

# ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ РЕШЕНИЯ ЗАДАЧИ ОПТИМАЛЬНОГО УПРАВЛЕНИЯ НА ПРИМЕРЕ РАЗВИТИЯ ТРАНСПОРТНОЙ СЕТИ НА СУПЕРКОМПЬЮТЕРАХ BLUEGENE/P И «ЛОМОНОСОВ»

Н.Л. Григоренко, А.В. Жарков, Д.Г. Пивоварчук, Н.Н. Попова

**Введение.** Работа посвящена тематике, связанной с решением задач оптимизации управления сложными системами на суперкомпьютерах. Рассматривается программный комплекс для решения задачи оптимизации развития динамической транспортной сети с непрерывным потоком. Задача исследуется для транспортных сетей с большим числом узлов и дуг, для моделирования которых требуется большое количество вычислений.

Проведение данного исследования с одной стороны имеет практическую значимость в виде разработки программного комплекса, позволяющего решать задачи оптимизации управления транспортной сетью с непрерывным потоком, которые могут быть сформулированы в рамках предлагаемой математической модели. Программный комплекс разработан с учетом возможных реальных параметров транспортных сетей. Проведена оценка требуемых объемов памяти в зависимости от параметров модели. В частности, на суперкомпьютере «Ломоносов» возможно решение для транспортных сетей с суммарным числом дуг и ребер порядка 108. Проведено исследование эффективности применения параллельных вычислений в зависимости от размера входных данных. Все эти факторы позволяют успешно решать оптимизационные задачи для транспортных сетей, которых становится всё больше и больше в реальном мире.

С другой стороны, данная работа имеет теоритическую значимость, расширяя спектр решаемых на суперкомпьютерах задач, который традиционно представлен задачами вычислительной гидро- и газодинамики, механики твердого и жидкого тела, задачами биоинформатики и нанотехнологий.

**Постановка задачи.** Сеть поставок с непрерывным потоком представляет собой направленный граф. Каждая вершина графа является одним из следующих объектов: поставщик, потребитель или распределителями. Будем обозначать: поставщиков –  $S_1, \dots, S_{Ns}$ , потребителей –  $D_1, \dots, D_{Nd}$  и распределителей –  $B_1, \dots, B_{Nb}$ .

Направленную связь, соединяющую вершины  $i, j$ , и направленную от вершины  $i$  к вершине  $j$  будем обозначать  $C_{i,j}$ . Соответствующую этой связи пропускную способность будем обозначать  $c_{i,j}$ , а поток через эту связь будем обозначать  $f_{i,j}$ . Например, связь, соединяющую поставщика  $S_i$  и потребителя  $D_j$  обозначим  $C_{S_i, D_j}$ , ее пропускную способность  $c_{S_i, D_j}$ , а поток  $f_{S_i, D_j}$ .

Введём обозначения:

- a) Каждый поставщик  $S_i, i=1..Ns$  генерирует исходящий поток и характеризуется числом  $s_i$ , которое определяет объем генерируемого им потока в единицу времени.
- b) Каждый потребитель  $D, i=1..ND$ , принимает входящий поток и характеризуется числом  $d_i$ , которое определяет объем потока, который этот потребитель может принять в единицу времени.
- c) Каждый распределитель  $B_i, i=1..NB$  принимаемый входящий поток и превращает его в исходящий поток. Распределитель характеризуется числом  $b_i$ , которое определяет объем потока, который этот дистрибьютер может перераспределить.
- d) Потоки, произведенные всеми поставщиками, распределяются по сети в соответствии с некоторым правилом, которое будем называть статическим правилом распределения потоков.
- e) Суммарный поток, ушедший от поставщиков, в полном объеме доходит до потребителей. Причем, не предполагается, что от потребителей уходит весь поток, который он может сгенерировать.

