

3-бөлім

Раздел 3

Section 3

Информатика

Информатика

Computer
science

УДК 519.683.5

Мәткерім Б. *, Ахмед-Заки Д.Ж.**, Мансурова М.Е.***

Казахский национальный университет имени аль-Фараби, Республика Казахстан, г. Алматы
E-mail: *bazargulmm@gmail.com, **darhan.ahmed-zaki@kaznu.kz, ***mansurova01@mail.ru

Проектирование и разработка приложений для высокопроизводительных научных вычислений с применением методологии MDD

В данной работе представлены результаты исследований по применению методологии MDD при разработке высокопроизводительных научных HPSC приложений (HPSC - High Performance Scientific Computing). В качестве одного из стандартов MDD выбрана концепция MDA. Процесс разработки HPSC приложений описан с помощью моделей MDA: вычислительно-независимой (CIM), платформенно-независимой (PIM) и платформенно-зависимой (PSM) моделей. Построенная машиночитаемая (machine readable) модель PSM позволяет производить автоматическую генерацию программного кода. Для создания сложных научных приложений на основе технологии MDA для HPSC приложений предложен подход с передачей эстафеты, который распределяет работу между специалистами из различных областей. В качестве конкретного примера на основе разработанной модели MDA реализовано приложение для решения задачи движения жидкости в нефтяном пласте с применением языка программирования Java и стандарта параллельного программирования MPI.
Ключевые слова: методология MDD, концепция MDA, высокопроизводительные научные вычисления.

Matkerim B., Akhmed-Zaki D.Zh., Mansurova M.E.

Design and development high performance scientific computing applications using MDD methodology

Paper presents the research result of using Model-Driven development (MDD) methodology to model and develop high performance scientific computing (HPSC) applications. We choose MDA technology as one of the MDD standards, and specify HPSC application development process with MDA models: computation independent model (CIM), platform independent model (PIM) and platform specific model (PSM). The refined machine-readable PSM model could be automatically generated to programming code. In order to developing complex HPSC application presents to organize relay race of specialties based on MDA technology, which is allowed to distribute work between specialties from different research areas. As a case study, we implement HPSC application based on MDA models for the problem of fluid flow in oil reservoir, which is used programming language Java and parallel programming standard MPI.

Key words: MDD methodology, MDA technology, high-performance computing.

Мәткерім Б., Ахмед-Заки Д.Ж., Мансурова М.Е.

Жоғары өнімді ғылыми есептеу қосымшаларын MDD методологиясын пайдаланып жобалау және өңдеу

Бұл зерттеу жұмысында Модельдермен Меңгерілетін өңдеу методологиясы - MDD-ді жоғары өнімді ғылыми есептеу (ЖӨҒЕ) қосымшаларын жобалау және өңдеуде пайдалану нәтижелерді ұсынылды. MDD методологиясының стандарттарының бірі ретінде MDA технологиясын таңдалған. ЖӨҒЕ қосымшаларын өңдеу процесі MDA модельдерімен сипатталған, олар есептеуден тәуелсіз (CIM), платформадан тәуелсіз (PIM) және платформаға тәуелді модельдер (PSM).

Құрылған машинамен оқылатын (machine readable) PSM моделі автоматты түрде бағдарламалық кодты генерациялауға мүмкіндік береді. MDD методологиясын пайдаланып күрделі ЖӨҒЕ қосымшаларын құру үшін әр түрлі саланың мамандары арасында жұмысты үлестіруге мүмкіндік беретін эстафетаны ұйымдастыру ұсынылды. Нақты мысал ретінде өңделген MDA модельдерінің негізінде мұнай пластындағы сұйықтықтың ағысын зерттейтін үш өлшемді есепті шешуге арналған қосымшаны Java бағдарламалау тілінде және параллельді бағдарламалау стандарты MPI-ды пайдаланып жүзеге асырылды.

Түйін сөздер: MDD методологиясы, MDA технологиясы, жоғары өнімді ғылыми есептеу.

