

Библиотека формирования инвариантов рабочей нагрузки программно-технологического комплекса исследования ВП в МВС

О. М. Демиденко, С. Ф. Маслович, А. Б. Демуськов

Введение

В данной работе сообщается о разработанной системе формирования инвариантов задач рабочей нагрузки (РН), поступающей на обработку в узлы МВС и ЛВС. Эта система является частью ПТКИ МВС, используемого при исследовании вычислительного процесса (ВП) в МВС и ЛВС.

В таблице 1 представлена состав и структура предлагаемой библиотеки «LIB.INVARIANT», реализующей функцию формирования инвариантов РН, используемых в дальнейшем для организации имитационных экспериментов по исследованию ВП в МВС и ЛВС. Библиотека состоит из модулей, каждый из которых реализует специальный функционал.

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

Название библиотеки	Состав программных модулей	
	Назначение	Обозначение
Формирование инвариантов задач РН для имитационных моделей (LIB.INVARIANT)	Анализ содержимого выходного информационного файла результатов мониторинга и формирование файла моделей задач диалогового и счетного типов узла МВС	RN_UZEL
	Графическое представление задач РН узла МВС	RN_UZEL_GRAPH
	Анализ содержимого выходного информационного файла результатов мониторинга и формирование файла моделей задач диалогового и счетного типов сети МВС	RN_NET
	Графическое представление задач РН сети МВС	RN_NET_GRAPH
	Анализ содержимого выходного информационного файла результатов мониторинга и формирование файла модели графа задачи распределенной обработки информации	RN_ROI
	Графическое представление задач РН РОИ	RN_ROI_GRAPH
	Анализ содержимого выходного информационного файла результатов мониторинга и формирование файла моделей кортежей МВС	RN_KORTEG
	Графическое представление кортежей задач РН МВС	RN_KORTEG_GRAPH

1 Состав и структура библиотеки «LIB.INVARIANT»

Библиотека «LIB.INVARIANT» состоит из восьми модулей. Все модули были разработаны на языке программирования Object Pascal в визуальной среде программирования Bor-

land Delphi 7. Модуль «RN_UZEL» анализирует содержимое выходного информационного файла результатов мониторинга и формирует результирующий файл моделей задач диалогового и счетного типов для узла МВС. На рисунке 1 приведена блок-схема алгоритма формирования инвариантов задач РН, реализуемого следующей последовательностью шагов.