Обозначим множество входящих связей в вершину  $k$  через  $I_k = \{ C_{i,j} | j=k \}$

а множество исходящих связей из вершины  $k$  через  $O_k = \{ C_{i,j} | i=k \}$

Используя введенные обозначения, запишем ограничения, которым должны удовлетворять потоки  $f_{i,j}$  между связями:

$$R1) \quad 0 \leq f_{i,j} \leq c_{i,j} \quad (1)$$

$$R2) \quad \sum_{j \in I_{s_i}} f_{s_i, j} \leq s_i \quad (2)$$

$$R3) \quad \sum_{j \in O_{d_i}} f_{j, d_i} \leq d_i \quad (3)$$

$$R4) \sum_{j \in O_{b_i}} f_{j, b_i} \leq b_i \quad (4)$$

$$R5) \sum_{j \in O_{b_i}} f_{j, b_i} = \sum_{j \in I_{b_i}} f_{j, b_i} \quad (5)$$

Статическое правило распределения потоков задается задачей минимизации

$$\sum_{i=1}^{N_d} [d_i - \sum_{j \in I_{d_i}} f_{j, d_i}] \rightarrow \min_{\{f_{i,j}\}} \quad (6)$$

при ограничениях R1–R5.

Для обозначения времени будем использовать переменную  $t=0,1,\dots,N$ . Считаем, что каждая переменная, зависящая от времени, постоянная на интервале  $[t, t+1)$ ,  $t=0,1,\dots,N-1$ .

Начальные пропускные способности всех связей заданы

$$c_{i,j}(0) = c_{i,j}^0 \quad (7)$$

Предполагаем, что мы имеем возможность управлять пропускными способностями связей с помощью групп связей. Заданные группы связей будем обозначать  $G1 \dots GM$ , где каждая группа характеризуется набором

$$G^q = \{u_{i,j}^q\}, q = 1, \dots, M \quad (8)$$

где  $u_{i,j}^q$  – заданное значение, на которое будет увеличена пропускная способность связи  $C_{i,j}$  из группы  $G_q$ . Предполагаем, что  $M \geq N$ , т.е. количество возможных мероприятий не меньше, чем количество интервалов времени.

Обозначим через  $G(t)$  множество индексов тех групп, которые еще не были задействованы к моменту времени  $t$ , и обозначим через  $\bar{G}(t)$  индексы тех групп, которые уже были задействованы к моменту

$$\text{времени } t, \text{ тогда } \begin{cases} \bar{G}(t+1) = \bar{G}(t) \cup q(t) \\ \bar{G}(0) = \emptyset \end{cases} \quad (9)$$

где управляющий параметр  $q(t)$  может принимать значения только из множества  $G(t)$ .

Множество индексов  $\bar{G}(t)$  для момента времени  $t$  однозначно определяет пропускные способности всех связей в момент  $t$  следующим образом  $c_{i,j}(\bar{G}(t)) = c_{i,j}^0 + \sum_{q \in \bar{G}(t)} u_{i,j}^q$  (10)

Управление развитием инфраструктуры заключается в последовательном выборе на каждом интервале времени группы связей, чьи пропускные способности должны быть увеличены.

Будем предполагать, что задан набор сценариев (функций от времени)  $(s_1(t), \dots, s_{N_s}(t), d_1(t), \dots, d_{N_d}(t), b_1(t), \dots, b_{N_b}(t))$

Для того, чтобы описать целый набор сценариев и каждому из сценариев поставить в соответствие вероятность, воспользуемся понятием марковского процесса. Зададим переходные вероятности между состояниями  $(s_1(t), \dots, s_{N_s}(t), d_1(t), \dots, d_{N_d}(t), b_1(t), \dots, b_{N_b}(t))$  (12)

для всех  $(s_{i+1}, d_{i+1}, b_{i+1}), (s_i, d_i, b_i) \in H$ . Так определенный марковский процесс будет описывать множество сценариев и вероятность каждого сценария.

