

# **СХЕМА ОРГАНИЗАЦИИ СЕРВИСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, СОДЕЙСТВУЮЩЕГО УДАЛЁННЫМ ПОЛЬЗОВАТЕЛЯМ В АДАПТАЦИИ ИХ ПРОГРАММ К ВЫСОКОПРОИЗВОДИТЕЛЬНЫМ ВЫЧИСЛИТЕЛЬНЫМ РЕСУРСАМ**

**А.В. Филатов**

Одним из видов научного сервиса в интернет является предоставление пользователям высокопроизводительных вычислительных ресурсов (далее просто – ресурсов). Во-первых, такие ресурсы могут предоставляться для решения научных и производственных задач в рамках предприятия – владельца ресурсов, входящим в него подразделениям и физическим лицам. Во-вторых, владельцы всё чаще предоставляют свои ресурсы удалённым пользователям для решения их задач.

На сегодня, эффективное использование предоставляемого высокопроизводительного вычислительного ресурса подразумевает подготовку адаптированной под него параллельной программы. Под адаптированностью параллельной программы будем подразумевать её приспособленность к особенностям предоставляемых вычислительных ресурсов (к особенностям относятся параметры: архитектуры, структуры, системного программного обеспечения, алгоритмов функционирования и т.д.) с точки зрения достижения наилучшего результата по выбранному критерию. В качестве основного критерия при использовании таких ресурсов, как правило, берётся сокращение времени вычислений.

Во многих случаях, пользователи используют предоставляемые им ресурсы для выполнения программ, которые они сами создают или модернизируют. Подготовить адаптированную программу пользователь может сам, может использовать в программе компоненты адаптированного пакета, а также может обратиться к специальному сервису.

Как правило, при самостоятельной подготовке адаптированной программы пользователь встречает ряд трудностей. Первая из них может заключаться в недостаточной подготовленности пользователя в области создания эффективных параллельных программ (пользователь, как правило, является специалистом в своей области). Вторая заключается в недостаточном знакомстве пользователя с особенностями доступного вычислительного ресурса. Это наиболее актуально для пользователей удалённых ресурсов, особенно если состав их непостоянен. Третья трудность заключается в необходимости проведения значительного числа натурных экспериментов на выделенном ресурсе, что приводит к непроизводительному его использованию и отвлечению от выполнения уже готовых программ, в том числе и программ других пользователей.

Применение пользователем в своей программе компонентов адаптированных пакетов является неплохим выходом из положения, однако это требует наличия унифицированных пакетов для всех используемых пользователем ресурсов, а также может быть крайне неэффективно при решении нестандартных задач.

Выходом из сложившейся ситуации может быть использование пользователем специального сервиса, помогающего ему в создании своих программ адаптированных под конкретные вычислительные ресурсы. Такой автоматизированный сервис может быть предоставлен владельцем ресурса его пользователям.

Проследим этапы подготовки пользовательской задачи для выполнения её на выделенном ресурсе. В качестве ресурса возьмём кластерную вычислительную систему (систему, кластер). Такие системы получают всё большее распространение в мире.

Пример одного из типичных алгоритмов подготовки программы пользователем показан на схеме (см. рис. 1.).

