**Параллельное моделирование компьютерной сети в системе Prime**

Астахов А.В., Ладыженский Ю.В.

Донецкий национальный технический университет

Компьютерная сеть представляет собой сложное сочетание прикладных программ, протоколов связи, соединений, потоков данных и алгоритмов маршрутизации. Проектирования сети является довольно сложной задачей, которая требует от разработчиков сопоставлять ожидаемую производительность в сети с затратами на время проектирования и объемы памяти. Одним из подходов к решению данной задачи является использование средств моделирования компьютерной сети. В настоящее время существует множество таких средств, созданных отдельными компаниями и группами разработчиков в научном сообществе, которые применяются в образовательных, коммерческих и специализированных целях. В соответствии с их использованием и доступностью, сетевые средства могут быть классифицированы на 4 группы: аналитические, имитационные, распознавания топологии сети и создания топологии [1]. Средства моделирования имитируют динамическое поведение сетевой модели. Одной из наиболее эффективных систем, позволяющих моделировать компьютерные сети, является система PRIME.

PRIME (Parallel Real-time Immersive Modeling Environment) – это научно-исследовательский проект, главной целью которого является исследование фундаментальных технологий, допускающих крупномасштабное сетевое моделирование, и разработка сетевой среды моделирования в реальном времени [2].

Моделирование в реальном масштабе времени объединяет преимущества и моделирования и эмуляции для выполнения различных имитационных моделей и одновременно взаимодействия с физическим миром. Слово "immersive", часто используемое для описания среды виртуального мира, подразумевает, что для тех, кто использует компьютерную сеть, поведение виртуальной сети невозможно отличить от физической. То есть моделирование должно фиксировать важные характеристики целевой сети и поддерживать взаимодействия с распределенными приложениями.

Основным программным инструментом для исследований является симулятор PRIME, написанный преимущественно на языке C++, и выполняющийся в среде Unix. Для его работы необходим Linux, Mac OS или cygwin. Симулятор состоит из двух компонент: PRIME SSF и PRIME SSFNet.

PRIME SSF (Scalable Simulation Framework) – это механизм моделирования для запуска больших моделей на параллельных машинах (хотя процесс моделирования может выполняться и последовательно). SSF – это масштабируемая среда моделирования, которая является API стандартом для разработки моделей. PRIME SSF реализует SSF спецификацию. Система разработана на основе предыдущих реализаций DaSSF и iSSF. DaSSF – это симулятор, который разработали Jason Liu и David Nicol в Дартмутском колледже. Главной его особенностью является ядро для моделирования. Последняя версия ядра 3.0 поддерживает параллельное моделирование на распределенной памяти.

PRIME SSFNet – это симулятор, разработанный на основе PRIME SSF, который связан с моделированием сетевой инфраструктуры, такой, как Internet. В PRIME SSFNet реализованы необходимые сетевые компоненты: маршрутизаторы, соединения, протоколы и др. По сравнению с другими сетевыми симуляторами, таких как NS, JavaSim, SSFNet достигает значительно лучшей производительности по скорости и при потреблении памяти.

Интерфейс SSF API содержит только 5 базовых класса, которые служат для построения сложных имитационных моделей. Ниже кратко описаны их функции [3].

Объект (entity) – базовый класс, который представляет логический объект моделирования. Объект моделирования – это контейнер для переменных состояния моделирования. Например, в сетевом моделировании, хосты и маршрутизаторы обычно моделируются как объекты. Каждый объект содержит переменные, которые описывают состояние маршрутизатора или хоста, к примеру, размер очереди вывода или оставшееся время на повторную передачу пакета. Предполагается, что пользователь создает новый класс на основе базового класса Entity, содержащего определяемые пользователем переменные состояния, включая процессы и каналы.

Процесс (process) рассматривают как часть объекта, который определяет процесс изменения состояния объекта. Каждый процесс моделирования в SSF представлен экземпляром класса Process или класса, полученного на основе базового. Эти процессы, как и процессы Unix, представляют собой независимые потоки управления.

Входной канал (inСhannel) представляет приемный конец направленной связи между объектами. В SSF связь между объектами обеспечивается передачей сообщений. Объект может получить сообщения с другого объекта только на входной канал. Входной канал принадлежит определенному объекту, который нельзя изменять после создания канала.

Выходной канал (outChannel) – отправная точка связи между объектами. Также как и inChannel выходной канал должен принадлежать объекту. Выходной канал объекта может быть отображен на множество входных каналов, которые принадлежат этому или другим объектам. Сообщение, посланное на выходной канал, будет доставлено системой моделирования на все соответствующий входные каналы.

Событие (event) – базовый класс, который представляет сообщения, посылаемые между объектами через каналы связи. Пользователь определяет его собственные классы на основе базового класса Event. При этом он может добавить определенную информацию, которую нужно передавать между объектами.

В системе существует два метода создания имитационной модели. Первый предполагает использование SSF описания модели на языке DML (Domain Modeling Language). Второй – main-функцию, для программного способа создания модели в дополнение или вместо SSF DML описания.

SSF API универсален для систем, которые могут быть промоделированы как совокупность объектов, связывающиеся между собой через передачу сообщений. Этот тип имитационной модели может быть автоматически отображен на множество процессоров для параллельной обработки. Распределенное моделирование в SSF может рассматриваться как совокупность SSF симуляторов, каждый из которых запускается на многопроцессорной машине в кластере.

**Список литературы**

M.A. Rahman, A. Pakstas, F. Z. Wang. "Network Modelling and Simulation Tools".

MESM’2006, Alexandria, Egypt, August 28-30 2006.

Jason Liu. "Parallel Real-time Immersive Modeling Environment. User’s manual".

Colorado School of Mines, June 9, 2006.

Jason Liu. "Parallel Simulation of Large Scale Wireless Ad Hoc Networks".

Dartmouth College, Hanover, August 15, 2001.