Получаем следующие ограничения, которые должны выполняться для каждого  $t=0,1,\dots,N$

$$\begin{cases} 0 \leq f_{i,j}(t) \leq c_{i,j}(\bar{G}(t)) \\ \sum_{j \in I_{s_i}} f_{s_i, j}(t) \leq s_i(t) \\ \sum_{j \in O_{d_i}} f_{j, d_i}(t) \leq d_i(t) \\ \sum_{j \in O_{b_i}} f_{j, b_i}(t) \leq b_i(t) \\ \sum_{j \in O_{b_i}} f_{j, b_i}(t) = \sum_{j \in I_{b_i}} f_{j, b_i}(t) \end{cases} \quad (13)$$

Рассмотрим следующую критерий качества

$$L(s_0, d_0, b_0) = E \left[ \sum_{t=0}^N \beta_k \sum_{i=1}^{N_d} (d_i - \sum_{j \in I_{d_i}} f_{j, d_i}(t)) \right] \rightarrow \min_{\{f_{i,j}(t|OMEGA)\}_{t=0,\dots,N}, \{q(t|OMEGA)\}_{t=0,\dots,N-1}} \quad (14)$$

где  $\beta_k > 0$  при  $k=0,\dots,N$ , – весовые коэффициенты упорядоченные следующим образом  $ibeta_0 < ibeta_1 < \dots < ibeta_N$  (15)

Подход к решению задачи основан на методе динамического программирования и линейного программирования и подробнее описан в [1].

Параллельный алгоритм решения задачи. Опишем алгоритм нахождения оптимального решения задачи:

1. Для каждого  $\tau=0,1,\dots,N$ , множества  $\bar{G}\in\bar{G}_\tau$  и набора  $(s,d,b)\in H$  решаем вспомогательную задачу линейного программирования (14) и находим значения  $W(\bar{G},(s,d,b))$ .
2. Для каждого  $\tau=0,1,\dots,N$ , множества  $\bar{G}\in\bar{G}_\tau$  и набора  $(s,d,b)\in H$  вычисляем значения  $V(\bar{G},(s,d,b))$ .
3. Пусть наблюдается последовательность состояний  $\{(s_t,d_t,b_t)\}_{t=0,\dots,N}$ . Строим оптимальную последовательность  $(q_0,\dots,q_{N-1})$  включения групп связей, соответствующую заданной наблюдаемой последовательности состояний.

Для вычисления значений  $W(\bar{G},(s,d,b))$  необходимо вспомогательную задачу линейного программирования (14), которая представляет собой поиск максимального потока в графе транспортной сети при ограничениях R1-R5. Для решения этой задачи предлагается использовать алгоритм Эдмондса-Карпа[2]. Для этого выполняется преобразование графа:

- Добавляется вершина S, которая становится единственным источником потока;
- Добавляется дуга от S к  $S_i, i=1,\dots,NS$  с пропускной способностью  $Cs, si=si$ ;
- Добавляется новая вершина D, которая становится единственным потребителем потока;
- Добавляется дуга от D к  $D_i, i=1,\dots,ND$  с пропускной способностью  $CD, Di=di$ ;
- Каждая вершина  $B_i, i=1,\dots,NB$  заменяется двумя вершинами  $B1_i$  и  $B2_i$ , соединенными дугой из  $B1_i$  в  $B2_i$  пропускной способности  $CB1_i, B2_i=bi$
- Ограничения пропускных способностей  $si, bi, di$  с вершин снимаются;

Для преобразованного графа решается задача поиска максимального потока алгоритмом Эдмондса-Карпа и вычисляется значение  $W(\bar{G},(s,d,b))$ .

Обозначим:

NL - число дуг графа транспортной сети

NS - число поставщиков

ND - число потребителей

NB - число распределителей

E - число состояний марковского процесса

Q=M - число групп, моментов времени

Предложенный алгоритм имеет экспоненциальную сложность относительно числа моментов времени Q, поэтому его использование на реальных моделях транспортных сетей требует большого количества вычислений. С развитием и массовым распространением параллельных вычислительных систем актуальна разработка параллельного алгоритма решения данной задачи. Согласно предложенному последовательному алгоритму, вычисление значений  $W(\bar{G},(s,d,b))$  может быть выполнено независимо на основе входных данных. Этот факт положен в основу предлагаемого параллельного алгоритма. Каждый параллельный процесс вычисляет некоторые значения  $W(\bar{G},(s,d,b))$ . После вычисления всех необходимых значений производится нахождение искомой последовательности оптимального управления (26).