Введение

Традиционно процесс разработки приложений для решения сложных научных задач состоит из нескольких этапов:

- исследование особенностей физических процессов;
- математическое моделирование физических процессов;
- численное моделирование процессов;
- разработка программного обеспечения;
- валидация и верификация программного обеспечения;

При разработке научных приложений сложные физические процессы описываются с помощью математических моделей, которые затем уточняются и верифицируются. Построение таких моделей осуществляется с помощью сложных математических алгоритмов, которые требуют больших вычислительных ресурсов. В силу этого решение масштабных расчетных научных задач возможно только с применением высокопроизводительных технологий. Повышение эффективности решаемых задач напрямую зависит от мощности высокопроизводительных вычислительных систем. На различных этапах разработки научных приложений возникают сложности, связанные с неоднозначностью интерпретации моделей, зависимостью от платформы и технологий параллельного программирования, низким уровнем абстракции и др. [1]. Это приводит к тому, что реализация приложения происходит на низком программном уровне с учетом особенностей архитектуры аппаратного обеспечения. Вследствие этого выполнение таких действий как устранение ошибок программного кода, перенос программного кода (portability) на новую технологию или на новую версию существующей технологии, документирование, модификация программного обеспечения, поддержка программного обеспечения, применение существующего приложения для решения новых задач вызывает значительные затруднения. В данной работе для преодоления вышеуказанных сложностей при проектировании и разработке научных приложений применяется методология MDD (Model-Driven Development - разработка, управляемая моделями), а именно ее стандарт MDA (Model-Driven Architecture - архитектура, управляемая моделью). Вопросам построения приложений на основе MDA посвящено большое количество работ [2-4]. Особенностью данной работы является применение технологии MDA к разработке приложений для высокопроизводительных научных вычислений. Процесс проектирования HPSC приложений описывается с помощью разработанных авторами специальных компонентов: InOutPut, SciEquations, NumerMethods и PEOrganize, которые отражают специфику научных вычислительных задач [2]. Статья организована следующим образом. В разделе 1 представлена методология MDD и концепция стандарта MDA. В разделе 2 описан процесс проектирования HPSC приложений с применением технологии MDA, а также подход с передачей эстафеты между специалистами из различных

отраслей. В разделе 3 представлен конкретный пример решения задачи движения жидкости в нефтяном пласте на основе построенной модели MDA. В разделе заключения подведены итоги выполненной работы.

1 Применение методологии MDD и стандарта MDA для проектирования приложений

MDD (Model-Driven Development) - методология, согласно которой при создании программного обеспечения основными объектами разработки становятся модели. Из моделей в дальнейшем генерируется код и другие объекты. Здесь модель означает формальную спецификацию функций, структуры и действий системы [5]. По методологии MDD все этапы проектирования и разработки приложения основываются на моделях, поэтому данная методология называется "управляемая моделями" (model-driven). В настоящее время методология MDD в области информационных технологий представляет собой еще один уровень абстрагирования в объектно-ориентированном подходе при разработке программного обеспечения. С повышением уровня абстрагирования облегчается процесс проектирования и разработки приложений. В начале 2000 года консорциум OMG [6] выдвинул в качестве инициативы концептуальную инфраструктуру MDA и ее основные принципы для методологии MDD, а также стандартные языки спецификаций для описания и трансформации моделей. В руководстве MDA [5] указано, что "MDA представляет собой подход к разработке систем, который увеличивает возможности моделей в этой работе". Этот подход является управляемым моделями, потому что он обеспечивает средства для использования моделей на этапах представления, проектирования, создания, развертывания, эксплуатации, технического обслуживания и модификации систем. Основные концепции MDA моделирования можно описать следующим образом:

CIM (Computation Independent Model) - Вычислительно-независимая модель:

- Представляет собой бизнес модель или модель предметной области.
- Определяет, что должна выполнять система.
- Скрывает информацию о том, как реализуется система.
- Две последующие модели должны следовать требованиям CIM.

PIM (Platform Independent Model) - Платформенно-независимая модель:

- Определяет сервисы системы посредством абстрагирования.
- Не содержит сведений о структуре конкретной платформы.

PSM (Platform Specific Model) - Платформенно-зависимая модель:

- Объединяет спецификацию PIM и конкретную платформу.

