

**РАЗРАБОТКА СИСТЕМЫ РАСПРЕДЕЛЕННОГО ИМИТАЦИОННОГО
МОДЕЛИРОВАНИЯ ДЛЯ РАЗЛИЧНЫХ ОПЕРАЦИОННЫХ СРЕД****В. В. Окольнішников (Новосибирск)****Введение**

Имитационное моделирование – одно из наиболее распространенных и эффективных средств исследования сложных систем и процессов. Область использования методов имитационного моделирования существенно расширилась и включает технические, военные, человеко-машинные, экономические, экологические, социальные и другие объекты исследования. Наиболее эффективно использование имитационного моделирования для решения задач, связанных с принятием решений, управлением, обучением.

Современные задачи исследования таких систем требуют разработки имитационных моделей большого масштаба, требующих для исполнения все большее и большее количество вычислительных ресурсов, которые может предоставить только мультипроцессорная и распределенная вычислительная техника.

С другой стороны, в России наблюдается рост вычислительных мощностей благодаря использованию суперкомпьютеров, кластеров, вычислительных сетей. Примером суперкомпьютера является многопроцессорная вычислительная система МВС–1000/М, находящаяся в эксплуатации в Межведомственном суперкомпьютерном центре РАН (Москва), Сибирском суперкомпьютерном центре СО РАН (Институт вычислительной математики и математической геофизики СО РАН) и других организациях.

С третьей стороны, в мировой практике имеется хорошо развитая теория и технология, получившая название *распределенного имитационного моделирования*. Классическое последовательное имитационное моделирование реализуется на однопроцессорном компьютере. Распределенное имитационное моделирование охватывает весь спектр современной вычислительной техники: суперкомпьютеры, кластерные вычислительные системы, локальные и глобальные сети.

Распределенное имитационное моделирование позволяет решать задачи, требующие большого количества процессорного времени, интегрировать модели, исполняющиеся на различных (в том числе и географически отдаленных) вычислительных средствах, разрабатывать тренажеры для исследования систем и обучения обслуживающего персонала.

Имеется множество программных реализаций систем распределенного имитационного моделирования. Развитие технологии распределенного имитационного моделирования привело к созданию стандарта IEEE, получившего название HLA (High Level Architecture) [1, 2].

Таким образом, в России есть спрос на мультипроцессорные вычислительные ресурсы для имитационного моделирования, имеются предложения таких ресурсов, а также зарубежные разработки систем распределенного имитационного моделирования. Использование распределенного имитационного моделирования тормозится отсутствием или недостаточным предложением доступных систем распределенного имитационного моделирования.

Для заполнения этой "ниши" была разработана система распределенного имитационного моделирования *Мера*. Эта система представляет собой универсальный программный пакет имитационного моделирования, предназначенный для моделирования широкого спектра сложных систем (производственных систем, транспортных систем, систем управления).

Цели разработки системы распределенного имитационного моделирования *Мера*

Целями создания системы распределенного имитационного моделирования *Мера* [3, 4] и ее особенностями являются:

- распределенное исполнение;
- переносимость;
- эффективность;
- расширяемость.

Система *Мера* позволяет решать задачи имитационного моделирования для мультипроцессорных систем и, в частности, для многопроцессорной вычислительной системы МВС–1000/М.

Переносимость системы распределенного имитационного моделирования *Мера* обеспечивается использованием стандартов, которые реализованы в современных вычислительных системах. Такими стандартами являются нити-процессы или процессы с общей памятью (threads), протокол отправки сигналов MPI (Message Passing Interface) [5], стандарт HLA в части спецификации интерфейса, которая определяет взаимодействие с сервисами управления временем RTI.

Эффективное использование параллелизма в распределенном имитационном моделировании ограничивается необходимостью синхронизации модельного времени в параллельно исполняемых процессах имитационной модели.

Эффективность системы распределенного имитационного моделирования *Мера* обеспечивается благодаря наличию библиотеки, реализующей консервативный и оптимистический алгоритмы синхронизации, и возможности ее пополнения.

Расширяемость системы *Мера* обеспечивается вследствие выделения ядра системы, на основе которого могут быть реализованы специализированные системы.

Цели создания системы распределенного имитационного моделирования *Мера* были достигнуты за счет разработки ее архитектуры.