Для реализации параллельного алгоритма необходимо правильно решить проблему загрузки, синхронизации и балансировки параллельных процессов. Из двух основных групп методов решения проблем балансировки: статических и динамических, был выбран метод динамической балансировки, т.к. время решения каждой подзадачи  $T_{subtask}$  априорно неизвестно и может варьироваться в пределах

$$0 \leq T_{subtask} \leq N_V N_L^2.$$

Опишем метод динамической балансировки в предложенном алгоритме. Множество процессов, реализующих параллельный алгоритм, разбивается на две группы: управляющий процесс (в дальнейшем будем называть его master-процессом) и процессы-обработчики (slave-процессы). В контексте решаемой задачи каждый slave-процесс выполняет элементарное задание (подзадачу), то есть решение системы линейных неравенств (18)-(19) с соответствующими локальными параметрами. На master-процесс возложены функции распределения элементарных заданий по обрабатывающим slave-процессам и сбора полученных результатов. Количество slave-процессов фиксировано и равно P-1, где P — число параллельных процессов, определяемое при запуске параллельной программы.



Рис. 1. Диаграмма последовательности действий параллельных процессов

На рисунке 1 представлена диаграмма последовательности действий предложенного алгоритма. На очередном шаге действия «вычисление значений  $W$ » (пункт 3 рис. 1) каждый из slave-процессов ожидает новую подзадачу, решает её, возвращает результат и переходит к ожиданию следующего задания либо завершает свою работу. Отсутствие необходимости выполнения синхронизации между подзадачами позволяет выполнять их независимо. Предложенная схема балансировки позволяет автоматически учитывать возможную неоднородность вычислительной системы, на которой будет выполняться параллельная программа.

Оценка параметров транспортной сети, для которой может быть решена задача оптимизации, определяется объемами оперативной памяти, необходимой для работы алгоритма. Обозначим - число  $N_V = 2 + N_S + N_D + 2N_B$  вершин в преобразованном графе. Для внутреннего представления графа транспортной сети требуется  $K(7N_L + 13N_V)$  байт, информации о состояниях марковского процесса  $KE(E + N_V)$  байт, информации о группах связей  $4KQ$  байт. Коэффициент  $K = \text{sizeof}(\text{long long})$  зависит от используемого компилятора и для проведенного вычислительного эксперимента на суперкомпьютере «Ломоносов»  $K=8$ . Суммарный объем требуемой оперативной памяти равен  $K(7N_L + N_V(E + 13) + E^2 + 4Q)$  байт.

Объемы пересылаемых данных между master-процессом и slave-процессами при решении одной подзадачи оцениваются размером памяти, необходимой для представления номера состояния марковского процесса и индексов задействованных групп, что составляет  $KQ$  байт. Время передачи этих данных будет значительно меньше времени решения системы неравенств для больших транспортных сетей. Например, для транспортных сетей с  $N_V > 1000$ ,  $Q < 1000$  на суперкомпьютере Ломоносов время передачи данных составляет 0.5 мс, а время решения одной подзадачи от 1 секунды.

Программный комплекс. Предложенный параллельный алгоритм был положен в основу программного комплекса, предназначенного для решения задачи оптимального развития динамической транспортной сети с непрерывным потоком. Решение поставленной задачи связано с проведением многочисленных экспериментов, требующих диалогового взаимодействия пользователя с системой, поддержки формирования входных данных, выполнения расчётов и визуализации результатов. С учётом была выбрана структура программного комплекса. Программный комплекс состоит из двух основных модулей: модуль FrontEnd, устанавливаемый на персональный компьютер исследователя и модуль ParallelSolver, устанавливаемый на суперкомпьютерную систему. В качестве языка описания модели транспортной сети предложено расширение языка описания графов Dot. Язык Dot дополнен возможностью описания весов рёбер, матрицы переходных вероятностей и задания групп рёбер. Для визуализации результатов работы выбран пакет GraphViz и предложено его расширение возможностями вывода динамики развития транспортной сети. Для построения оптимальной последовательности развития на основе наблюдаемых состояний разработана программа OptimalSequance, реализующая пункты 2 и 3 описанного выше алгоритма нахождения оптимального решения задачи (26).