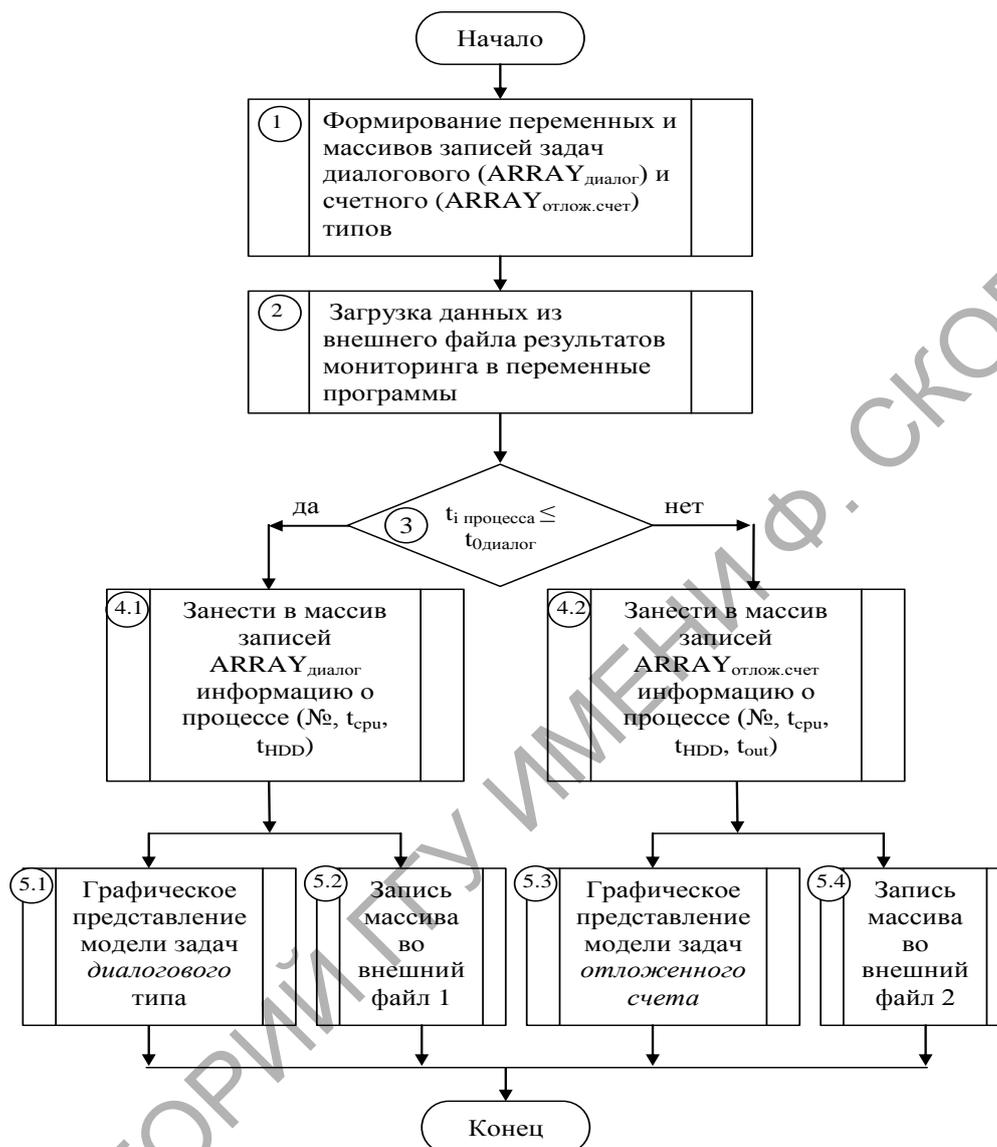


Рисунок 1 – Схема алгоритма формирования инвариантов задач РН

На *шаге 1* алгоритма объявляются массивы записей для задач диалогового типа и для задач отложенного счета.

Шаг 2 осуществляет загрузку данных из содержимого результирующих файлов программ мониторинга. В зависимости от типа загружаемого файла (xml, dbf, txt, csv, tsv) вызывается соответствующая процедура анализа его содержимого и дальнейшей проверки преобразованных данных на принадлежность к определенному типу задач РН.

На *шаге 3* проверяется тип задачи РН (диалог или отложенный счет) согласно условию:

$$t_i \text{ процесса} \leq t_{0 \text{ диалог}}, \text{ где } t_i \text{ процесса}, \quad (1)$$

где $t_i \text{ процесса}$ – время существования процесса i ; $t_{0 \text{ диалог}}$ – параметр, задающий верхнюю границу длительности выполнения задач диалогового типа. В случае истинности условия (1) i – я задача относится к диалоговому типу и происходит переход к *шагу 4.1*, в противном случае осуществляется переход к *шагу 4.2*.

Шаг 4.1 формирует массив записей информацией о задачах диалогового типа (AR-

$RAY_{\text{диалог}}$), шаг 4.2 формируется информация о структуре задач отложенного счета ($ARRAY_{\text{отлож.счет}}$). Эти шаги являются идентичными с точки зрения способов их формирования и отличаются структурой моделей формируемых задач.

На шагах 5.1 и 5.3 возможно графическое представление моделей задач диалогового и счетного типов соответственно. Эти шаги не являются обязательными при формировании инвариантов задач РН, однако представляют собой удобный способ пользовательского анализа. На шагах 5.2 и 5.4 производится запись данных из массивов $ARRAY_{\text{диалог}}$ и $ARRAY_{\text{отлож.счет}}$ во внешние файлы dialog и calc.

Модуль «RN_UZEL_GRAPH» обеспечивает графическое представление моделей задач РН узла МВС и необходим для визуального отображения задач РН при анализе состава задач исследователем.

Модуль «RN_NET» анализирует содержимое выходного информационного файла результатов мониторинга и формирует результирующий файл моделей задач диалогового и отложенного счета для имитационной модели (ИМ) сети МВС. Этот модуль аналогичен модулю «RN_UZEL», за исключением состава параметров и структуры выходной информации.