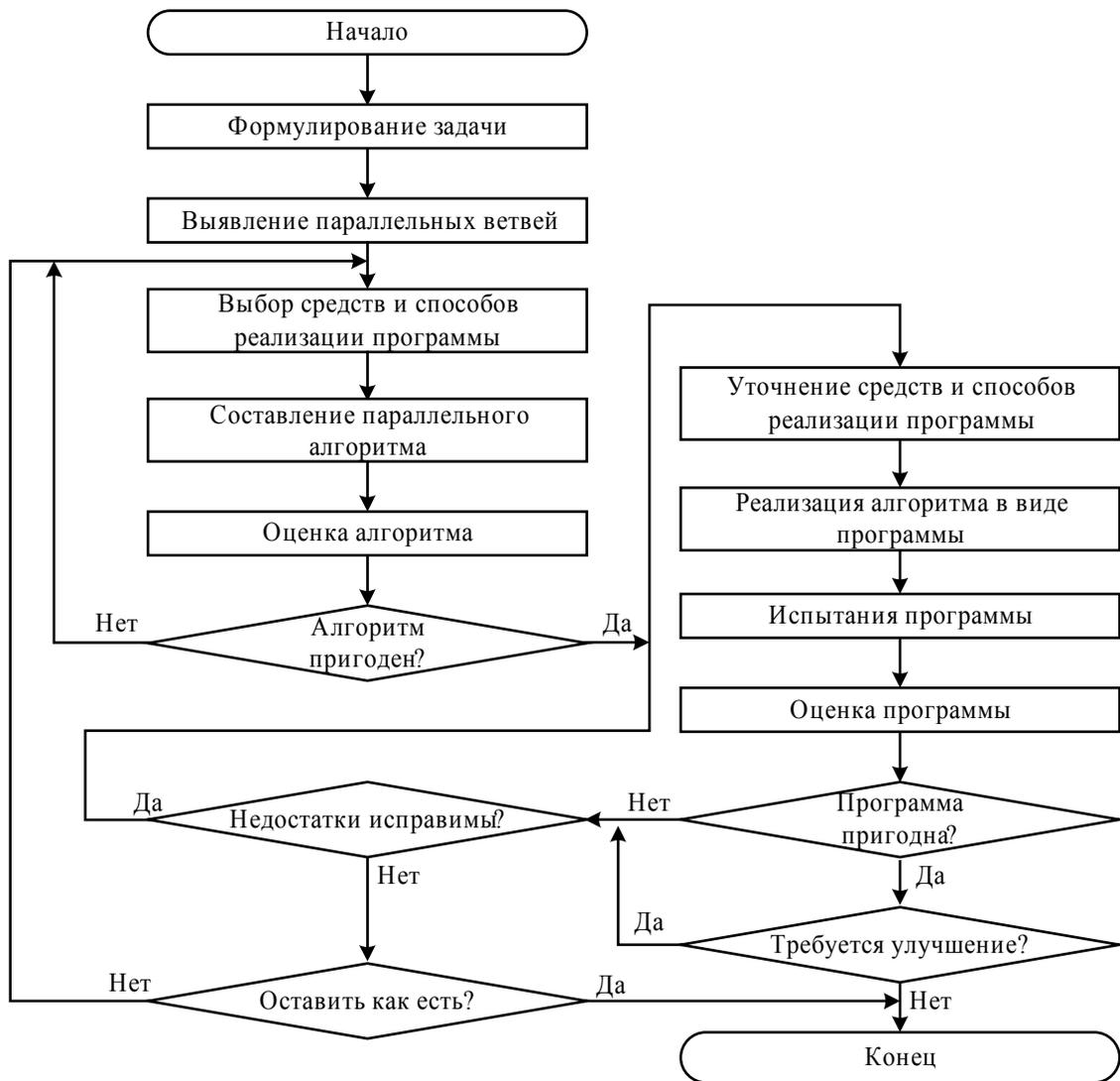


Рис. 1. Схема алгоритма подготовки программы пользователем

Сначала пользователь формулирует задачу и составляет предварительный алгоритм её решения. После этого он производит поиск возможностей параллельного выполнения вычислений (выявляет параллельные ветви) и осуществляется выбор средств (язык, транслятор, средства выполнения, средства взаимодействия и синхронизации ветвей параллельной программы) и способов (какие средства и как применять) программной реализации параллельного алгоритма. На основе полученных сведений о возможности параллельного выполнения вычислений и сведений о системе пользователь формирует параллельный алгоритм решения задачи. Полученный алгоритм оценивается на пригодность его программной реализации с учётом параметров интересующей вычислительной системы.

В случае пригодности алгоритма, пользователь уточняет набор средств и способы его реализации. Реализовав алгоритм в виде параллельной программы, он проводит различные (как правило, натурные) испытания этой программы и оценивает результаты. Оценка результатов позволяет выявлять ошибки, и проверять соответствие программы предъявляемым к ней требованиям, а также определять соответствие её параметров параметрам системы.

В случае положительной оценки программы, может быть поставлен вопрос о её улучшении. Если программа непригодна или её требуется улучшить, то решается вопрос о возможности выявления и исправления недостатков. Если пользователь не видит возможности исправить недостатки, то он должен принять решение о составлении нового алгоритма или на свой страх и риск оставить программу в неизменном виде.

Грамотно провести весь цикл подготовки параллельной программы может лишь очень опытный и искушённый в параллельном программировании пользователь, хорошо знающий особенности вычислительной системы, входящих в неё аппаратных и программных ресурсов, а также особенности средств реализации параллельных программ.

