УДК 519.711

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ ПРОЦЕССОВ НАПРАВЛЕННОГО ПОИСКА ОПТИМАЛЬНЫХ РЕШЕНИЙ

В.А. Пепеляев, М.А. Сахнюк, Ю.М. Чёрный

Інститут кибернетики имени В.М. Глушкова НАН Украины,
03680, ГСП, Киев-187, проспект Академика Глушкова, 40,
тел. (380-44)-526-4107, (380-44) 526 3308,
e-mail: emk160ik@gmail.com

Предложен один из возможных подходов к распараллеливанию процессов направленного поиска оптимальных решений, базирующийся на концепции параллельных программ и использовании возможностей интеграции методов имитационного моделирования, метаэвристических оптимизационных стратегий и технологий распределённых вычислений.

In this paper is proposed a possible approach to parallelizing processes of the directed search for optimal solutions based on the concept of parallel programs and the use of the possibilities of integrating simulation techniques, metaheuristic optimization strategies and distributed computing technologies.

Введение

Учитывая широкое распространение высокопродуктивных платформ вычислительной техники, в том числе на основе сетевых и кластерных архитектур, актуальными являются вопросы разработки и эффективного использования схем распараллеливания реальных процессов при создании программного обеспечения, ориентированного на исследование и проектирование сложных стохастических систем. В первую очередь идёт речь о системах, разрабатываемых в таких прикладных областях, как экономика, финансы, маркетинг, транспорт, логистика, биология, медицина, атомная энергетика. Исследуемые здесь системы, как правило, являются стохастическими по своей природе, требуют больших объёмов моделирования и характеризуются критическими значениями показателей времени принятия соответствующих управляющих или проектных решений. Принятые в мировой и отечественной практике подходы к исследованию такого класса систем опираются на использование многофункциональных программных сред и технологий, интегрирующих возможности методов имитационного моделирования, различных оптимизационных стратегий и технологий распределенных вычислений [1 – 4].

Вопросам разработки различных схем распараллеливания и их практического применения посвящены многие зарубежные публикации [5, 6], в том числе в материалах Международной конференции Winter Simulation Conference (WSC'2006 – WSC'2009). Проблема распараллеливания многоаспектна по своей сути: распараллеливание на уровне алгоритмов, на уровне программного обеспечения и средств вычислительной техники (специальные и супермощные компьютеры). Среди отечественных исследований, посвященных исследованию различных аспектов указанной проблемы, следует отметить работы, выполненные под руководством А.Е. Дорошенко, Ф.И. Андона, Г.Е. Цейтлина [7, 8]

В данной работе представлены результаты исследований, посвященные разработке схемы распараллеливания, на которой базируется созданная в Институте кибернетики имени В.М. Глушкова НАН Украины система оптимизационно-имитационного моделирования NEDISOPT_D [9].

Концепция параллельных программ

Одним из классических определений понятия "параллельная программа" является определение, сформулированное в работе Д.Дж. Ивенса [10]. Согласно этому определению любая параллельная программа рассматривается как цепочка последовательно или параллельно исполняемых сегментов (рис. 1).

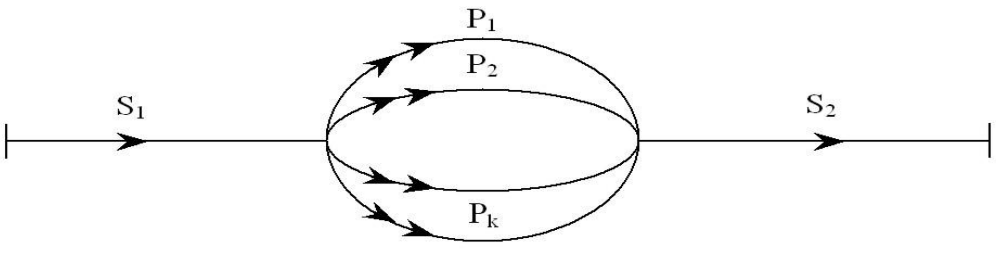

Рис. 1. Сегмент типичной параллельной программы

Здесь отношения предшествования изображены стрелками. Выполнение сегмента S_1 должно быть завершено к моменту начала исполнения P_i (i = 1, 2,…, k), а выполнение всех P_i должно быть завершено к моменту начала исполнения сегмента S_1. Между сегментами P_1, …, P_k отношения предшествования отсутствуют, поэтому они могут исполняться независимо.

