

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ НАДЕЖНОГО ШИРОКОПОЛОСНОГО ДОСТУПА К СЕТИ ИНТЕРНЕТ

Мельдер М.И.

научный руководитель д-р техн. наук Ступина А.А.

Сибирский федеральный университет

Сегодня сотни миллионов людей на всем Земном шаре пользуются услугами сети Интернет. Если первоначально Всемирная паутина служила, в основном, как средство передачи данных, то сейчас есть все основания рассматривать Интернет в качестве универсального транспортного средства, используемого в равной степени как для передачи данных, так и для передачи мультимедийного контента. Мультимедиа контент распределен на просторах глобальных сетей передачи данных. Такие сети содержат миллионы высокопроизводительных серверов, сотни миллионов рабочих станций, а также каналов связи, имеющие множество различных характеристик, от ширины и стоимости до надежности и заменимости. Несмотря на то, что мультимедийные услуги имеют свою специфику, а именно большой объем одновременно передаваемых данных, система передачи этих данных ничем не отличается от передачи прочих данных.

Исключительная важность преобразования Интернета в универсальную транспортную систему, обеспечивающую достаточно высокое качество передачи мультимедийной информации, не вызывает сомнений.

Под широкополосными IP-сетями понимается распределительная система, в которой несколько последних километров доступа к пользователю осуществляется по высокопроизводительной сети передачи данных.

Широкополосный Интернет, внедряемый в настоящее время большинством передовых операторов связи, сам по себе не создает некой добавочной стоимости, но обеспечивает возможность внедрения новых услуг, для которых пропускной способности существующих систем передачи было недостаточно. Одной из наиболее перспективных услуг на данной платформе является без сомнения доставка интерактивного мультимедиа контента (т.е. информационного содержания) пользователю.

Большое значение на современном этапе среди множества требований к качеству программного обеспечения (ПО), методам и методикам его разработки и сопровождения приобретают такие требования, как скорость и стоимостная эффективность разработки, внешние средства поддержки исполнения, тестирования и анализа, а также доступность методов и инструментальных средств широкому кругу специалистов, ответственных за доступ к широкополосным мультимедийным услугам.

Мультиверсионное программирование как методология проектирования отказоустойчивых и высоконадежных систем ПО позволяет успешно решать указанные задачи при условии адекватного отражения технологии доступа к широкополосным мультимедийным услугам, в результате чего возможно использовать многолетний опыт и интуицию проектировщиков, ответственных за его функционирование и выполнение целевых функций.

Мультиверсионная отказоустойчивость основана на использовании двух или более версий (или «вариантов») модуля программного обеспечения, исполняемых последовательно или параллельно [2]. Версии используются как альтернативы (с отдельными средствами обнаружения ошибок), в парах (чтобы осуществить обнаружение проверками дублирования) или в больших группах (чтобы маскировать

ошибки через голосование). Использование множественных версий обосновывается предположением о том, что по-разному построенные компоненты (то есть, различными проектировщиками, различными алгоритмами, различными инструментальными средствами проектирования, и т.д.) должны иметь разные ошибки [3]. Поэтому, если одна версия производит сбой на специфическом вводе, по крайней мере, одна из альтернативных версий должна обеспечить корректный вывод.

N-версионное программирование [1] – это мультиверсионная методика, в которой все версии разработаны в соответствии с идентичными основными требованиями, а решение о правильности вывода основано на сравнении всех выводов (рис. 1).