На рисунке 1 показан цикл разработки MDA приложений, который состоит из трех этапов: анализа, проектирования и написания программного кода. При этом соответственно на первом этапе основное внимание уделяется разработке модели PIM, на втором этапе - модели PSM, и на третьем этапе - написанию программного кода. Под разработкой моделей подразумевается не только их понимание, описание, но также и представление в виде, в котором их можно обрабатывать на компьютере, а также предоставление возможности трансформации модели PIM в модель PSM, генерации из модели PSM программного кода. На рисунке рядом со стрелками, соединяющими элементы, показаны номера, означающие последовательные шаги цикла разработки систем. Начальный прототип модели может содержать неточности или несоответствия, поэтому в результате проведения валидации и внесения изменений создаются последующие прото-

типы. Только когда созданная модель пройдет проверку и будет соответствовать предъявляемым к ней требованиям, данная модель может трансформироваться в следующую модель. В тех случаях, когда в функционирующую систему нужно внести изменения, например, запустить ее с помощью другой технологии, нужно пересмотреть модели PIM, PSM (шаги 4, 7 на рис. 1). Затем создать с учетом изменений новые модели PSM (например, шаги 2, 5) и выполнить генерацию кода.

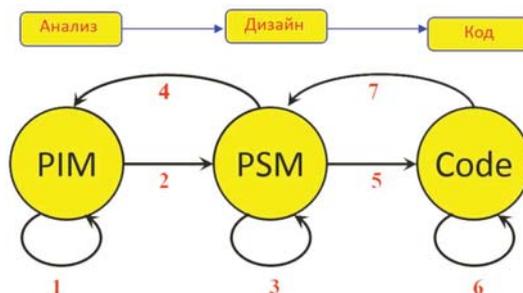


Рисунок 1 - Цикл разработки приложений MDA [6]

Для создания моделей MDA может быть использован унифицированный язык моделирования UML (Unified Modeling Language) [8, 9]. В этом случае для разработки MDA приложений применяются следующие диаграммы языка UML: диаграммы прецедентов, диаграммы классов и объектов, диаграммы состояния, диаграммы взаимодействия. Технология MDA находит широкое применение при разработке приложений в различных сферах: финансовой, коммерческой, государственной, образовательной [10]. Помимо этого MDA применяется для разработки приложений в научной сфере.

2 Проектирование и разработка приложений для высокопроизводительных вычислений

В области высокопроизводительных научных вычислений модель CIM представляет собой математическую модель, которая является математическим представлением приложения с расчетно-независимой точки зрения. Модель PIM преобразовывается из модели CIM, и она не зависит от следующих факторов: от конкретного поколения (третьего или четвертого) языков программирования; распределенного промежуточного компонента. Модель PIM HPSC-приложения является технологически нейтральной. Она определяет работу системы, скрывая детали, необходимые для конкретной платформы. Модель PIM может быть преобразована в одну или более моделей PSM в зависимости от конкретных требований HPSC-приложения. В современной практике HPSC приложения реализованы, в основном, с помощью технологий CUDA, MPI, OpenMP MapReduce. Также на основе данных технологий создаются гибридные подходы в зависимости от HPSC аппаратных ресурсов, и языка программирования Java, C/C++ или Fortran. На последнем этапе PSM автоматически трансформируется в соответствующий Java или C/C++ код конкретной технологии. На рисунке 2 представлена MDA модель для HPSC приложений.

На рисунке 3 процесс проектирования и разработки приложений для высокопроизводительных научных вычислений (HPSC - High Performance Scientific Computing) пред-

ставлен с помощью диаграммы деятельности языка UML. Согласно рисунку 3 процесс разработки HPSC приложения с помощью технологии MDA делится на 5 этапов.