Отличительными особенностями представленной на рис. 1 программы являются следующие:

- каждый сегмент программы (S_1, P_1, P_2, …) должен выполняться только одним процессором; когда некоторый процессор берется за выполнение сегмента, остальные процессоры должны быть проинформированы об этом и не должны пытаться его выполнить; аналогично, о завершении выполнения сегмента некоторым процессором также сообщается всем процессорам;
- выполнение сегмента может быть начато только в том случае, если все предшествующие сегменты выполнены, т.е. к моменту начала выполнения сегмента S_2 сегменты P_1, …, P_k должны быть уже выполнены;
- переменые, определяемые в S_1 и используемые в P_1, …, P_k должны быть доступны всем процессорам, а вычисляемые в P_1, …, P_k и используемые в S_2, должны быть доступны процессору, выполняющему S_2. Это же относится и к переменным, определяемым в сегменте S_1 и используемым в S_2.

Любая система программирования, поддерживающая выполнение параллельных программ, должна обеспечивать выполнение как последовательных, так и параллельных сегментов, включая назначение и блокировку соответствующих ресурсов.

Особенности реализации параллельных процессов в системе NEDISOPT_D

Система NEDISOPT_D поддерживает стратегии поиска оптимальных решений на основе экспериментов, реализуемых на однопроцессорных или сетевых архитектурах в формате сессий моделирования [12]. При этом направленный поиск осуществляется на основе специально разработанной схемы распараллеливания, которая в общем случае является аналогом параллельной программы в соответствии с определением Д.Дж. Ивенса.

Основными факторами, определяющими особенности разработки и реализации указанной схемы распараллеливания являются следующие:

- осуществление направленного поиска оптимальных решений в формате оптимизационно-имитационных экспериментов;
- использование концепции "имитационное приложение", согласно которой программные средства поддержки процессов исследования любой сложной системы наряду с определением имитационной модели включают стандартизованные сценарии прогона таких моделей и определение используемых в эксперименте данных;
- использование такой концепции эволюционных вычислений, как "популяция хромосом". В системе NEDISOPT_D принята концепция "популяция хромосом-решений". Гены таких хромосом выступают в качестве характеристик оцениваемых альтернатив. При прогоне имитационных приложений набор генов хромосомы определяет набор факторов имитационной модели. Множество подлежащих оцениванию альтернатив формирует соответствующую популяцию хромосом-решений;
- двухуровневая иерархия параллелизации (на уровне прогонов имитационных приложений и моделей);
- возможность использования различных схем моделирования: последовательное (запуск одного приложения на одном компьютере), распределенное (параллельный запуск нескольких приложений на компьютерах сети), на основе локальной параллелизации (параллельный запуск нескольких приложений на одном компьютере сети);
- использование концепции "область глобальных переменных", к которой имеют доступ процессы, размещенные на компьютерах сети;
- использование нескольких типов оптимизационных стратегий: последовательный перебор вариантов, генетический алгоритм, репликационные прогоны.

Принятая в системе NEDISOPT_D схема поиска оптимальных решений показана на рис. 2, где линии и дуги с одинарными стрелками определяют последовательно исполняемые сегменты (процессы), а с двумя стрелками – исполняемые параллельно. При этом каждый процесс представляется как соответствующий поток (thread). Заметим, что для каждой сессии моделирования априори выбирается соответствующая конфигурация сети, один из компьютеров которой выбирается в качестве главного с номером 0, а остальные используются как периферийные. Информация о заданной конфигурации сети содержится в управляющих параметрах сессии моделирования.

Символы "звездочка" (∗) на рис. 2 определяют точки ветвления последовательно исполняемых сегментов, а треугольники указывают на точки порождения и завершения пучков параллельно исполняемых сегментов (процессов).

Структурная организация всех сценариев системы NEDISOPT_D представляется тремя сегментами.

S_{01} – начальный сегмент главного управляющего сценария системы NEDISOPT_D. В рамках этого сценария вводятся управляющие параметры сессии моделирования и динамически формируется цепочка сценариев оптимизационных стратегий. Сегменты S_{02} и S_{03} поддерживают анализ и обработку результатов сессии моделирования, а также их выдачу соответственно.

