

АГЕНТНО-ОРИЕНТИРОВАННЫЙ ПОДХОД К ИМИТАЦИОННОМУ МОДЕЛИРОВАНИЮ СУПЕРЭВМ ЭКЗАФЛОПСНОЙ ПРОИЗВОДИТЕЛЬНОСТИ В ПРИЛОЖЕНИИ К РАСПРЕДЕЛЕННОМУ СТАТИСТИЧЕСКОМУ МОДЕЛИРОВАНИЮ

Б.М. Глинский, А.С. Родионов, М.А. Марченко, Д.И. Подкорытов, Д.В. Винс

В работе рассматривается возможность применения агентно-ориентированной системы имитационного моделирования для решения ряда проблем, возникающих при создании экзафлопсных компьютеров, содержащих десятки и сотни миллионов вычислительных узлов. Предлагается двухуровневая децентрализованная схема управления вычислениями и соответствующая имитационная модель, в которой все вычислительные узлы поделены между областями вычислений, которые контролируются своими локальными управляющими агентами. Головной управляющий агент распределяет между областями поток больших задач и контролирует общие ресурсы. В качестве примера масштабируемого алгоритма рассматривается метод Монте-Карло, перспективный для компьютерного моделирования на экзафлопсных компьютерах. В этом методе существенно то, что чем больше объем выборки из независимых реализаций, тем выше точность оценивания. В работе предлагается генератор базовых псевдослучайных чисел, пригодный для больших расчетов по методу Монте-Карло. При распределении вычислений по узлам допускается возможность реализации различных объемов выборки на различных узлах с использованием статистически оптимального способа осреднения результатов. Объем памяти, доступный каждому вычислительному узлу, и его быстродействие должны быть достаточными для эффективного моделирования реализаций. Данный алгоритм распределенного статистического моделирования асинхронен, и при использовании предлагаемого генератора базовых псевдослучайных чисел масштабируется практически на неограниченное число узлов. Примером масштабируемого приложения распределенного статистического моделирования для современных компьютеров терафлопсного уровня производительности является библиотека PARMONC. Кроме того, в работе рассматривается вариант реализации мультиагентного моделирования для прогнозирования сбоев и отказов вычислительных узлов. Предлагается архитектура динамической системы прогнозирования сбоев, которая состоит из агентов различного назначения, каждый из которых выполняет свою функцию для достижения общей цели.

Ключевые слова: агентно-ориентированное моделирование, экзафлопсные суперЭВМ, методы Монте-Карло, распределенное статистическое моделирование, параллельные вычисления.

Введение

Создание суперЭВМ экзафлопсной производительности потребует решения целого ряда технических проблем, связанных с энергопотреблением, памятью большого объема, температурных проблем, обусловленных функционированием от 10 до 100 млн. вычислительных ядер. Другая проблема будет связана с разработкой и применением системного программного обеспечения, ориентированного на миллионы процессорных ядер с гибридной архитектурой. Следующая проблема связана с надежностью и оценкой реальной производительности вычислительной системы с учетом возможных отказов отдельных элементов системы,

имеющей десятки и сотни миллионов вычислительных ядер. Потребуется алгоритмы, учитывающие стохастические свойства большого числа вычислительных процессов.

Пути решения вышеперечисленных проблем ищет группа экспертов в рамках проекта *IESP (International Exascale Software Project)*, одним из инициаторов которого является Джек Донгарра. По мнению экспертов, системы экзафлопсного уровня потребуют переработки ОС, среды исполнения, компиляторов, библиотек, сред программирования, принципов отказоустойчивости. Если сбои возникают, то выполняемая задача должна уметь их распознавать и уметь восстанавливаться без вмешательства оператора.

Реальные экзафлопсные компьютеры по прогнозам экспертов появятся в районе 2018 года и эксперименты с реальными компьютерами такой производительности в настоящее время невозможны, однако уже сейчас некоторые из вышеперечисленных проблем можно попытаться решить методами имитационного моделирования, которое представляет собой мощное средство анализа сложных систем.

В данной работе рассматривается возможность применения агентно-ориентированной системы имитационного моделирования для решения некоторых проблем, возникающих при создании экзафлопсных компьютеров, содержащих большое количество вычислительных ядер: отработки алгоритмов управления и оптимизации вычислениями в системах экзафлопсной производительности; прогнозирование и предотвращение сбоев и отказов вычислительных ядер. В качестве примера масштабируемого алгоритма рассматривается алгоритм метода Монте-Карло.

Общие принципы организации процесса моделирования

При решении поставленных задач имитационное моделирование может применяться по трём сценариям:

- для анализа предлагаемых алгоритмов управления с использованием модели суперЭВМ на ЭВМ меньшей производительности в режиме чистого моделирования;
- для оперативного выбора наилучших решений по управлению счётом задач с использованием эмулятора суперЭВМ на ЭВМ меньшей производительности в режиме реального времени;
- для оперативного выбора наилучших решений по управлению счётом задач на реальной суперЭВМ в режиме реального времени.

Последние два сценария предусматривают реализацию определённой системы принятия решений, т.е. требуют взаимодействия имитационной модели с некоторой системой искусственного интеллекта. Как известно, существует четыре классических способа такого взаимодействия [1, 2].