Модуль «RN_NET_GRAPH» реализует функции графического представления моделей задач РН сети МВС.

Анализ содержимого выходного информационного файла результатов мониторинга и формирование файла модели графа задачи распределенной обработки информации (РОИ) осуществляется модулем «RN_ROI». Структура задачи РОИ представляет собой вероятностный сетевой график (ВСПР) реализации процесса. В результате формируется матрица смежности, описывающая структуру этого сетевого графика.

На шаге 1 формируются необходимые массивы записей для хранения и дальнейшей обработки информации о характеристиках процессов, загружаемых из внешних файлов на следующем шаге (шаг2), сформированных модулями библиотеки «LIB.SITEMON». Шаг 3 необходим для подготовки полученных данных для анализа результатов имитации. На шаге 4 данные о процессах (номер, время начала, время окончания) записываются в во временной массив $ARRAY_{\text{times}}$ в порядке, сформированном на предыдущем шаге 3. Шаг 5 готовит информацию для первого шага к циклу по массиву $ARRAY_{\text{times}}$ для формирования массива записей подзадач задачи РОИ ($ARRAY_{\text{счетРОИ}}$). Дальнейшие шаги 6–11 представляют собой процесс формирования массива $ARRAY_{\text{счетРОИ}}$. После окончания цикла анализа, на шаге 13, формируется матрица смежности (M_{ROI}) из массива $ARRAY_{\text{счетРОИ}}$. На шаге 14 с помощью модуля «RN_ROI_GRAPH» формируется графическое представление задачи РОИ. На шаге 15 формируется внешний файл, содержащий матрицу смежности с информацией о составе подзадач РОИ. Полученный файл в дальнейшем может быть использован программой моделирования для имитации задачи РОИ.

Анализ содержимого выходного информационного файла результатов мониторинга и формирование файла моделей кортежей МВС осуществляется модулем «RN_KORTEG». При изучении динамики развития процессов можно наблюдать, что у некоторых процессов имеются «перекрытия» во время их выполнения. Данные перекрытия означают, что ресурсы узла совместно используются более чем одним процессом. Следовательно, эти процессы связаны между собой и это означает, что они зависимы друг от друга и существует информационная зависимость.

Формирование инвариантов РН во времени выполнения процессов (t_{proci}) основывается на разности времени его окончания (t_{endproci}) и времени его начала ($t_{\text{startproci}}$). Соответственно время выполнения i -го «кортежа» ($t_{\text{корт}i}$) равно разности времени окончания самого позднего m -го процесса ($t_{\text{startproc}m}$), принадлежащего данному «кортежу» и временем самого раннего l -го процесса ($t_{\text{startprocl}}$) из этого «кортежа». Процессам p_1 и p_2 , у которых имеет место перекрытие времен использования ресурсов и интервалов их реализации τ_{pij} на узле присваивается признак «связности» ($\pi_{p_1p_2\dots p_m}$). «Кортежи» процессов снабжаются списками (SPKOR). Структура списка SPKOR стандартизована:

$$SPKOR_k = \{idk; lsp; (p_1, \tau_{p1lj}), (p_2, \tau_{p2lj}), \dots, (p_l, \tau_{plmj})\}, \quad (2)$$

где idk – идентификатор «кортежа»; lsp – текущая длина списка; (p_i, τ_{pij}) – список номеров процессов p_i , входящих в список, и интервалов использования им l_j -го ресурса по заказам i -го процесса. Все процессы, входящие в один «кортеж», реализуются на одном и том же узле МВС и используют одни и те же номера ресурсов (l_j). Допускается, что «кортежи» процессов считаются независимыми между собой и поэтому могут реализовываться в любом порядке, а в случае пересылки и на других узлах МВС. Можно выделить следующие типы «кортежей»: «кортежу» содержит только один процесс (рисунок 2 (а)); «кортеж» содержит все процессы узла (рисунок 2 (б)); «кортеж» содержит часть процессов узла (рисунок 2 (с)). Деление на типы позволяет любой процесс обработки информации отождествить с понятием «кортежа».