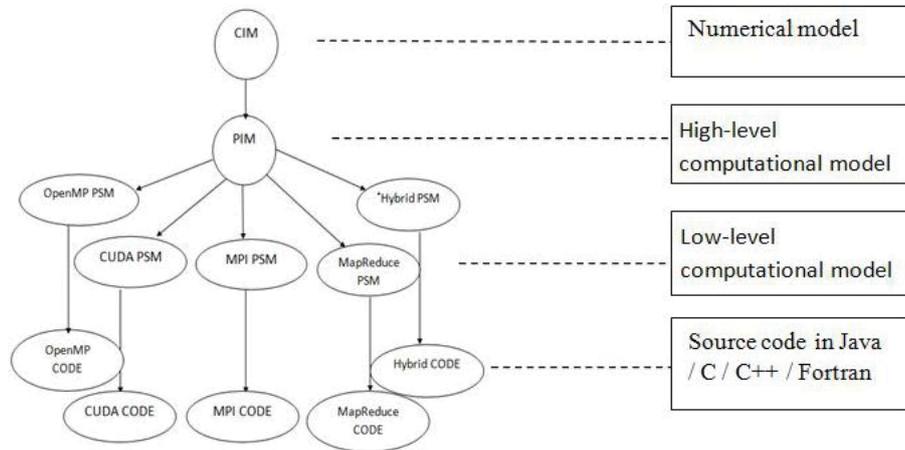


Рисунок 2 - MDA модель для HPSC приложений

Специалист, отвечающий за соответствующий этап, указан с левой стороны рисунка. С правой стороны показана модель MDA, полученная в результате выполнения работы на соответствующем этапе. Такое распределение работы при разработке приложений для высокопроизводительных научных вычислений привело к идее организации "Эстафеты специалистов" (рис. 4).

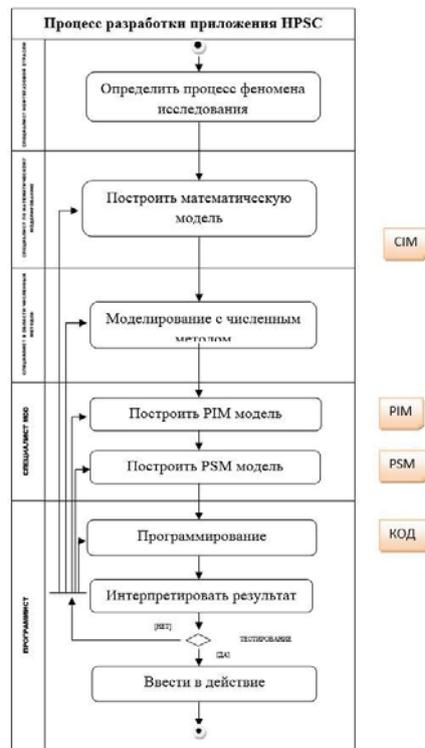


Рисунок 3 - Процесс разработки программного обеспечения при помощи MDA

Как показано на рисунке 4, каждый специалист вносит свой вклад в разработку приложения в соответствии со спецификой своей профессиональной деятельности. Специалист в области научных исследований определяет предмет исследований и отвечает за постановку задачи. Специалист, следующим принимающий эстафету, является специалистом по математическому моделированию. Именно он создает математическую модель задачи. Далее эстафета переходит к специалисту в области численных методов. Этот специалист подбирает наиболее подходящий численный метод для решения представленной математической модели. Эстафета передается специалисту по MDD. Он в свою очередь, согласно циклу разработки приложений (рис. 1) и диаграмме деятельности UML (рис. 3), создает PIM модель, далее производит трансформацию этой модели в PSM модель в соответствии с выбранной платформой и технологиями.



Рисунок 4 - Эстафета специалистов, разработчиков научных приложений

Из PSM модели генерируется программный код, который передается на доработку программисту, специалисту в области параллельных вычислений. Он дополняет сгенерированный код и завершает программную реализацию приложения. Далее, согласно последовательности процесса разработки приложения (рис. 3), программист проводит тестирование. В случае неудовлетворительного результата причины возникшей проблемы анализируются соответствующими специалистами, которые затем вносят свои изменения. Приложение снова тестируется. Этот цикл продолжается до тех пор, пока приложение не пройдет успешного тестирования. На последнем этапе готовое приложение предоставляется заказчику.

3 Пример проектирования и разработки научного приложения для расчета давления жидкости в нефтяном пласте

3.1 Постановка задачи

В качестве конкретного примера проектирования и разработки научного приложения в соответствии с описанной выше методологией рассматривается задача нефтегазовой отрасли расчета давления жидкости в нефтяном пласте. Согласно подходу с передачей эстафеты (рис. 4) процесс разработки приложения начинается с работы специалиста нефтегазовой отрасли, объектом исследования для которого является давление жидкости в нефтяном пласте. Далее эстафета передается специалисту по математическому моделированию, который формулирует задачу в виде трехмерных дифференциальных уравнений в частных производных, представленных ниже.