Наиболее очевидным представляется способ «включения» или E-способ (от «*Embedding*») – в этом случае или экспертная система включается в имитационную модель или, наоборот, имитационная модель входит в экспертную систему. Во втором случае имитационная модель и экспертная система создаются независимо (как самостоятельные программные системы), а для их связи – программа, передающая при необходимости рекомендации от экспертной системы в модель и обратно, результаты моделирования в качестве новых знаний в экспертную систему. Такой способ связи называется P-способом (от «*Parallel*»). Если экспертная система и имитационная модель предназначены для решения одной задачи и имеют общие данные, то это называется C-способом объединения (от «*Cooperation*»). Наконец, существует способ, при котором экспертная система находится между пользователем и системой моделирования. В этом случае экспертная система в диалоге с

пользователем строит имитационную модель, исполняет ее и интерпретирует результаты для пользователя. Этот способ называется *IFE*-способом (от «*Intelligent Front End*»).

В последние годы существенное развитие получило новое направление в создании интеллектуальных программных систем, связанное с так называемыми интеллектуальными программными агентами (далее для краткости просто программные агенты) [3]. Технология программных агентов оказалась удобной и для создания систем имитационного моделирования [4]. При этом получилось новое качество взаимодействия искусственного интеллекта и имитационного моделирования, а именно их полное слияние: сами моделирующие агенты наряду с изменением переменных состояния принимают решение о выборе дальнейшего поведения системы.

Исходя из этого свойства агентно-ориентированного подхода и учитывая признанный факт что «для моделирования распределённых систем наилучшим образом подходит распределённое моделирование, основанное на передаче сообщений» [5], для реализации имитационных моделей был выбран агентный подход.

Агенты функционируют асинхронно по своим законам, взаимодействуя с другими агентами для достижения общих целей. В процессе функционирования программный агент может изменять как внешнюю среду, так и свое поведение [3, 6]. Распределённый, асинхронный характер поведения «чрезвычайно важен при построении имитационной модели экзафлопсного компьютера, поскольку обеспечить централизованное управление десятками и сотнями миллионов ядер практически невозможно». Выбору агентного подхода к построению моделей поспособствовал и тот факт, что с точки зрения управляющей агентами системы реальные управляющие агенты и агенты-имитаторы неразличимы, т.е. возможен постепенный переход от чистой модели к полунатурной и далее к реальной системе управления вычислениями.

Особо интересен в теоретическом плане вопрос синхронизации событий в модельном времени при взаимодействии агентов в реальном моделировании экзафлопсных систем: известная глобальная проблема распределённого имитационного моделирования систем с дискретными событиями заключается в быстром росте доли затрат на синхронизацию событий при увеличении числа параллельно выполняемых подмоделей. Совершенно очевидно, что классические подходы к синхронизации событий в распределённом моделировании [7] не могут справиться с задачей тотальной синхронизации миллионов и миллиардов событий, планируемых на коротких интервалах времени. При использовании мультиагентного подхода дополнительные затраты связаны с обеспечением работы агентных платформ и взаимодействия между агентами. При большом числе агентов, занятых в моделировании (в случае установления соответствия «одно вычислительное ядро» – «один агент» их число в модели может достигать до сотен тысяч и миллионов) малое число платформ означает большой размер таблиц для сервиса «белых» и «жёлтых» страниц и, соответственно, больших затрат на поиск, а увеличение числа платформ ведет к росту числа межплатформенных обменов, которые требуют больше времени, чем обмены внутри одной платформы. Нахождение оптимального соотношения между числом агентов и платформ, а также реально достижимого числа агентов в модели, требует дополнительных исследований и экспериментов. При этом локальные имитационные модели, используемые для принятия локальных же решений в реальном масштабе времени, могут выполняться классическим образом с использованием временных календарей событий. Моделирование же экзафлопсных систем в целом, как и реальное управление ими, требует разработки принципиально новых подходов, скорее всего допускающих наличие определённой погрешности в управлении событиями.

Одно из возможных решений по повышению эффективности моделирования заключается в допущении потери отдельных событий вследствие несовершенных либо неполных откатов по времени при получении «сигналов из прошлого». Такое возможно если прене-

бредь тотальной синхронизацией событий и ограничиться предельной глубиной хранения состояний в подмоделях, т.е. заданным объёмом буфера хранения состояний. Такой подход избавляет от многих обменов. Подобное допущение возможно при сборе статистики, когда потеря от долей процента до нескольких процентов наблюдений несущественно при определении средних значений, что хорошо согласуется с многими задачами, решаемыми методами Монте-Карло. С другой стороны, подобное допущение может привести к грубым ошибкам при решении вычислительных задач с жёсткими условиями, например при моделировании непрерывно-дискретных систем. Тем самым ограничивается класс алгоритмов, для которых возможно детальное моделирование.

В части моделирования среды исполнения компромиссное решение заключается в нагружении одного агента моделированием совокупности (группы, кластера) ядер, занятых выполнением одной ветви большой параллельной программы или целой программы, требующей относительно небольших ресурсов, например системной программы, обеспечивающей внешний интерфейс. За счёт усложнения поведения агента, который теперь должен отслеживать состояния и взаимодействия контролируемых им ядер моделируемой ВС получаем уменьшение количества агентов модели и, соответственно, снижение затрат на их регистрацию, поиск и взаимодействие.

Кроме этого, предлагается естественная схема децентрализации управления вычислениями и соответствующей имитационной модели (рис. 1).