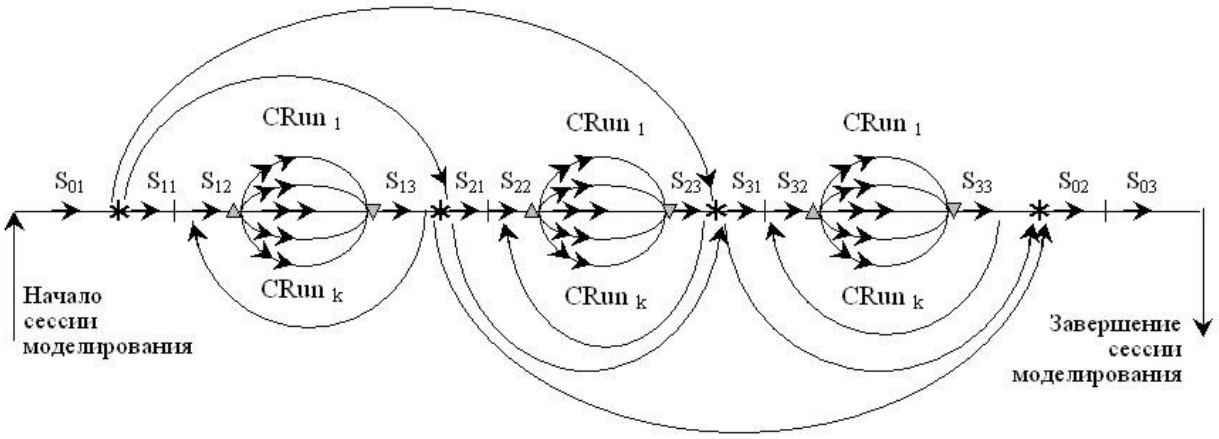

Рис. 2. Схема исполнения последовательно-параллельных сегментов в системе NEDISOPT_D

Между каждой парой символов '*' размещены сегменты, соответствующие реализованным в системе NEDISOPT_D сценариям оптимизационных стратегий.

По аналогии с главным сценарием сегменты S_{i1}, где i = 1, 2, 3, – начальные сегменты соответствующих сценариев. В рамках данных сегментов осуществляется ввод управляющих параметров и формирование начальной популяции хромосом-решений для соответствующих оптимизационных сценариев. Сегменты S_{i2} и S_{i3} обеспечивают управление исполнением и завершением соответствующих сценариев. В рамках сегментов S_{i2} дополнительно осуществляется пересылка характеристик оцениваемых альтернатив в область глобальных переменных. В сегментах S_{i3} дополнительно осуществляется пересылка полученных оценок альтернатив (откликов модели и значений функции цели) из области глобальных переменных в соответствующие поля для откликов имитационной модели. Идентификаторы $CRun_i$ определяют параллельно исполняемые на компьютерах сети прогоны соответствующих имитационных приложений в среде имитатора, представленного системой распределенного дискретно-событийного моделирования NEDIS_D [11].

Все представленные на рис. 2 сегменты составляют программную основу компоненты CONTROL, выступающей в роли оптимизатора системы NEDISOPT_D.

В общем случае число параллельно исполняемых процессов $k \le NComp$, где NComp – число компьютеров в сети, сконфигурированной применительно к конкретной сессии моделирования.

Точки, отмеченные треугольниками – точки блокировки главного управления оптимизатора, а точки, отмеченные перевернутыми треугольниками, выступают в роли точек синхронизации параллельно исполняемых процессов $CRun_i$. Запуск пучков $CRun_i$ осуществляется с помощью процедуры RunModel (), специально разработанной для поддержки принятой в системе NEDISOPT_D схемы распараллеливания. Пока все запущенные параллельно $CRun_i$ не будут завершены, главный поток оптимизатора CONTROL будет заблокирован. Здесь имеют место накладные расходы, обусловленные необходимостью синхронизации процессов $CRun_i$.

Общая схема исполнения $CRun_i$ показана на рис. 3. Каждый процесс $CRun_i$ представлен объектом служебного класса CRun (selfnum), который наследует системный класс SimThread и обладает единственным атрибутом, являющимся номером указанного процесса.