Рассматривается гиперкуб в пористой эластичной анизотропной среде $\Omega = [0, T] \times K\{0 \leq x \leq 1, 0 \leq y \leq 1, 0 \leq z \leq 1\}$. Пусть уравнение (1) описывает давление жидкости в гиперкубе с начальными условиями (2) и граничными условиями (3):

$$\frac{\partial P}{\partial t} = \frac{\partial}{\partial x}(\phi(x, y, z) \frac{\partial P}{\partial x}) + \frac{\partial}{\partial y}(\phi(x, y, z) \frac{\partial P}{\partial y}) + \frac{\partial}{\partial z}(\phi(x, y, z) \frac{\partial P}{\partial z}). \quad (1)$$

$$P(0, x, y, z) = \varphi(0, x, y, z). \quad (2)$$

$$\frac{\partial P}{\partial n} \Big|_{\Gamma} = 0. \quad (3)$$

Здесь в условии (3) является площадью поверхности куба Ω . В уравнении (1) функция решения $P(t, x, y, z)$ является пластовым давлением в точке (x, y, z) в момент времени t ; $\phi(x, y, z)$ является коэффициентом диффузии в пласте; $f(t, x, y, z)$ - плотностью источников. Для решения задачи (1) - (3) специалист по численным методам использует метод конечных разностей по явной схеме. Записав разностный аналог задачи (1) - (3), он передает его специалисту по MDD.

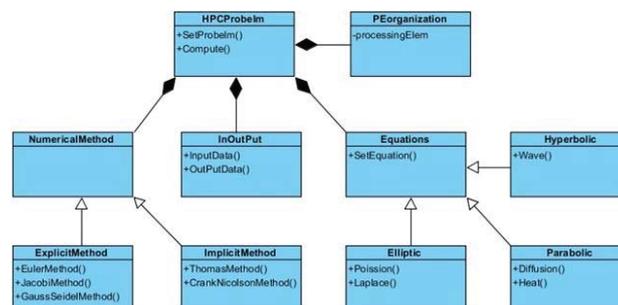


Рисунок 5 - PIM модель высокопроизводительного приложения

Специалист по MDD создает PIM модель (рис. 5) с помощью компонентов для проектирования HPSC приложений [2]. Согласно [2] любое приложение для высокопроизводительных вычислений может быть представлено с помощью четырех компонентов: PEOrganize, NumerMethods, SciEquations, InOutPut. Каждый компонент состоит из нескольких классов, которые поддерживают все свойства классов согласно парадигме объектно-ориентированного программирования. Так как для разработки приложения используется язык программирования Java и стандарт параллельного программирования MPI, далее производится трансформация PIM модели в модель MPI Java PSM (рис. 6).

Модели преобразования UML могут быть классифицированы по нескольким категориям [11]: с улучшением качества преобразования, с усовершенствованием разработки, с уточнением, со специализацией, с переводом, с абстракцией, обобщением и с шаблонами проектирования. В нашем случае переход от модели PIM к модели MPI Java PSM относится к категории с уточнением. Refining означает уточнение при переходе из CIM

в PIM и переходе из PIM в PSM. Уточнения могут быть добавлены на одном уровне абстракции. Трансформация модели PIM в модель MPI Java PSM (рис. 6) заключается в преобразовании UML-диаграммы классов в диаграмму классов Java с добавлением MPI спецификации к PSM. В преобразовании модели PIM в MPI Java PSM должно быть удалено множественное наследование, ассоциации классов и квалифицированные ассоциации.

