проведения имитационных экспериментов ПТКИ МВС автоматизирует создание и эксплуатацию ИМ как различных типов структур вычислительных систем (МВС и ЛВС), так и различных типов рабочей нагрузки.

**Резюме.** В статье рассматриваются библиотеки обеспечения процесса имитационного моделирования вычислительного процесса многопроцессорной вычислительной системы (МВС) и локальной вычислительной сети (ЛВС). Каждая из этих библиотек состоит из множества программных модулей, являющихся заготовками для формирования имитационных моделей. Помимо модулей организации имитации каждая из перечисленных библиотек включает модули анализа результатов моделирования, позволяющих разработчику верифицировать и проверять адекватность создаваемой имитационной модели.

**Abstract.** Imitating modeling the multiprocessing computing system (MCS) and the local computer network (LCN) computing process maintenance libraries are considered in the article. Each of these libraries consists of a set of program modules that are preparations for imitating model formation. Besides imitation organization modules each of the listed libraries includes modeling result analysis modules allowing the developer to verify and check created imitating model adequacy.

### Литература

1 Левчук, В. Д. Программно-технологические комплексы имитации сложных дискретных систем / В.Д. Левчук, И.В. Максимей. – Гомельский государственный университет им. Ф.Скорины. – Гомель: ГГУ им. Скорины, 2006. – 263 с.

2 Максимей, И. В. Имитационная модель вычислительной системы с распределенной обработкой информации / С.Ф. Маслович, О.М. Демиденко, В.И. Селицкий // Третья международная научная конференция «Сетевые компьютерные технологии»: сб. тр. III Междунар. науч. конф., 17–19 окт. 2007 г., Минск / редкол.: М.К. Буза (отв. ред.), А.Н. Курбацкий [и др.]. – Минск: Изд. центр БГУ, 2007. – С. 67–72.