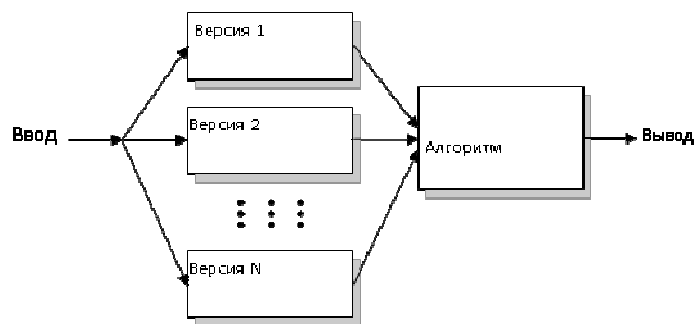


Рисунок 1 –Модель N-версионного программирования

Использование универсального алгоритма выбора решения (обычно голосование) для выбора правильного вывода – фундаментальное различие этого подхода от подхода с блоком восстановления, который требует зависимого от приложения приемочного теста. Так как все версии построены в соответствии с идентичными требованиями, использование N-версионного программирования требует значительно больших затрат при разработке, но сложность (то есть, трудность разработки) не обязательно будет намного большей, чем сложность формирования одной версии. Проектирование выбирающего алгоритма может быть усложнено необходимостью реализовать алгоритм неточного голосования. Многие исследования направлены на разработку методологий, которые увеличивают вероятность достижения эффективного разнообразия в конечном продукте. Фактическое выполнение версий может быть последовательным или параллельным. Последовательное выполнение может потребовать использования контрольных точек, чтобы перезагрузить состояние перед выполнением дополнительной версии.

N-версионное программирование с самоконтролем – это мультиверсионное программное обеспечение, комбинирующее методологию блока восстановления и N-версионное программирование. N-версионное программирование с самоконтролем использует приемочные тесты как показано на рис. 2. В данном случае версии и приемочные тесты разработаны независимо от общих требований. Здесь используются отдельные приемочные тесты на каждую версию, что является основным различием этой модели от методологии с блоком восстановления. Подобно методологии с блоком восстановления, выполнение версий и их испытание может быть сделано последовательно или параллельно, но вывод принят от версии, которая выбрана голосованием и которая прошла приемочный тест. Последовательное выполнение требует использования контрольных точек, и параллельное выполнение требует использования алгоритмов синхронизации ввода и состояний.

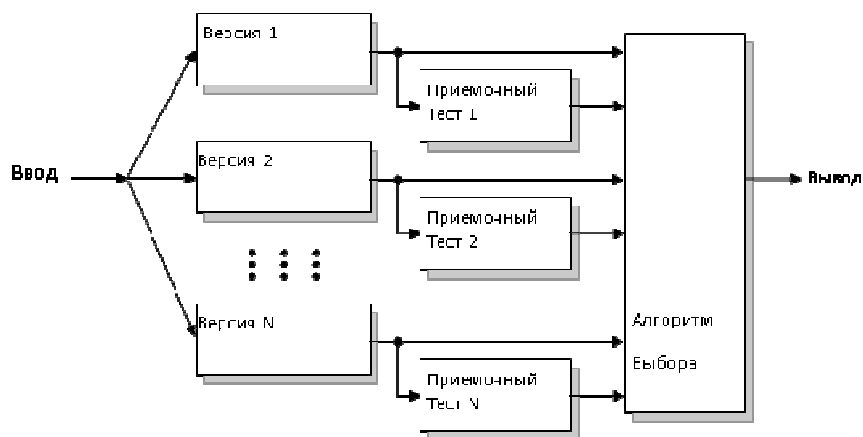


Рисунок 1 – Модель N-версионного программирования с самоконтролем

Благодаря применению N-вариантного подхода к проектированию ПО (вводя избыточность версий программных модулей) достигается улучшение качества программ.

Кроме того, следует отметить, что одно из главных преимуществ N-вариантного ПО – это независимость отказа избыточных версий модулей, так как модули разрабатываются полностью независимыми программистами с использованием различных инструментальных средств и, как правило, в различных операционных средах.

Список литературы

1. Андерсон, Дэвид П. Общественный компьютеринг. Вовлечение людей в науку [Электронный ресурс] / Дэвид П. Андерсон // Лаборатория космических исследований (SpaceSciencesLaboratory). Калифорнийский университет, Беркли.
2. Барский, А.Б. Параллельные процессы в вычислительных системах. Планирование и организация [Текст]. – М.: Радио и связь, 1990. – 256 с.
3. Булочникова, Н.М. Система поддержки сбора и анализа статистики о работе вычислительной системы [Электронный ресурс] / Н.М. Булочникова, В.Ю. Горицкая, А.Н. Сальников. – Методы и средства обработки информации. Труды второй Всероссийской научной конференции. Под ред. Л.Н. Королева. – М.: Издательский отдел факультета вычислительной математики и кибернетики МГУ им. М.В. Ломоносова, 2005.
4. Мельдер, М.И. Мультиверсионные методы формирования программного обеспечения для надежного доступа к широкополосным мультимедийным услугам // 51-ая международная научная студенческая конференция «Студент и научно-технический прогресс» – 2013 – С. 187.