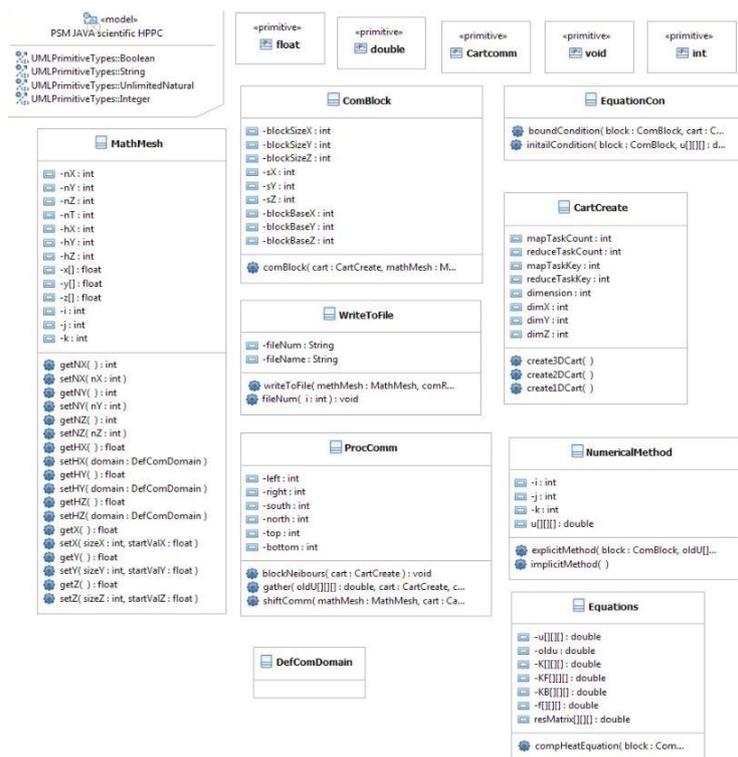


Рисунок 6 - PSM модель высокопроизводительного приложения

В соответствии с особенностями стандарта MPI для модели MPI PSM производится спецификация компонента "PEorganization" а именно определяются операции для классов, которые создают 3D виртуальную декартову топологию для процессов, участвующих в вычислениях, осуществляют 3D декомпозицию данных трехмерной задачи. Классы взаимодействуют друг с другом посредством параметров операций. Далее производится автоматическая генерация кода из модели MPI PSM. Автоматический переход от MPI PSM в Java код реализуется с помощью генератора Asceleo [12]. Asceleo является прагматической реализацией Object Management Group (OMG) модели MOF. Asceleo UML2 для Java является генератором кода на основе Asceleo 3.2. Этот генератор поддерживает создание исходного кода Java для классов и интерфейсов. В результате работы Asceleo автоматически генерируются java файлы, как показано на рис. 7. Описанная диаграммой классов UML модель PSM вместе с алгоритмом реализации операций классов компонентов, либо вместе с псевдокодом передается по эстафете последнему звену цепочки специалистов - программисту. Он, в свою очередь, дополняет автоматически сгенерированный код в соответствии с информацией, полученной от специалиста по MDD, проводит тестирование до тех пор, пока не получит удовлетворительные результаты.



```

1
2
3 package MapReduce PSM;
4
5 // Start of user code for imports
6 import java.util.*;
7
8 // End of user code
9
10
11
12
13 public class MathMesh {
14     /**
15      * the hX attribute.
16      */
17     private int hX;
18     /**
19      * the j attribute.
20      */
21     private int j;
22     /**
23      * the nX attribute.
24      */
25     private int nX;
26     /**
27      * the i attribute.
28      */
29     private int i;
30     /**
31      * the z[] attribute.
32      */
33     private float z[];
34     /**

```

Рисунок 7 - Фрагмент сгенерированного кода

3.2 Результаты вычислительного эксперимента

После завершения стадии разработки MPI Java приложения для задачи (1)-(3) производится запуск приложения на высокопроизводительном кластере для получения результатов вычислений. Для оценки ускорения (рис. 8) и эффективности (рис. 9) были рассмотрены несколько параллельных вариантов позволяющих, осуществлять моделирование в трехмерной области с размерами:

1. $64 \times 64 \times 64 = 262,144$;
2. $128 \times 128 \times 128 = 2,097,152$;
3. $240 \times 240 \times 240 = 13,824,000$;
4. $480 \times 480 \times 480 = 110,592,000$.

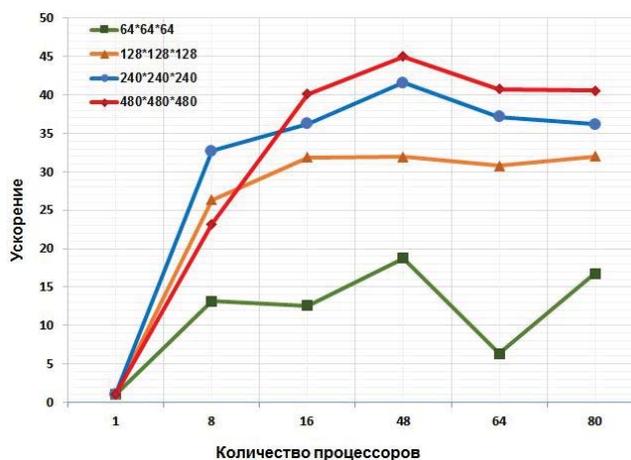


Рисунок 8 - График ускорения

Расчеты точек производились на высокопроизводительном кластере T-Cluster КазНУ им. аль-Фараби. Основу высокопроизводительного аппаратно-программного комплекса