Архитектура системы распределенного имитационного моделирования *Мера*

Программное обеспечение системы *Мера* представляет собой иерархическую структуру, состоящую из следующих слоев:

- ядро;
- последовательная имитационная машина;
- распределенная имитационная машина;
- коммуникационная машина.

Каждый слой программного обеспечения системы распределенного имитационного моделирования *Мера* может послать команду одной из вышестоящей по уровню иерархии машине. Формой отправки команды и получения ответа является вызов интерфейсной функции.

Ядро системы *Мера* – это процессно-ориентированная система дискретного имитационного моделирования, взаимодействие процессов в которой осуществляется с помощью механизма передачи сообщений. Ядро реализовано как объектно-ориентированная библиотека на языках C++ и Java.

Основными функциями ядра являются:

- определение структуры модели;
- передача сообщений;

- задержка процессов на заданное значение модельного времени или поступления сообщений во входные порты;
- выполнение модели.

Перечисленные функции определяют компактный, легко изучаемый, минимально необходимый программный пакет дискретного имитационного моделирования. Этот пакет для практического использования может быть естественным образом расширен библиотеками общего пользования и библиотеками для специализированных приложений.

Последовательная имитационная машина – это система поддержки исполнения (Run time system) модельных процессов в условном модельном времени, исполняемых на одном процессоре. Такие функции ядра, как задержка процессов и выполнение модели, являются командами последовательной имитационной машины.

Распределенная имитационная машина – это система поддержки исполнения модельных процессов в условном модельном времени, выполняемых на различных процессорах мультипроцессорной или мультikomпьютерной вычислительной техники. В ней реализованы базовые консервативный и оптимистический алгоритмы синхронизации модельного времени.

Система команд (интерфейс) распределенной имитационной машины совпадает с интерфейсом HLA в части сервисов управления времени (Time management services), которые синхронизируют продвижение локального модельного времени федератов при исполнении федерации. При наличии библиотеки HLA выполнение большей части функций распределенной имитационной машины может заменить инфраструктура RTI (Runtime Infrastructure) HLA, что делает систему *Мера* более надежной и переносимой, а модели, разработанные с ее помощью, способными к переиспользованию и взаимодействию с моделями, разработанными в рамках других систем имитационного моделирования, соответствующих стандарту HLA.

Коммуникационная машина обеспечивает обмен сообщениями между системами, параллельно исполняющимися на различных процессорах мультипроцессорной или мультikomпьютерной вычислительной техники. Передача сообщений в коммуникационной машине реализована на основе протокола отправки сообщений MPI. Реализация коммуникационной машины использует лишь подмножество MPI – 15 общеупотребительных функций и констант библиотеки MPI.

Для переноса системы *Мера* на вычислительные системы, в которых не существуют или не доступны реализации библиотеки MPI, требуется включить в коммуникационную машину библиотеку, реализующую указанное множество функций средствами операционной системы или систем программирования соответствующей вычислительной системы.

Для автономного однопроцессорного компьютера (персонального компьютера) система *Мера* может использоваться без "внешних оболочек" своего программного обеспечения. Конфигурация однопроцессорной реализации системы *Мера* содержит лишь ядро и последовательную имитационную машину.

На мультипроцессорной или мультikomпьютерной вычислительной системе на каждом процессоре запускается на уровне ядра своя подмодель – структурная часть распределенной имитационной модели и экземпляры последовательной имитационной, распределенной имитационной и коммуникационной машин.

Семейство реализаций системы распределенного имитационного моделирования *Мера*

Система *Мера* имеет семейство совместимых по входному языку реализаций для различных операционных сред.

Однопроцессорные реализации системы *Мера* имеются под *Windows* и *Java*. Они представляют собой пакет дискретного имитационного моделирования в объеме ядра. Наличие специализированных библиотек расширяет функции ядра системы. Имеется возможность задавать масштаб моделирования – отношение единицы модельного времени к единице процессорного времени – для возможности визуализации процесса моделирования.

В среде *Java* на базе ядра системы *Мера* реализована специализированная визуально-интерактивная система моделирования транспортных систем [6]. Она позволяет строить из стандартных блоков транспортную сеть и визуализировать потоки транспортных средств по ней в соответствии с задаваемой дисциплиной обслуживания.

Имеется возможность интерактивно изменять атрибуты транспортных средств и топологию транспортной сети. Пользователь может послать команду транспортному средству, остановить модель, продолжить выполнение модели. Состояние модели запоминается в контрольных точках. Модель может быть остановлена, а затем запущена на выполнение с последнего или одного из промежуточных состояний.