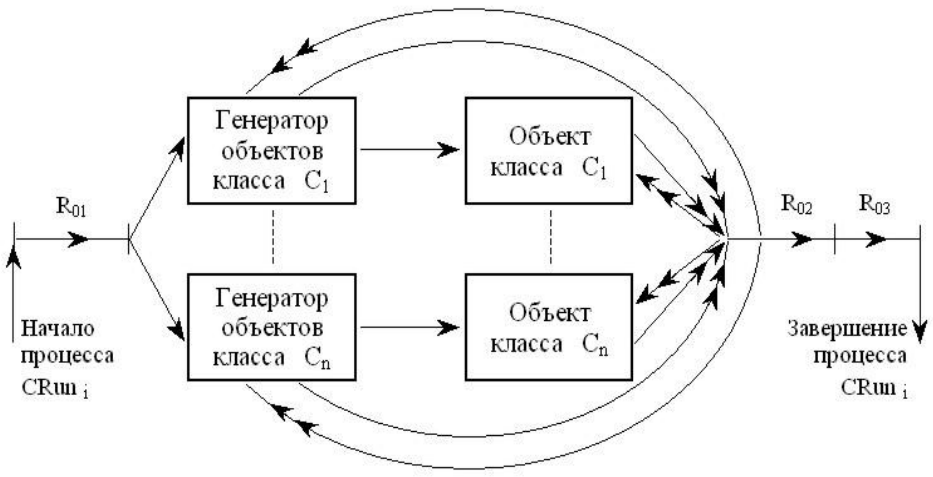

Рис. 3. Схема исполнения процессов $CRun_i$

Здесь сегменты R_{01}, R_{02}, R_{03} включают исполняемые последовательно функциональные модули сценариев имитационных приложений. Процессы, задающие функциональность активных объектов исследуемой системы, исполняются квазипараллельно в виде цепочки дискретных событий. После завершения очередного события управление возвращается в соответствующую точку процесса. Все объекты, которые порождаются в процессе реализации $CRun_i$ должны генерироваться в том же процессе, в котором исполняется имитационное приложение. Для этой цели используется оператор порождения объектов (как активных так и пассивных) в формате new(selfnum), где selfnum – номер соответствующего $CRun_i$.

В рамках сегмента R_{01} осуществляется импорт факторов модели, представленных генами соответствующих хромосом-решений, из области глобальных переменных главного компьютера в рабочую область приложения. Модули из сегмента R_{02} поддерживают формирование календаря и управление квазипараллельным исполнением активных процессов, представленных в имитационной модели. В сегменте R_{03} определяется значение функции цели, осуществляется анализ полученных результатов моделирования и экспорт откликов модели и значение функции цели в область глобальных переменных главного компьютера сети. При этом процедуре RunModel(), управляющей синхронизацией процессов, посылается сигнал о завершении процесса $CRun_i$.

Сетевое размещение компонент системы NEDISOPT_D

Основными компонентами многофункциональной программной среды системы NEDISOPT_D являются:

- резидентная компонента оптимизатора;
- проблемно-ориентированная компонента оптимизатора;
- резидентная компонента имитатора;
- проблемно-ориентированная компонента имитатора.

Резидентная компонента оптимизатора кроме сценариев соответствующих оптимизационных стратегий включает модули поддержки интерфейса между оптимизатором и имитатором как на информационном уровне, так и на уровне потоков. Проблемно-ориентированная составляющая оптимизатора, как правило, разрабатывается специалистами из соответствующей проблемной области и включает определение объектов хромосом-решений, объектов откликов имитационной модели, а также процедуры формирования начальных популяций,. В нее же включаются процедуры выдачи результатов поиска оптимальных решений в специально разработанных форматах. Резидентная и проблемно-ориентированная составляющие оптимизатора всегда размещаются на главном (нулевом) компьютере сети, сконфигурированной применительно к конкретной сессии моделирования. В общем случае компоненты имитатора могут размещаться на любом компьютере сети, включая главный.

Резидентная компонента имитатора представлена системой распределенного дискретно-событийного моделирования NEDIS_D. Для поддержки принятой схемы распараллеливания базовый язык был расширен средствами многоразовых прогонов имитационных моделей и модифицированы модули системы NEDIS_D, которые поддерживают схему таких прогонов.

Проблемно-ориентированная составляющая имитатора включает программу и данные соответствующего имитационного приложения, необходимые для прогона имитационных моделей.

При запуске очередной сессии моделирования система NEDISOPT_D на основании информации, размещенной в специальном host-файле, автоматически формирует перечень компьютеров, которые будут включены в конфигурацию сети и задействованы в процессе сессии моделирования. Начальный запуск системы NEDISOPT_D обеспечивает автоматическую загрузку системы NEDIS_D на все компьютеры сети.

При порождении процессов $CRun_i$ за каждым из них закрепляется компьютер с соответствующим именем. После завершения текущего $CRun_i$ для обеспечения запуска следующего $CRun_i$ система NEDIS_D переводится в начальное состояние (очищается календарь, системные объекты), компьютер освобождается и может быть назначен для других процессов.

В режиме последовательного моделирования оптимизационно-имитационные эксперименты реализуются на одном компьютере. При этом получение оценок для соответствующих хромосом-решений осуществляется в рамках традиционного цикла по перебору хромосом из популяции.

Если число приложений, которые необходимо запустить на данном шаге поиска оптимальных решений, превышает число компьютеров в сети, то для всех "оставшихся" приложений осуществляется запуск $CRun_i$ на нулевом компьютере.