В состав модуля FrontEnd включены синтаксический анализатор расширения языка Дот, расширенный пакет GraphViz и инструмент OptimalSequance построения оптимальной последовательности развития на основе последовательности наблюдаемых состояний.

Модуль ParallelSolver, устанавливаемый на суперкомпьютер, включает в себя реализацию параллельного алгоритма вычисления значений матрицы  $W(\bar{G}, (s, d, b))$ .

Пересылка данных между модулями осуществляется пользователем.

Схема взаимодействия модулей программного комплекса при решении задачи показана на рисунке 2.

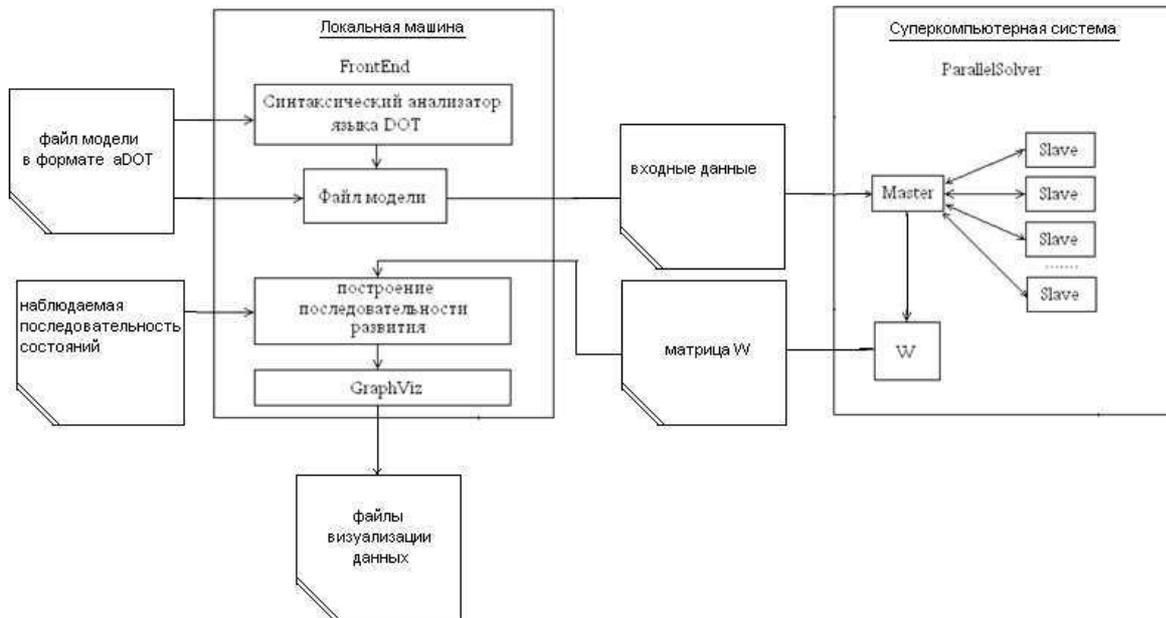


Рис. 2. Схема взаимодействия модулей программного комплекса.

Вычислительный эксперимент. Реализация программного комплекса выполнена на суперкомпьютере «Ломоносов», установленном в Московском государственном университете им. М.В.Ломоносова, а так же на суперкомпьютерах «СКИФ-МГУ», BlueGene/P и BlueGene/Q. Программный комплекс может быть использован и на других многопроцессорных вычислительных системах, поддерживающих стандарт MPI.