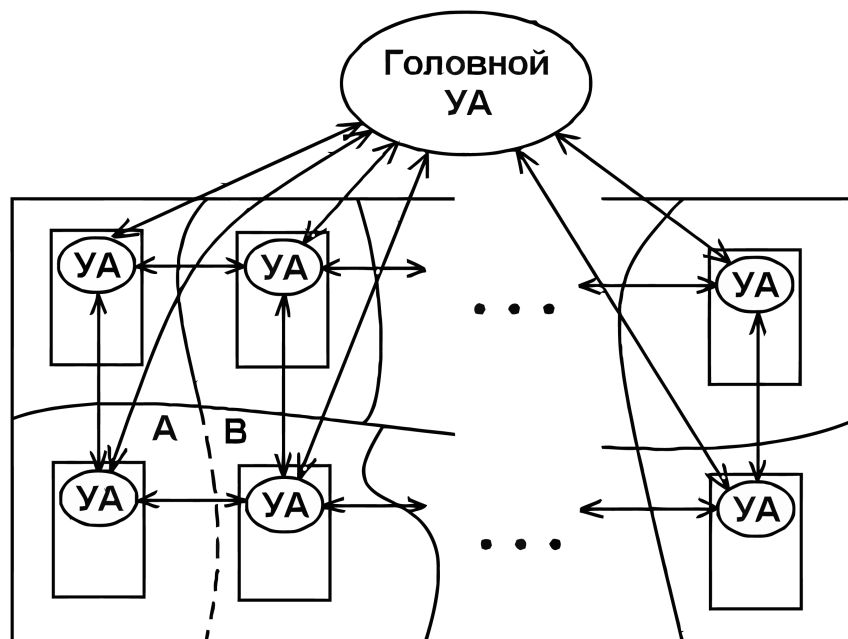


Рис. 1. Двухуровневая схема управления моделью вычислений

Все вычислительные узлы поделены между областями вычислений, управляемыми своими управляющими агентами (УА). Головной УА (ГУА) распределяет между областями поток больших задач и управляет общими ресурсами. Локальные УА (ЛУА) управляют потоками мелких задач, поступающих непосредственно в управляемые области вычислений, и принимают на счёт большие задачи от ГУА. ЛУА взаимодействуют с соседними ЛУА, сообщая о состоянии используемых ресурсов и, возможно, запрашивая недостающие ресурсы. При отдаче ресурсов соседям ЛУА сохраняет за собой область заданного объема, не допуская их монополизации. Несколько областей могут объединяться для решения одной задачи по требованию ГУА (области А и В на рисунке), при этом по завершению вычис-

лений все соответствующие ЛУА снова получают право управления своими областями. В процессе функционирования ЛУА периодически (а также по требованию) передаёт ГУА информацию о состоянии управляемой области (занятые и свободные ресурсы, перспективы освобождения ресурсов, занятость в совместных с соседними областями вычислениями и т.п.).

Для принятия решений по управлению УА любых уровней могут исполнять вспомогательные имитационные модели со своими собственными локальными календарями событий. При моделировании взаимодействия между областями вычислений может требоваться синхронизация модельного времени с откатом по времени одной или нескольких (зависит от масштаба взаимодействий) из соответствующих подмоделей. Для минимизации числа откатов ГУА периодически запускает программу синхронизации локальных календарей. Запуск этой программы возможен и в экстренном порядке при превышении порога числа откатов за заданный период времени. В реальной системе это соответствует приостановлению потоков новых задач, досчитыванию запущенных и перезапуску основного процесса управления.

Реализация функциональных агентов моделирующей системы естественным образом зависит от конфигурации моделируемой вычислительной системы, однако общий набор этих агентов достаточно стабилен и с необходимостью включает агенты управления памятью, агенты, соответствующие вычислительным узлам и ядрам и агенты управления внешними устройствами. ЛУА и ГУА, каждый на своём уровне, для реализации функций управления моделью взаимодействуют со вспомогательными агентами среды моделирования, обеспечивающими синхронизацию событий, сбор и хранение данных по поведению модели и средства восстановления состояния модели при сбоях.

Пример организации конкретной модели рассмотрен ниже.

Имитация загрузки экзафлопсной суперЭВМ расчетными задачами

В рамках разрабатываемого агентно-ориентированного подхода предлагается имитировать загрузку вычислительной системы решением тестовых задач с использованием перспективных численных методов.

В настоящее время ведущими специалистами по вычислительной математике высказывается убеждение в том, что в ближайшем будущем в области компьютерного моделирования будут широко применяться вероятностные имитационные модели и методы Монте-Карло (методы численного статистического моделирования) [8].

Под численным статистическим моделированием обычно понимают реализацию с помощью компьютера вероятностной модели некоторого объекта с целью оценивания изучаемых интегральных характеристик на основе закона больших чисел [9]. При этом чем больше объем выборки, составленной из смоделированных независимых реализаций, тем выше точность оценивания, причем статистическая погрешность убывает обратно пропорционально квадратному корню от числа реализаций.

Как правило, при параллельной реализации необходимый объем выборки базовых случайных чисел очень велик, поэтому целесообразно использовать «длиннопериодные» псевдослучайные последовательности. А именно, для решения т.н. «больших задач» по методу Монте-Карло предлагается использовать генератор следующего вида [8]:

$$u_0 = 1, \quad u_n \equiv u_{n-1}A \pmod{2^{128}}, \quad \alpha_n = u_n 2^{-128}, \quad n = 1, 2, \dots,$$

где

$$A \equiv 5^{100109} \pmod{2^{128}}.$$

Последовательность чисел $\{\alpha_n\}$ является периодической, длина ее периода равна $L = 2^{126} \approx 10^{38}$.