В системе NEDISOPT_D допускается такой режим моделирования, при котором прогон пучков $CRun_i$ осуществляется на одном компьютере. Указанный режим используется на этапах отладки для оценки будущей загрузки сетевых компьютеров и выбора наиболее эффективной конфигурации сети применительно к конкретным сессиям моделирования.

В процессе отладки и верификации средств поддержки разработанной схемы распараллеливания были получены результаты, подтверждающие ее функциональность и эффективность. В таблице представлены временны*е* показатели поиска оптимальных решений для различных конфигураций сети и различного числа параллельных пучков процессов $CRun_i$. Приведены данные, полученные при исследовании стохастической системы "нефтеналивной морской порт". Была проведена серия оптимизационно-имитационных экспериментов для популяции из двадцати хромосом-решений, включающей характеристики различных двадцати проектных альтернатив модернизации указанного морского порта. При этом множество характеристик каждой альтернативы представлялось набором из пяти генов, включенных в состав соответствующих хромосом-решений.

Таблица. Временные показатели схемы распараллеливания

Число приложений	Время поиска оптимальных решений (мин.)		
	1 компьютер	2 компьютера	3 компьютера
1	2,11	–	–
2	4,18	1,93	–
3	5,14	2,85	1,98

Приведенные в таблице оценки получены для сценария последовательного перебора вариантов. Анализ результатов подтверждает эффективность разработанной схемы распараллеливания процессов направленного поиска оптимальных решений.

Заключение

Рассмотрены особенности разработки и реализации схемы распараллеливания процессов направленного поиска оптимальных решений на базе системы распределенного оптимизационно-имитационного моделирования NEDISOPT_D. Программные средства поддержки указанной схемы обеспечивают направленный поиск при различных режимах моделирования (последовательное моделирование на однопроцессорных архитектурах, распределенное моделирование на однопроцессорных и сетевых архитектурах).

Одно из перспективных направлений дальнейшего развития проведенных исследований связано с реализацией разработанной схемы распараллеливания на кластерных архитектурах. Практически важным является расширение проблемных областей приложения указанной схемы, накопление и обобщение опыта ее использования с целью разработки соответствующих рекомендаций, включая методологические и технологические стандарты.

1. *Fujimoto R.M.* Parallel and Distributed Simulation // Proc. of the Winter Simulation Conf. – 1999. –P. 122–131.
2. *Davis D.M., Baer G.D., Gottschalk T.D.* 21st Centure Simulation: Exploiting High Performance Computing and Data Analysis // Interservice/ Industry Training Simulation, and Education Conference. –2004. – Paper N 1517. – P. 1–14.
3. *Fu M.* Optimization for Simulation: Theory and Practice // INFORMS J. on Computing. – 2002. – N 14 (3). – P. 192–215.
4. *April J., Glover F., Kelly J.P., Laguna M.* Practical introduction to simulation optimization // Proc. of the 2003 Winter Simul. Conf. – 2003. – P. 71–78.
5. *Грегори Р. Э.* Основы многопоточного, параллельного и распределенного программирования // Пер. с англ. – М: Издательский дом "Вильямс", 2003. – 512 с.
6. *Миллер Р., Боксер Л.* Последовательные и параллельные алгоритмы. // Пер. с англ. – М: БИНОМ, 2006. – 406 с.
7. *Дорошенко А.Е.* Математические модели и методы организации высокопроизводительных параллельных вычислений – К: "Наук. думка", 2000. – 177 с.
8. *Андон Ф.И., Дорошенко А.Е., Цейтлин Г.Е. Яценко Е.А.* Алгебоаические модели и методы параллельного программирования. – К: Академпериодика, 2007. – 634 с.
9. *Галаган Т. Н., Пепеляев В.А., Сахнюк М. А..* Особенности реализации многослойного сценария распределенного поиска оптимального решения. // Проблеми програмування. – 2008. – № 2–3. – С. 636 – 640.
10. *Ивенс Д.Дж.* Параллельные численные алгоритмы решения систем линейных алгебраических уравнений // В сб. Системы параллельной обработки. – М.: Мир, 1985. – С. 357–384.
11. *Гусев В.В., Галаган Т.Н., Яценко Н.М.* Технологическая система распределенного имитационного моделирования NEDIS_D // Тр. Первой науч.-практ. конф. "Математичне та імітаційне моделювання систем – МОДС'2006". – Киев, 2006. – С. 139–143.
12. *Галаган Т.Т., Пепеляев В.А., Сахнюк М.А., Черный Ю.М., Шваб Н.Д.* О моделях сценариев распределенного поиска оптимальных решений // Компьютерная математика.– 2007.– № 2 – С. 148–156.