УДК 681.3

КОНВЕЙЕРНЫЕ ПЛАНЫ В МУЛЬВЕРСИОННОМ ПРОГРАММНОМ ОБЕСПЕЧЕНИИ

Прокопенко А.В.

Научный руководитель – доцент Царев Р.Ю.

Сибирский федеральный университет

Программные средства, использующиеся в информационно-управляющих системах (ИУС), обладают всеми свойствами сложных систем. Они имеют общую задачу и цель функционирования; содержат большое количество компонент – модулей, тесно взаимодействующих в процессе решения общей целевой задачи; отличаются сложностью поведения системы, связанной со случайными внешними воздействиями и большим количеством обратных связей внутри системы и пр.

Надежность функционирования программного обеспечения информационно-управляющих систем – один из наиболее значимых критериев разработки программных средств. Существует подход к обеспечению высокого уровня надежности и отказоустойчивости программного обеспечения, основанный на введении временной, информационной, а также программной избыточности.

Одним из наиболее перспективных методов создания программного обеспечения информационно-управляющих систем на основе методологии избыточности является мультиверсионное проектирование. В настоящее время только мультиверсионное проектирование является возможной альтернативой методам тестирования и доказательства правильности программ, обеспечивая высокий уровень надежности исполнения программных компонент. Данный метод гарантирует, что ошибки одной из версий модуля не приведут к нарушению процесса работы информационно-управляющих систем, для которых характерны жесткие требования по надежности.

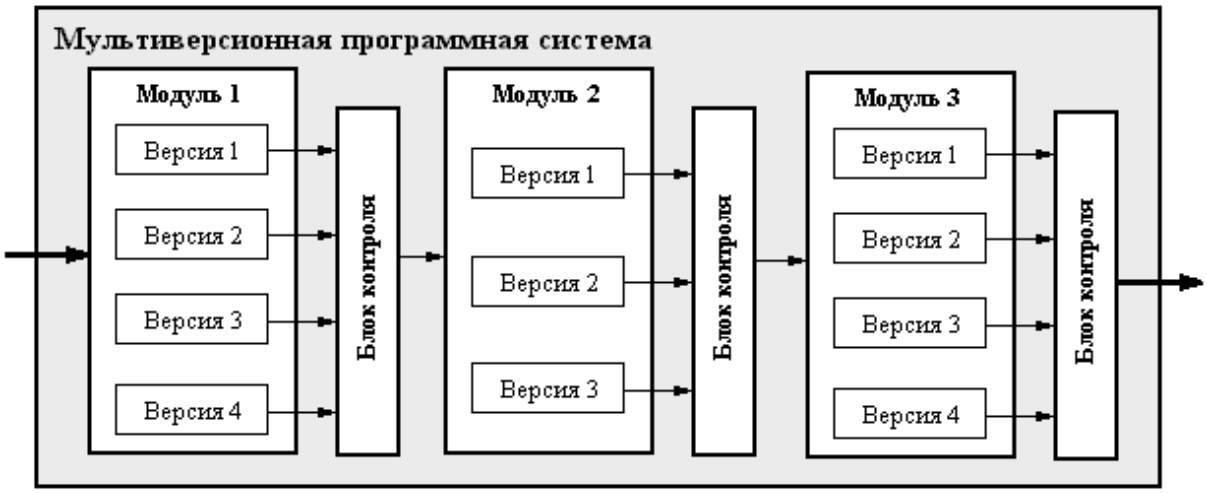

Рис. 1. Модель мультиверсионной программной системы

Мультиверсионное программное обеспечение определяется как независимая генерация $N \geq 2$ функционально эквивалентных программ в соответствии с идентичными исходными спецификациями на проектирование программной системы. Для этих N программ предоставлены средства конкурентного исполнения, по ходу которого в определенных точках контроля («cc-points», точки перекрестного контроля) программами генерируются векторы сравнения («c-vectors», векторы сравнения). Составляющие векторов сравнения и контрольные точки генерации «c-векторов» предварительно определены еще на этапе исходных спецификаций.

Под независимой генерацией программ в данном контексте понимается организация проектирования программной системы с тем, чтобы каждая из N вовлеченных в разработку отдельных программных модулей групп не взаимодействовала с другой по отношению к процессу реализации программной системы. По возможности каждая из групп разработчиков должна использовать различные алгоритмические, инструментальные и языковые средства проектирования.