Для распределения псевдослучайных чисел между разными вычислительными ядрами предлагается следующий порядок их использования. Последовательность $\{u_n\}$ предварительно разбивается на подпоследовательности длины μ , начинающиеся с чисел $u_{m\mu}$, $m = 0, 1, 2, \dots$. Разные подпоследовательности используются на разных ядрах. Значение «прыжка» генератора μ должно выбираться из соображений, чтобы μ псевдослучайных чисел хватало для моделирования на каждом ядре. Ясно, что для метода вычетов начальные значения $u_{m\mu}$ указанных подпоследовательностей получаются по формуле

$$u_{(m+1)\mu} \equiv u_{m\mu} A_\mu \pmod{2^{128}}, A_\mu \equiv A^\mu \pmod{2^{128}}, \alpha_{m\mu} = u_{m\mu} \cdot 2^{-128}. \quad (1)$$

Этот параллельный генератор был проверен с помощью содержательных статистических тестов [8] и успешно используется в ряде институтов СО РАН на протяжении последних десяти лет.

С целью понижения трудоемкости моделирование независимых выборочных значений можно распределить по разным вычислительным ядрам. Очевидно, что главным критерием осуществимости такой параллельной реализации является возможность «поместить» вычислительную программу моделирования выборочного значения в память каждого ядра.

При распределении вычислений по ядрам следует допускать возможность реализации различных объемов выборки на различных ядрах с использованием статистически оптимального способа осреднения результатов по формулам вида

$$\bar{\zeta} = \frac{\sum_{m=1}^M n_m \bar{\zeta}_m}{\sum_{m=1}^M n_m},$$

где M – число ядер, n_m – объем выборки для m -го процессора, $\bar{\zeta}_m$ – соответствующее среднее значение. Таким образом, можно добиться обратной пропорциональной зависимости величины трудоемкости случайной оценки от числа ядер (при условии, что используемые ядра имеют одинаковую производительность) [8].

Описанную методику параллельного моделирования можно легко модифицировать с целью эффективной оценки функционалов, зависящих от параметра $x \in X$:

$$\varphi(x) \approx E\zeta(x; \omega).$$

При этом моделируемое распределение случайной величины ω может зависеть, а может и не зависеть от параметра x . Аналогичную методику можно также использовать для оценки функционалов, возникающих в результате осреднения базовых функционалов по случайному параметру σ задачи, например по случайной плотности среды. Вероятностное представление при этом имеет вид двойного математического ожидания

$$\varphi \approx EE\zeta(\omega, \sigma).$$

Примеры подобных задач приведены в [8].

В зависимости от условий задачи и параметров алгоритма статистического моделирования применяются разные методики параллельной реализации. Метод «распределенного статистического моделирования» состоит в распределении независимых реализаций ансамбля тестовых частиц по отдельным ядрам с периодическим осреднением полученных результатов по статистически эффективной формуле. При этом обмен между ядрами сводится к минимуму и в самом простом случае состоит лишь в начальном и финальном обмене данными.

Распределенное статистическое моделирование является примером идеально масштабируемого параллельного алгоритма. Возможные отказы компонент вычислительной системы и связанные с этим потери данных приводят лишь к отсутствию соответствующих смоделированных выборочных значений, что компенсируется расчетами на других вычислительных узлах.

Предлагаемый агентно-ориентированный подход к построению имитационной модели экзафлопсного компьютера, не требующий глобального централизованного управления, хорошо соответствует методике распределенного статистического моделирования [8, 10]. При распараллеливании статистического моделирования вычисление независимых реализаций случайной величины распределяется по свободным вычислительным ядрам. Объем памяти, доступной каждому вычислительному ядру, и его быстродействие должны быть достаточными для эффективного моделирования реализаций. Такой алгоритм распределенного статистического моделирования, при правильно выбранном параллельном генераторе базовых случайных чисел, масштабируется практически на неограниченное число ядер [8].

Количество ядер, участвующих в расчетах, влияет на совокупный объем выборки из смоделированных независимых реализаций. В случае отказа какого-либо вычислительного ядра, участвующего в статистическом моделировании, пропавшие данные могут быть скомпенсированы расчетами на других ядрах. Таким образом, распределённое статистическое моделирование представляет собой расчетный метод, хорошо защищённый от отказов вычислительных узлов системы.

Целесообразно осуществлять периодическую пересылку результатов промежуточного осреднения реализаций, независимо полученных на загруженных ядрах, на выделенные ядра, объединенные в иерархическую структуру. Выделенные ядра будут периодически получать переданные им данные и осреднять их, передавая затем результаты на ядро, соответствующее вершине иерархической структуры (см. рис. 2). Рассчитанные на последнем ядре осредненные значения будут соответствовать выборке, полученной совокупно на всех ядрах. Отметим, что при распределении нагрузки расположение загружаемых ядер в топологии вычислительной системы не играет большой роли, поскольку обмен данными между ядрами по необходимости осуществляется достаточно редко.

Распределенное статистическое моделирование на разных вычислительных ядрах производится в асинхронном режиме. Отправка и получение результатов статистического моделирования также осуществляется в асинхронном режиме. К расчетам возможно подключать свободные вычислительные ядра по мере их освобождения от других задач и эффективно учитывать получаемые на них результаты.

Остановка распределенного статистического моделирования на всех загруженных ядрах (или на части загруженных ядер) возможна в любой момент времени и его возобновление можно начать в удобное время, причем данные расчетов, полученные до остановки, будут эффективно учтены при возобновлении счета. Повторный расчет приведет к получению реализаций случайной величины, независимых в статистическом смысле от реализаций, полученных до остановки.