составляет вычислительный кластер на базе 26-и вычислительных лезвий, содержащих 2 шестиядерных процессора (Intel® Xeon® CPU E5645 2.40 GHz). Общий объем оперативной памяти составляет 624 гигабайтов, общий объем HDD составляет 20-терабайт. Как можно заметить по полученным графикам, с увеличением количества процессов ускорение растет лишь до некоторого значения, что связано с увеличением накладных расходов на коммуникации. Самым оптимальным количеством процессов оказалось значение 48, при котором наблюдалось максимальное значение ускорения, но наибольшее значение эффективности было достигнуто при 8 процессорах.

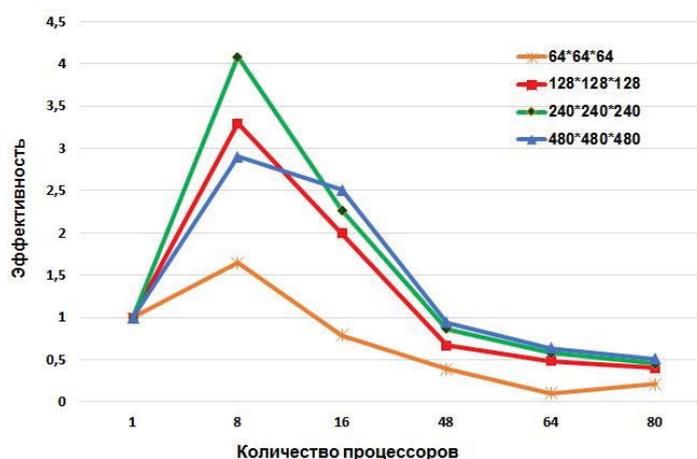


Рисунок 9 - График эффективности

Заключение

В данной работе описан подход по применению стандарта MDA методологии MDD при разработке высокопроизводительных научных HPSC приложений. На основе предложенной MDA модели реализовано приложение для решения задачи движения жидкости в нефтяном пласте.

По сравнению с традиционным подходом проектирования и разработки приложений подход с применением стандарта MDA имеет следующие преимущества:

1. При проектировании и разработке приложения фокус внимания переносится с этапа написания кода на этап моделирования, что позволяет применять модели абстракции высокого (PIM) и низкого уровней (PSM). Применение данной технологии дает возможность рассматривать спецификацию области научных исследований и техническую реализацию высокопроизводительных вычислений отдельно друг от друга. 2. С помощью разработанных компонентов проектирования можно строить различные MDA модели для научных приложений.

3. При проектировании приложений разработка модели PIM ведется без учета особенностей конкретной технологии параллельных вычислений: MPI, CUDA, MapReduce или их гибридного соединения, так как спецификация технологии отражается в модели PSM.

4. Возможность автоматической генерации программного кода из модели PSM позволяет на 50-60 % сократить время, затрачиваемое программистом на разработку, а также уменьшить количество ошибок при написании кода.

5. Технология MDA дает возможность организовать эстафету специалистов, распределение работы между которыми облегчает работу над созданием крупных научных приложений. Технология MDA позволяет специалистам из различных предметных областей вносить вклад именно по части своей специализации, что способствует улучшению качества разрабатываемого программного обеспечения.

Указанные выше преимущества MDA моделирования доказывают перспективность применения данной технологии при проектировании и разработке приложений для научных вычислений. В дальнейшем планируется продолжить исследования в данном направлении и разработать различные HPSC приложения на основе предложенной MDA модели.