Многопроцессорные реализации системы *Мера* имеются для операционных систем *Linux*, *QNX-4* (операционная система семейства *Unix*), *Windows*.

Система распределенного имитационного моделирования *Мера* реализована для суперкомпьютера МВС-1000/М.

Многопроцессорная вычислительная система МВС-1000/М имеет производительность более 50 GFlops и состоит из управляющего модуля и 16 вычислительных модулей, каждый из которых содержит два процессора DEC Alpha 21264 (833 МГц/4 Мб SLC) и 2 Гб разделяемой оперативной памяти. МВС-1000/М работает под операционной системой *Linux* с использованием библиотеки MPI.

Для работы системы *Мера* требуется библиотека MPI – MPICH2 версии 1.0.1. Система *Мера* была оттестирована на модели передачи сообщений между отправителем и получателем сообщений.

Система *Мера* реализована на локальной сети рабочих станций под *Windows NT*. Коммуникационная машина реализована на базе библиотеки передачи сообщений Bus [7], входящей в состав программного обеспечения автоматизированной системы управления технологическими процессами Северомуйского тоннеля (АСУ ТП СМТ).

С использованием системы *Мера* был разработан интегрированный с АСУ ТП СМТ имитационный стенд, включающий комплекс моделей для отладки и тестирования подсистемы управления АСУ ТП СМТ [8, 9].

Выводы

Целью стандарта HLA является возможность переносимости (переиспользования) моделей, разработанных в рамках различных систем моделирования, исчисляемых сотнями (за рубежом), для уменьшения времени и стоимости разработок новых моделей [10].

Для России более актуальна задача переносимости систем моделирования, исчисляемых единицами, на новые мультипроцессорные и распределенные вычислительные архитектуры для уменьшения времени и стоимости разработок систем моделирования для новых вычислительных архитектур.

Литература

1. **Fujimoto R. M.** Distributed Simulation Systems//Proc. of the Winter Simulation Conference. – 2003. – P. 124 – 134.
2. **Бигдан В. Б., Марьянович Т. П., Сахнюк М. А.** От последовательных к распределенным технологиям в имитационном моделировании//Сборник докладов Первой Всероссийской научно-практической конференции "Опыт практического применения языков и программных систем имитационного моделирования в промышленности и прикладных разработках" (ИММОД 2003). – СПб., – 2003. – Т. 1. – С. 59 – 63.
3. **Окольнишников В. В.** Система распределенного имитационного моделирования//Тр. Первой Всерос. научн. конференции "Методы и средства обработки информации". – М.: МГУ им. М.В. Ломоносова, – 2003. – С. 468 – 473.
4. **Okol'nishnikov V., Rudometov S.** Development of distributed simulation system//Proc. of the Seventh International Conference "Parallel Computing Technologies" (PaCT-2003). – Lecture Notes Comput. Sci. Berlin, – 2003. – Vol. 2763. – P. 524 – 527.
5. **Антонов А. С.** Параллельное программирование с использованием технологии MPI. – М.: МГУ, – 2004.
6. **Okol'nishnikov V.V., Rudometov S. V.** Simulation of Complex Transportation Systems//Proc. of the Second IASTED International Multi-Conference Software Engineering (ACIT-SE). – Novosibirsk, Russia, – 2005. – P. 60 – 64.
7. **Pishchik B. N., Khundoyev A. A., Shevchenko D. O.** Signal transmission in SCADA system//Proc. of the IASTED International Conference "Automation, Control and Information Technology". – Novosibirsk, Russia, – 2002. – P. 65–68.
8. **Окольнишников В. В.** Использование имитационного моделирования при разработке АСУ ТП//Сборник докладов Первой Всероссийской научно-практической конференции "Опыт практического применения языков и программных систем имитационного моделирования в промышленности и прикладных разработках" (ИММОД 2003). – СПб., – 2003. – Т. 2. – С. 178–180.
9. **Окольнишников В. В.** Использование имитационного моделирования при разработке Автоматизированной системы управления технологическими процессами Северомуйского тоннеля // Вычисл. технологии. – 2004. – Т. 9, № 5. – С. 82 – 101.
10. **Fujimoto R. M.** Time Management in the High Level Architecture // Simulation. – 1998. – Vol. 71, No. 6. – P. 388 – 400.