В связи с вышесказанным, требуется разработка и имитационное моделирование системы управления заданиями, позволяющей в режиме реального времени:

- осуществлять мониторинг свободных ресурсов и подключать их к расчетам;
- назначать выделенные ядра с целью получения и осреднения данных с загруженных ядер, производящих независимое статистическое моделирование;
- осуществлять передачу начальных данных на загружаемые вычислительные ядра и выдавать задания на начало счета;

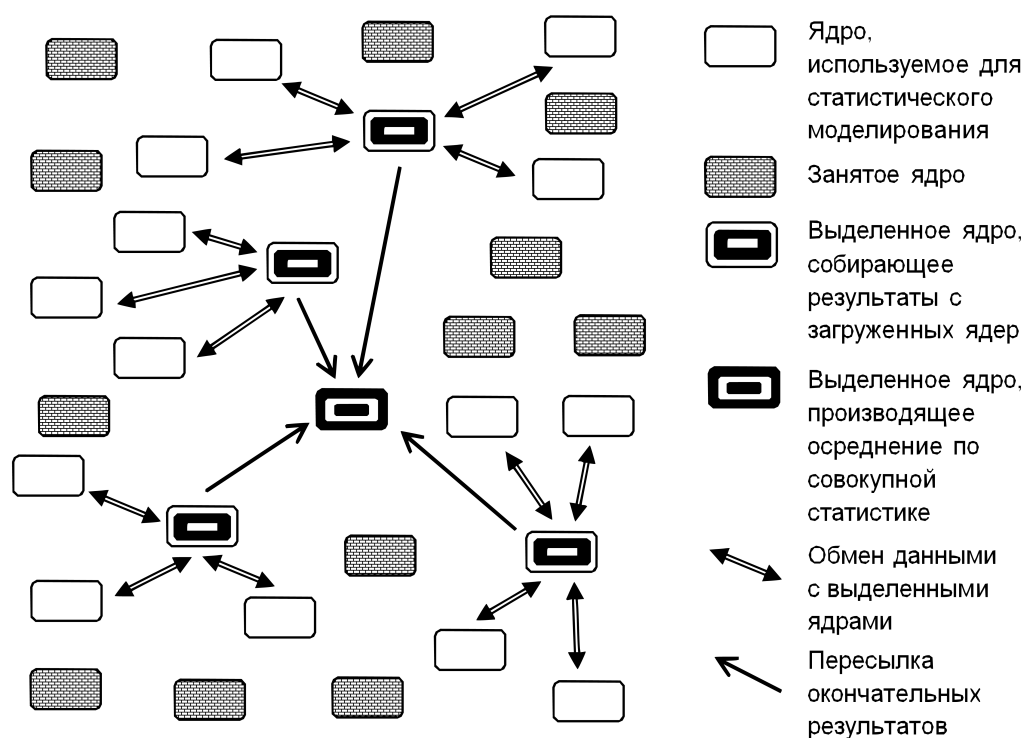


Рис. 2. Схема, поясняющая передачу нагрузки на свободные вычислительные ядра и обмен данными с выделенными ядрами при распределенном статистическом моделировании

- планировать и контролировать передачу данных в иерархической структуре выделенных ядер с целью получения окончательного результата осреднения.

Моделирование загрузки экзафлопной супер-ЭВМ можно осуществлять на основе отображения на ее архитектуру существующих параллельных алгоритмов, использующих распределенное статистическое моделирование. Параметры загрузки системы (объем данных, загрузка процессоров, использование коммуникационной сети) целесообразно оценивать на основе реальных расчетов. С этой целью может быть использована статистика о загрузке процессоров и коммуникационной сети при решении различных задач на кластерах ЦКП ССКЦ ИВМиМГ СО РАН, в частности, при использовании библиотеки *PARMONC*, предназначенной для распределенного статистического моделирования приложений метода Монте-Карло, обладающих большой вычислительной трудоемкостью [10].

Программная библиотека *PARMONC* разработана и внедрена в широкое использование с целью унификации реализации распределенного статистического моделирования для решения широкого круга задач. С ее помощью возможно масштабировать статистическое моделирование на практически неограниченное число ядер.

«Ядром» библиотеки является генератор (1). Библиотечные подпрограммы могут быть использованы в пользовательских программах, написанных на языках *C*, *C++* и *Fortran*, причем от пользователя не требуется знание языка *MPI*. В процессе счета происходит автоматическое получение выборочных средних и границ погрешностей для статистических оценок, алгоритм моделирования которых задается в пользовательской подпрограмме. Имя такой подпрограммы передается в качестве аргумента в соответствующую библиотечную подпрограмму. В процессе счета результаты вычислений периодически сохраняются на жестком диске в удобном для дальнейшей обработки виде. Библиотечные подпрограммы авто-

матически распределяют вычислительную нагрузку по вычислительным ядрам кластера. С помощью библиотеки *PARMONC* можно легко организовать продолжение ранее проведенных расчетов с автоматическим учетом их результатов. Также с помощью библиотеки можно получать коррелированные статистические оценки различных функционалов [9]. Возможность применения библиотеки *PARMONC* для распараллеливания широкого круга задач определяется естественной модульностью программ статистического моделирования.

При решении масштабных задач существенное влияние могут оказать возможные отказы вычислительных узлов. Следующий раздел посвящен подходам к моделированию таких ситуаций.

Динамическая система предотвращения сбоев на основе мультиагентного моделирования

Рассмотрим вариант реализации мультиагентного моделирования для предотвращения сбоев и отказов вычислительных узлов. Динамическая система предотвращения сбоев состоит из агентов различного назначения: агента датчика вычислительного узла; агента датчика имитационной модели; агента анализа; агента распределения; агента загрузки/отключения; экспертной системы, содержащей свод правил поведения системы для предотвращения сбоев вычислительных узлов.