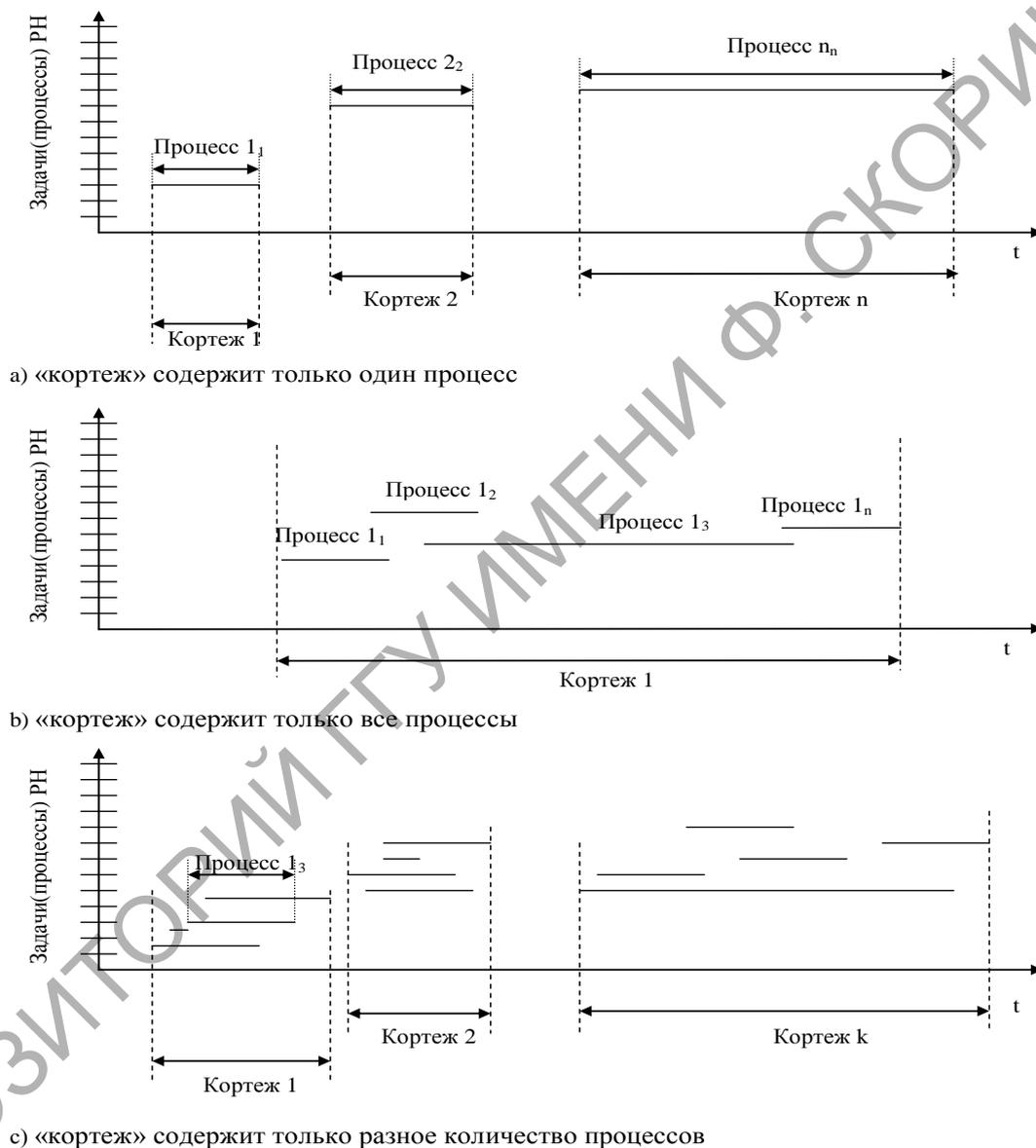


Рисунок 2 – Типы кортежей

Реализацию ВП в ЛВС или МВС можно представить набором множеств списков «кортежей» процессов $\{SPKOR\}$. Поэтому если результаты измерений в ходе натуральных экспериментов на ВС, то они обрабатываются таким образом, чтобы получить множества инвариантов «кортежей» процессов. Для «запитки» кортежей на входе в ИМ в структуру кортежей добавляется указатель типа процесса $type=\{dial; pak\}$ (диалоговый или отложенного счета), поэтому «кортеж» имеет более сложную структуру:

$$SPKOR_k = \{idk; lsp; (p_1, type, \tau_{p1j}), (p_2, type, \tau_{p2j}), \dots, (p_l, type, \tau_{plj})\} \quad (3)$$