Литература

- [1] *Lugato J., Bruel M., Ober I.* Model-Driven Engineering for High Performance Computing Applications // Proc. Modeling Simulation and Optimization Focus on Applications.-2010. -P. 303-308.
- [2] *Matkerim B., Akhmed-Zaki D., Barata M.* Development High Performance Scientific Computing Application Using Model-Driven Architecture // Applied Mathematical Sciences, Vol. 7, 2013, N. 100, 4961-4974.
- [3] *Daniluk.* Visual modeling for scientific software architecture design. A Practical approach // Computer Physics Communications. -2012. - No.183. - P. 213-230.
- [4] *Ober.* MDE4HPC: An Approach for Using Model-Driven Engineering in High-Performance Computing, SDL 2011, LNCS7083, pp. 247-261, 2011.
- [5] MDA Guide. Object Management Group Model Driven Architecture (MDA) MDA Guide rev. 2.0 OMG Document ormsc/2014-06-01 [online] Available at: <http://www.omg.org/cgi-bin/doc?ormsc/14-06-01>
- [6] *Bezivin Jean.* Object to Model Paradigm Change with the OMG/MDA Initiative, presentation of Summer School on MDA for Embedded System Development, CEA/ENSIETA/INSA Lyon, Brest, France, September 16-20, 2002.
- [7] OMG official website: www.omg.org.
- [8] UML official website: http://www.omg.org/gettingstarted/what_is_uml.htm.
- [9] *Scott W., Ambler.* The Elements of UML 2.0 Style, Cambridge University Press, 2005.
- [10] MDA success story. URL: http://www.omg.org/mda/products_success.htm. по состоянию на: 15.05.2013.
- [11] *Lano K.* Model-Driven software development with UML and Java, 2009.
- [12] Acceleo web site. URL: <http://www.eclipse.org/acceleo/>, по состоянию на: 15.08.2013.
- [13] *Мәткерім Б., Ахмед-Заки Д.Ж.* MDD методологиясын мұнай-газ саласы мәселелерін шешуге арналған жоғары өнімді есептеу қосымшаларын жобалау және өңдеуде пайдалану. // Сборник трудов VIII Казахстанско-Российской международной научно-практической конференции "Математическое моделирование в научно-технологических и экологических проблемах нефтегазовой отрасли Атырау, 2014, с. 477-481.
- [14] *Самарский А.А., Николаев Е.С.* Методы решения сеточных уравнений. Наука, 1978, 592 с.

References

- [1] *Lugato J., Bruel M., Ober I.* Model-Driven Engineering for High Performance Computing Applications // Proc. Modeling Simulation and Optimization Focus on Applications.-2010. -P. 303-308.
- [2] *Matkerim B., Akhmed-Zaki D., Barata M.* Development High Performance Scientific Computing Application Using Model-Driven Architecture // Applied Mathematical Sciences, Vol. 7, 2013, N. 100, 4961-4974.

-
- [3] *Daniluk*. Visual modeling for scientific software architecture design. A Practical approach // Computer Physics Communications. -2012. - No.183. - P. 213-230.
- [4] *Ober*. MDE4HPC: An Approach for Using Model-Driven Engineering in High-Performance Computing, SDL 2011, LNCS7083, pp. 247-261, 2011.
- [5] MDA Guide. Object Management Group Model Driven Architecture (MDA) MDA Guide rev. 2.0 OMG Document ormsc/2014-06-01 [online] Available at: <http://www.omg.org/cgi-bin/doc?ormsc/14-06-01>
- [6] *Bezivin Jean*. Object to Model Paradigm Change with the OMG/MDA Initiative, presentation of Summer School on MDA for Embedded System Development, CEA/ENSIETA/INSA Lyon, Brest, France, September 16-20, 2002.
- [7] OMG official website: www.omg.org.
- [8] UML official website: http://www.omg.org/gettingstarted/what_is_uml.htm.
- [9] *Scott W., Ambler*. The Elements of UML 2.0 Style, Cambridge University Press, 2005.
- [10] MDA success story. Available at: URL: http://www.omg.org/mda/products_success.htm/ 15.05.2013.
- [11] *Lano K*. Model-Driven software development with UML and Java, 2009.
- [12] Acceleo web site. Available at: URL: <http://www.eclipse.org/acceleo/> 15.08.2013.
- [13] *Мәткерім Б., Ахмед-Заки Д.Ж.* Matkerim B., Akhmed-Zaki D.Zh. MDD metodologiyasyn munay gas salasy maselelerin sheshuge arnalgan zhogary onimdi esepteu kosymshalaryn zhubalau zhane ondeude paydalanu. // Proceedings of the VIII International Kazakh-Russian scientific-practical conference "Mathematical modeling in scientific and technological, and environmental challenges of oil and gas industry Atyrau, 2014, c. 477-481.
- [14] *Samarisky A.A., Nikolayev E.S.* Methods for solving grid equations. M.: Nauka, 1978. 592 p.