Каждый агент выполняет свою функцию для достижения цели, а совместно реализуют динамическую систему предотвращения сбоев и отказов вычислительных узлов. Архитектура предлагаемой системы представлена на рис. 3.

Рассмотрим назначение основных узлов системы и принцип ее работы относительно N -го вычислительного узла.

Агент-датчик вычислительного узла (ВУ) осуществляет сбор информации о температурном режиме, нагрузке и состоянии линий связи между узлами и другой информации, характеризующей состояние узла.

Агент-датчик имитационной модели (ИМ) осуществляет мониторинг объектов имитационной модели во время имитационного прогона, моделируя температурный тренд узла, вычислительную нагрузку на узел, интенсивность обмена между узлами и другие параметры. В процессе работы он может учитывать аналогичные данные по другим вычислительным узлам.

Агент-анализа взаимодействует с агентами – датчиками вычислительного узла и имитационной модели и принимает решение об исправности данного вычислительного узла, возможности отключения или перераспределения нагрузки этого узла, на другие вычислительные узлы, руководствуясь правилами из экспертной системы. В его функцию также входит выдача информации для агента-распределителя о возможности включения данного узла.

Агент-распределения получает информацию от датчика анализа, и если принято решение о снижении вычислительной нагрузки на данный узел или его отключении, то он выясняет на каком ближайшем узле нагрузка минимальна и какую часть нагрузки можно передать на другой узел. Этот агент взаимодействует с другими агентами распределения имитационной модели и также действует по правилам экспертной системы. В функцию данного агента также входит возможность включения вычислительного узла при нормализации контролируемых параметров и извещение ближайших агентов-распределителей о готовности вычислительного узла принять дополнительную нагрузку.

Агент-загрузки/отключения осуществляет перенос нагрузки на выбранный узел. В случае отключения узла информация об этом сохраняется в агенте-распределения и она учитывается другими агентами-распределения.

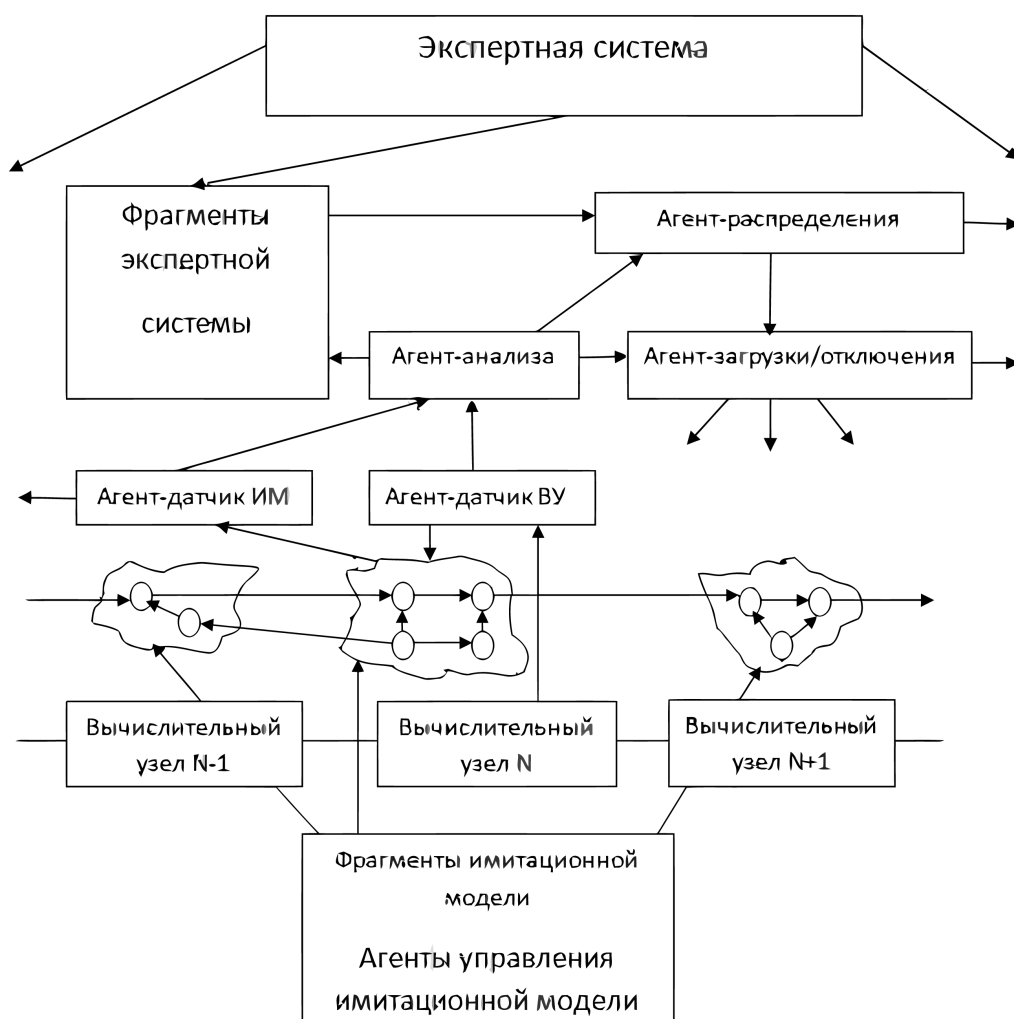


Рис. 3. Агентно-ориентированная система предотвращения сбоев и отказов вычислительных узлов

Вычислительный узел экзафлопсного компьютера вероятнее всего будет гибридным, состоящим из многоядерного *CPU*-процессора и *GPU*-процессора, либо другого ускорителя. Описание модели функционирования этого узла представляет самостоятельную задачу и в данной работе затрагиваться не будет.