По аналогии с многоканальными аппаратными средствами, работающими в критических условиях с повышенными требованиями по отказоустойчивости (наземные и бортовые комплексы управления летательными аппаратами, включая космические, компьютеризированные средства управления атомными реакторами и другими опасными для жизнедеятельности людей объектами), также предоставляющими собой $N \geq 2$ аппаратных блоков, необходимо расширение «простого» ПО путем добавления избыточных версий программных блоков.

Главными элементами для методологии мультиверсионного программирования является наличие двух или более избыточных компонент, а также решающего алгоритма, который функционирует в составе среды исполнения и с помощью которого принимается решение о результатах работы программ.

Данный подход к обеспечению высокого уровня надежности и отказоустойчивости программного обеспечения влечет не только увеличение информационной и программной избыточности. Время выполнения задач увеличивается с ростом количества версий. Поэтому минимизация времени выполнения подобного рода задач сама по себе становится актуальной задачей, требующей отдельного внимания.

Одним из вариантов решения данной задачи является применение нескольких процессоров (машин) для параллельного исполнения определенного множества версий мультиверсионного ПО. А решение задачи распределения версий по процессорам можно поискать в производственной среде. Например, для случая, когда количество версий больше количества доступных процессоров, и каждая версия должна пройти определенный этап выполнения на каждом процессоре, для распределения задач во

времени и по процессорам предлагается использовать теорию многомашинных конвейерных планов, модифицировав ее для решения вышеуказанных задач.

Конвейерными планами называют такие планы выполнения задач, в которых более чем один процессор включен в совместное выполнение ряда задач и в которых существует последовательная связь между процессорами. Задача, которую необходимо выполнить, должна быть обслужена одним из процессоров, а потом другими. Это чередование должно соблюдаться для всех задач, входящих в план, никакого требования идентичности процессоров при этом не вводится.

Рассматривая конвейерные планы, часто обращаются к мультипроцессорным планам, потому что в них включается более одного процессора. Для разграничения конвейерных и мультипроцессорных планов используют условие о том, что в конвейерных планах задача должна обслуживаться сначала одной машиной или процессором, а затем другой.

Своим происхождением планы конвейерного типа обязаны производственной среде, в которой работа должна производиться на целом ряде машин, каждая из которых выполняет уникальные, специфические операции. Аналогом в компьютерной среде является требование для задач монопольного использования процессора и каналов ввода-вывода. Чередование этих режимов монопольного использования соответствует прохождению задачи через ряд машин.

Преимущества наличия в информационно-управляющих системах механизмов конвейерного выполнения задач следуют из теории развития производства и преобразования информации.

Конвейерное выполнение типовых задач в распределенных АСУ определяет новый этап в проектировании АСУ, что требует модельно-алгоритмического обеспечения выполнения задач для мультиверсионных систем конвейерного типа.

Наиболее часто цитируемым является решение Джонсоном задачи о двухмашинном конвейерном плане. Алгоритм Джонсона упорядочивает определенное количество задач, одновременно доступных на двухмашинном конвейерном плане, таким образом, чтобы минимизировать максимальное время потока.

Вместо алгоритма Джонсона предлагается так называемое модифицированное упорядочивание Джонсона, основанное на упорядочивании Джонсона. Данный алгоритм применим для ситуации m процессорами типа A и n процессорами типа B. Исследования в данном направлении продолжаются.

Вывод

Применение конвейерных планов для распределения версий по доступным процессорам многопроцессорной системы позволяет ускорить время исполнения мультиверсионного ПО в сложных, критичных по времени системах. При этом, в мультиверсионной программной системе кроме блока контроля, принимающего решение о появлении ошибки в той или иной версии и повышающего общую

надежность системы, появляется еще и блок распределения, который строит конвейерный план исполнения определенного множества версий на доступном количестве процессоров, минимизируя время выполнения данного плана и всей программной системы в целом.

Предлагаемая технология обработки данных также предоставит возможность формирования показателей эффективности работы отдельных процессоров, реализующих конвейерные планы, модулей и мультиверсионной системы в целом. Описанный подход дает следующие преимущества: увеличение пропускной способности конвейерной мультиверсионной системы обработки данных, уменьшение времени и улучшение качества коммуникаций в системе.