Для ИМ «Сеть МВС» структура «кортежей» (3) дополняется типом процесса. Однако, интервалы использования l_j -го ресурса по заказам i -го процесса объединяются в одно значе-

ние (τ_{pj} – время выполнения j -го процесса). поскольку при наличии пересылок кортежей между узлами сети, то к «кортежу» добавляется еще одно поле указатель объема информации (v_k), тогда «кортеж» принимает следующий вид:

$$SPKOR_k = \{idk; lsp; (p_1, type, \tau_{p1}), (p_2, type, \tau_{p2}), \dots, (p_m, type, \tau_{pm}), v_k\} \quad (4)$$

Для ИМ «Сеть МВС с репликацией» структура «кортежей» (4) включает в себя описатель (3) с введением дополнительного поля v_k – объем «кортежа». Поэтому структура «кортежа» для данной ИМ имеет вид:

$$SPKOR_k = \{idk; lsp; (p_1, \tau_{p1lj}), (p_2, \tau_{p2lj}), \dots, (p_m, \tau_{pmlj}), v_k\} \quad (5)$$

Структуры «кортежей» (3) – (5) близки к первоначальной структуре (2). Алгоритма модуля «RN_KORTEG» реализуется следующей последовательностью шагов. На *шаге 1* формируются необходимые массивы записей для хранения и дальнейшей обработки информации о характеристиках процессов и кортежей, загружаемых из внешних файлов. На *шаге 2* осуществляется загрузка данных из внешних файлов системы мониторинга. *Шаг 3* подготавливает первый шаг к циклу по массивам $ARRAY_{\text{диалог}}$ и $ARRAY_{\text{отлож.счет}}$ для формирования массива записей ($ARRAY_{\text{кортеж}}$). На *шаге 4* проверяется ситуация — имел место выход из цикла обработки информации. *Шаг 5* означает условие выделения типов задач (диалоговые или отложенный счет). Запись задач диалогового типа в массив $ARRAY_{\text{диалог}}$ реализуется на *шаге 6*. На *шаге 7* проверяется условие формирования кортежей задач отложенного счета. Отношение очередной задачи к одному и тому же кортежу реализуется по транзитивному принципу благодаря наличию информации о временах начала и окончания выполнения задач отложенного счета. На *шаге 8* увеличивается количество кортежей. Добавление задач отложенного счета в кортеж происходит на *шаге 9*. *Шаг 10* осуществляет увеличение индексов в массивах задач. На *шаге 11.1* и *11.2* формируется графическое представление задач и запись их во внешний файл соответственно. *Шаги 12.1* и *12.2* предназначены для реализации действий аналогичных шагам 11.1 и 11.2, и осуществляются при формировании «кортежей» задач отложенного счета.

Заключение

Таким образом, представленная в данной статье система формирования инвариантов задач РН, поступающей на обработку РН в узлы МВС и ЛВС, позволяет автоматизировать процесс создания инвариантов. Являясь составной частью ПТКИ МВС, средства библиотеки «LIB. INVARIANT» автоматизируют работу для исследователей ВП в МВС и ЛВС при подготовке исходной информации для имитации ВП в этих системах.

Резюме. В статье рассматривается библиотека формирования инвариантов задач рабочей нагрузки, поступающей на обработку в узлы многопроцессорной вычислительно системы (МВС) и локальной вычислительной сети (ЛВС). Эта библиотека является частью программно-технологического комплекса имитации МВС, используемого при исследовании вычислительного процесса в МВС и ЛВС.

Abstract. The library of working loading problem invariants formation arriving on processing in the multiprocessing computing system (MCS) and local computer network (LCN) knots is considered in the article. This library is a part of a programme technological complex of MCS imitation used at research of computing process in MCS and LCN.

Литература

1. Маслович, С.Ф. Имитация динамики взаимодействия оборудования многопроцессорной вычислительной системы с моделями рабочей нагрузки / С.Ф. Маслович. – "Известия ГГУ им. Ф. Скорины №5(50)". – Гомель: ГГУ им. Скорины, 2008 – с. 54–56.