Одной из ключевых проблем при реализации данной системы является разработка правил, которые будут заложены в экспертной системе для оценки текущей ситуации в каждом вычислительном узле. Свод правил должен, например, учитывать температурный тренд вычислительного узла, соответственно задавать температурный диапазон, при котором возможны отказы, и выдавать рекомендацию на разгрузку этого узла, либо отключение при достижении критического уровня. Аналогичные правила должны быть заложены в экспертной системе и для других параметров вычислительного узла, регистрируемых агентом-датчиком ВУ.

Итак, система должна функционировать следующим образом: для каждой поступающей из очереди задачи инициируется система имитационного моделирования (при этом часть ресурса вычислительной системы будет отдана для моделирования); запускается задача и одновременно имитационная модель, функционирование которой было описано выше.

В качестве примера применения в данной системе моделирования агента-датчика рассмотрим возможности системы мониторинга *IDC (Input Data Center)* [11]. Данный модуль выполняет функцию сбора информации со всего кластера, построенного на серверах *HP ProLiant BL460c*. В частности, контролируются следующие параметры:

- **температура**, с помощью датчиков *APC*, расположенных внутри шкафа и сенсоров в лезвиях, доступ к которым осуществляется с использованием интерфейса «*iLO*» (*Integrated Lights-Out*). Отметим, что полученная информация позволяет контролировать данный параметр внутри каждого узла.
- **загрузка узла**, данные, доступные в логах *CMU (Cluster Management Utility)* и характеризуют загрузку процессоров и оперативной памяти узлов. А также, что немаловажно для агент-анализа, общее время работы системы после последней операции отключения или перезагрузки.

Таким образом, реализация подобной системы предотвращения сбоев хотя и потребует определенного дополнительного ресурса, однако существенно повысит надежность функционирования экзафлопсного компьютера.

Заключение

В работе предложена концепция организации агентно-ориентированной системы имитационного моделирования экзафлопсных суперЭВМ. Агентно-ориентированный подход к моделированию систем экзафлопсной производительности был выбран вследствие того, что для имитации поведения распределённых систем наилучшим образом подходит именно распределённое моделирование, основанное на передаче сообщений.

В работе рассматривается методология моделирования загрузки экзафлопсной суперЭВМ на основе отображения на ее архитектуру задач распределенного статистического моделирования. Сформулированы базовые принципы построения соответствующей системы управления заданиями.

В работе рассмотрены принципы создания системы мультиагентного моделирования для предотвращения сбоев и отказов вычислительных узлов. Такая система состоит из агентов различного назначения и экспертной системы, содержащей свод правил поведения вычислительной системы.

В дальнейшем рассматриваемый агентно-ориентированный подход предлагается применять при имитации загрузки экзафлопсной суперЭВМ на основе использования других перспективных параллельных численных методов.

Работа проводилась в рамках реализации федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2013 годы», государственный контракт 07.514.11.4016, а также при финансовой поддержке грантов РФФИ №№ 10-07-00454, 12-01-00034, 12-01-00727, 12-07-00499; МИП №39 СО РАН, МИП №47 СО РАН, МИП №130 СО РАН; гранта НШ-2175.2012.9 совета при Президенте РФ.

Статья рекомендована к публикации программным комитетом международной суперкомпьютерной конференции «Научный сервис в сети Интернет: экзафлопсное будущее».

Литература

1. O'Keefe, R. Simulation and expert systems – a taxonomy and some examples / R. O'Keefe // Simulation. – 1986. – V. 46, №1. – P. 10 – 15.
2. Родионов, А.С. Интеллектуальное моделирование – новое направление в системах имитации (обзор последних публикаций) / А.С. Родионов // Экспертные системы и базы данных. – Новосибирск, 1988. – С. 19 – 35.
3. Wooldridge, M. Introduction to MultiAgent Systems // England: JOHN WILEY & SONS, LTD, 2002.
4. Oren, T. On the Synergy of Simulation and Agents: An Innovation Paradigm Perspective / T. Oren, L. Yilmaz // International J. of Intelligent Control and Systems. – 2009. – V. 14, №1. – P. 4 – 19.
5. Bargodia, R.L. A Message-based approach to discrete-event simulation / R.L. Bargodia, K.M. Chandy, J. Misra // IEEE Trans. on Soft. Eng. – 1987. – V. 13, №6. – P. 654 – 665.
6. Карпов, Ю.Г. Имитационное моделирование систем. Введение в моделирование на AnyLogic 5 / Ю.Г. Карпов. – СПб.: БХВ-Петербург, 2005.
7. Niewiadomska-Szynkiewicz, E. Algorithms for Distributed Simulation – Comparative Study / E. Niewiadomska-Szynkiewicz, A. Sikora // PARELEC 2002. – Warsaw, Poland. – P. 261 – 266.
8. Марченко, М.А. Распределенные вычисления по методу Монте-Карло / М.А. Марченко, Г.А. Михайлов // Автоматика и телемеханика. – 2007. – Вып. 5. – С. 157 – 170.
9. Михайлов, Г.А. Численное статистическое моделирование. Методы Монте-Карло / Г.А. Михайлов, А.В. Войтишек. – М.: Академия, 2006.
10. Marchenko, M. PARMONC - A Software Library for Massively Parallel Stochastic Simulation / M. Marchenko // ПАСТ 2011, LNCS. – 2011. – V. 6873. – P. 302 – 315.
11. Гордиенко, Д.В. Разработка и моделирование системы управления энергопотреблением кластерных вычислительных систем / Д.В. Гордиенко // Труды межд. конф. «Высокопроизводительные параллельные вычисления на кластерных системах» (ВПВКС 2009). – Владимир, 2009. – С. 126 – 130.