Для пользователей менее опытных в области создания параллельных программ желательно создать средства, облегчающие решение стоящих перед ними проблем, путём организации некоторого сервиса, берущего на себя выполнение следующих функций:

1. выявление возможности параллельного выполнения вычислений;
2. распределение операций по ветвям параллельного алгоритма и выбор средств исполнения этих ветвей в программе (нити, процессы и т.д.);
3. организация взаимодействия ветвей (общая память, передача сообщений, сигналы) и выбор средств (средства OpenMP, MPI) для этого;
4. планирование режимов ввода, вывода и хранения данных;
5. организация и выбор средств для синхронизации ветвей параллельной программы;
6. организации эффективного использования ресурсов вычислительной системы

В данном докладе предлагается следующая функциональная схема реализации сервиса для автоматизированной подготовки задач пользователей предназначенных и адаптированных для решения на кластере. Схема показана на рис. 2. Прямоугольниками, очерченными сплошной линией показаны функции, выполняемые объектами. В число объектов входит пользователь, который ставит свою задачу и модули сервисного программного обеспечения. Функции, выполняемые одним объектом, объединены замкнутой пунктирной линией, в зоне которой курсивом приведено название объекта. На схеме также показаны линии взаимодействия. Рассмотрим функции объектов подробнее.

Пользователь представляет свою задачу в некотором формальном виде. В частности, задача может быть представлена в виде уже готовой последовательной программы или в виде соответствующего формального описания алгоритма. После этого он, с помощью модулей сервиса, осуществляет подбор алгоритма решения поставленной задачи, который наиболее адаптирован к системе и имеющимся средствам программной реализации. На основе этого алгоритма, пользователь создаёт параллельную программу, испытывает её на системе и осуществляет необходимую доводку.

Модуль выявления независимых операций производит анализ задачи и составляет ярусно-параллельную схему обработки данных. При проведении анализа в модуле может быть задействована одна или несколько параллелизирующих методик. Методики могут задействоваться автоматически или в соответствии с выбором пользователя.

Модуль определения параметров системы осуществляет генерацию тестовых программ, которые, выполняясь на системе, позволяют определить временные параметры интересующих вычислительных и коммуникационных операций. Тестовые программы подразделяются на базовые и пользовательские. Первые позволяют получить основные параметры системы. Вторые позволяют получить временные параметры выполнения системой специфических действий, присутствующих в пользовательской программе.



Рис. 2. Схема реализации сервиса для автоматизированной подготовки задач пользователей

Модуль формирования параллельных алгоритмов осуществляет предварительную оценку параметров системы в совокупности с имеющимися средствами реализации программы, после чего формирует набор параллельных алгоритмов решения задачи. Каждый алгоритм строится по своей методике. База методик построения параллельных алгоритмов может расширяться и модифицироваться. Применяемые методики могут быть аналитические, эвристические или шаблонные.

Модуль выбора подходящего алгоритма осуществляет выбор лучшего алгоритма решения пользовательской задачи. Выбор осуществляется из представленного модулем формирования параллельных алгоритмов набора. Для осуществления выбора, модуль выбора подходящего алгоритма, на основе каждого алгоритма формирует модель его программной реализации. Далее происходит оценка этих программных реализаций. При необходимости модуль делает запрос на уточнение временных параметров выполнения системой специфических действий, присутствующих в пользовательской программе. При выборе лучшего алгоритма модуль руководствуется тремя критериями. Первый критерий – это сокращение времени вычислений, второй – это число выделяемых вычислительных ресурсов (узлов, процессоров, ядер), третий – это режим выполнения задачи на системе (конкретный набор режимов зависит от системы). Результатом работы модуля может быть один или несколько (лучшие по своим критериям или по наборам ограничений) рекомендуемых алгоритмов решения задачи на конкретной системе.

При желании, можно задействовать корректирующую обратную связь (на схеме – связь А) между модулями выбора подходящего алгоритма и формирования параллельных алгоритмов.

В МЭИ (ТУ) в настоящее время (с 2007 года) проводятся работы по реализации предлагаемой автором доклада вышеприведенной схемы в виде специализированного программного комплекса позволяющего облегчить адаптацию программ пользователей к выполнению на имеющемся кластере. Предполагается, что в результате появится возможность заметно снизить трудозатраты на разработку параллельных программ пользователей и их отладку. Также ожидается, что предпринимаемые меры позволят снизить число и продолжительность натуральных экспериментов при подготовке программ пользователей.