Борис Михайлович Глинский, доктор технических наук, профессор, ИВМиМГ СО РАН, (г. Новосибирск, Российская Федерация), gbm@org.sscs.ru.

Алексей Сергеевич Родионов, доктор технических наук, ИВМиМГ СО РАН, (г. Новосибирск, Российская Федерация), alrod@rav.sscs.ru.

Михаил Александрович Марченко, кандидат физико-математических наук, ИВМиМГ СО РАН, (г. Новосибирск, Российская Федерация), mam@osmf.sscs.ru.

Дмитрий Игоревич Подкорытов, ИВМиМГ СО РАН, (г. Новосибирск, Российская Федерация), d.podkorytov@gmail.com.

Дмитрий Владимирович Винс, ИВМиМГ СО РАН, (г. Новосибирск, Российская Федерация), wns.dmitry@gmail.com.

MSC 68M01, 68M14, 68M15

Agent-Oriented Approach to Simulate Exaflop Supercomputer with Application to Distributed Stochastic Simulation

B.M. Glinsky, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (Novosibirsk, Russian Federation),

A.S. Rodionov, Institute of Computational Mathematics and Mathematical Geophysics
SB RAS (Novosibirsk, Russian Federation),

M.A. Marchenko, Institute of Computational Mathematics and Mathematical Geophysics
SB RAS (Novosibirsk, Russian Federation),

D.I. Podkorytov, Institute of Computational Mathematics and Mathematical Geophysics
SB RAS (Novosibirsk, Russian Federation),

D.V. Weins, Institute of Computational Mathematics and Mathematical Geophysics
SB RAS (Novosibirsk, Russian Federation)

A possibility of using an agent-oriented simulation system for solving a variety of problems that arise in design and implementation of exaflop supercomputers consisting of ten and hundred millions of computational nodes is discussed in the paper. We suggest two-level decentralized scheme of computations control and the corresponding simulation model in which all the computational nodes are distributed over computational domains controlled by their control agents. Master control agent distributes a flow of big problems over computational domains and manages common resources. Monte-Carlo method considered to be promising to use on exaflop supercomputers is given as an example of highly scalable algorithm. In this method, is essential that the larger sample size of independent realizations, the higher accuracy of estimating. We also suggest a parallel pseudorandom numbers generator suitable for large-scale computations with Monte Carlo method. When distributing stochastic computations over different nodes it is possible to simulate different sample volumes on different nodes using statistically optimal technique of results averaging. Naturally, an amount of computer resources available on each node must be quite enough to simulate the realizations effectively. The described algorithm of distributed stochastic simulation is asynchronous one and can be scaled to a practically infinite number of nodes using the described parallel pseudorandom numbers generator. An example of the highly scalable application utilizing distributed stochastic simulation on up-to-date teraflop supercomputers is the program library PARMONC. Also, the multi-agent simulation is used for the prediction and processing of possible failures of computational nodes. Architecture of dynamic system of failures prediction is given. The system consists of the agent for different purposes; each agent is playing its role to achieve the common goal.

Keywords: agent-oriented simulation, exaflop supercomputer, Monte Carlo method, distributed stochastic simulation, parallel computations.

References

1. O’Keefe R. Simulation and Expert Systems – a Taxonomy and Some Examples. *Simulation*, 1986, vol. 46, no. 1, pp. 10 – 15.
2. Rodionov A.S. Intellectual Simulation – New Approach in Imitation Systems (Review of Recent Publications). *Expert Systems and Data Bases*, Novosibirsk, 1988, pp. 19 – 35.
3. Wooldridge M. *Introduction to MultiAgent Systems*. England: JOHN WILEY & SONS, LTD, 2002.
4. Oren T., Yilmaz L. On the Synergy of Simulation and Agents: An Innovation Paradigm Perspective. *International J. of Intelligent Control and Systems*, 2009, vol. 14, no. 1, pp. 4 – 19.
5. Bargodia R.L., Chandy K.M., Misra J. A Message-Based Approach to Discrete-Event Simulation. *IEEE Trans. on Soft. Eng.*, 1987, vol. 13, no. 6, pp. 654 – 665.
6. Karpov Yu.G. *Imitation Modeling of Systems. Introduction to Simulation with Any-Logic 5*. SPB., BHV-Petersburg, 2005.
7. Niewiadomska-Szynkiewicz E., Sikora A. Algorithms for Distributed Simulation – Comparative Study. *PARELEC 2002*, Warsaw, Poland, pp. 261 – 266.

8. Marchenko M.A., Mikhailov G.A. Distributed Computing by the Monte Carlo Method. *Automation and Remote Control*, 2007, no. 68(5), pp. 888 – 900.
9. Mikhailov G.A., Voytishchik A.V. *Numerical Stochastic Simulation. Monte Carlo Method*. Publishing Center «Akademia», 2006.
10. Marchenko M. PARMONC – A Software Library for Massively Parallel Stochastic Simulation *PACT 2011, LNCS*. 2011, vol. 6873, pp. 302 – 315.
11. Gordienko D.V. Developing and Modeling the System of Controlling the Power Consumption of Cluster Computing Systems *Proceedings of International Conference «High performance computations on cluster systems»*. Vladimir, 2009, pp. 126 – 130.

Поступила в редакцию 8 декабря 2011 г.