Обозначим параметры, характеризующие размер задачи и влияющие на время ее решения:

$N_1 = N_V N_L^2$  – оценка сложности поиска максимального потока алгоритмом Эдмондса — Карпа,

$N_2 = 2^Q E$  – число экстремальных подзадач.

Значения параметра Q для выполнения тестовых расчетов было выбрано в диапазоне  $Q=2..20$ . Ограничения на параметр Q и на размер графа обусловлены максимальным временем, выделяемым для решения задачи в выбранной очереди суперкомпьютера (3 суток).

Связи между вершинами графа были выбраны следующим образом. Общее количество связей NL распределялось на две равные группы:

- $N_L/2$  связей, соединяющих случайно выбранные поставщик и распределитель
- $N_L/2$  связей, соединяющих случайно выбранные распределитель и потребитель

Данное распределение было выбрано в соответствии с методом тестирования алгоритмов поиска максимального потока, предложенного в [3], и таким образом сгенерированные сети относятся к классу random layered networks.

Группы дуг  $G^1 \dots G^Q$  выбраны так, что число связей в каждой группе равно  $N_L/Q$ , то есть все дуги графа транспортной сети равномерно распределены по группам и каждая группа состоит из случайно выбранных дуг графа. В соответствии с формальным описанием модели параметр E описывает число состояний, в которых может находиться динамическая транспортная сеть, и его значение принято  $E=64$ , что соответствует реальным прикладным задачам. Матрица P вероятностей переходов из состояния в состояние задана так, что  $P_{ij}=1/E$ .

Суперкомпьютер «Ломоносов» с пиковой производительностью 1373.1 Тфлопс содержит 43252 ядер. Для работы пользователей доступны несколько разделов, имеющих различные характеристики процессорных узлов. Для решения задачи на каждом из разделов организованы несколько очередей. Вычислительный эксперимент проводился на разделе, содержащем максимальное число ядер, каждый из 4096 узлов которого включает в себя 2 четырехъядерных процессора Intel® Xeon 5570 Nehale. С точки зрения решаемой задачи интерес представляет размер оперативной памяти (12 Гбайт на узел). Выбор параметров тестовых моделей проводился с учетом этих ограничений по размеру оперативной памяти. Суперкомпьютер оснащён коммуникационной сетью типа «fat tree»

Подробный список значений параметров модели Q, NL, NS, ND, NB и соответствующих им значений вычислительной сложности задачи N1, N2 для всех тестовых моделей эксперимента на суперкомпьютере «Ломоносов» приведены в таблице 1.

Таблица 1. Параметры тестовых моделей, сгенерированных для суперкомпьютера «Ломоносов»

№	N1	N2	NL	NS	ND	NB	Q
1	63*109	2048	11*103	130	130	130	6
2	63*109	8192	11*103	130	130	130	7
3	63*109	32768	11*103	130	130	130	9
4	63*109	524288	11*103	130	130	130	13
5	3*1021	128	107	107	107	107	1

В соответствии с формулой суммарного объема требуемой оперативной памяти и учитывая, что  $NL > NV$ , для 12 Гбайт ОЗУ узла суперкомпьютера «Ломоносов» возможно решение задач для транспортных сетей с числом дуг до 108. Для экспериментального подтверждения этой теоретической оценки сгенерирована модель 5 (см. таблицу 1) и для неё решена успешно задача оптимизации.

Ввиду стохастической генерации дуг в графе транспортной сети для каждого набора параметров сгенерировано 5 различных тестовых моделей. Итоговое время решения задачи оптимизации для каждого набора параметров рассчитано как среднее арифметическое по 5 тестовым моделям.

Эффективность параллельных вычислений для предложенного алгоритма определена классическим способом как отношение ускорения к числу параллельных процессов:  $Eff = T1 / Tr * p$ , где T1 – время решения задачи на 1 процессоре, Tr – время решения задачи на P процессорах.

Вычислительный эксперимент был проведен на тестовых задачах с использованием процессоров от 2 до 2048. Узлы суперкомпьютера работали в режиме SMP с одной нитью.

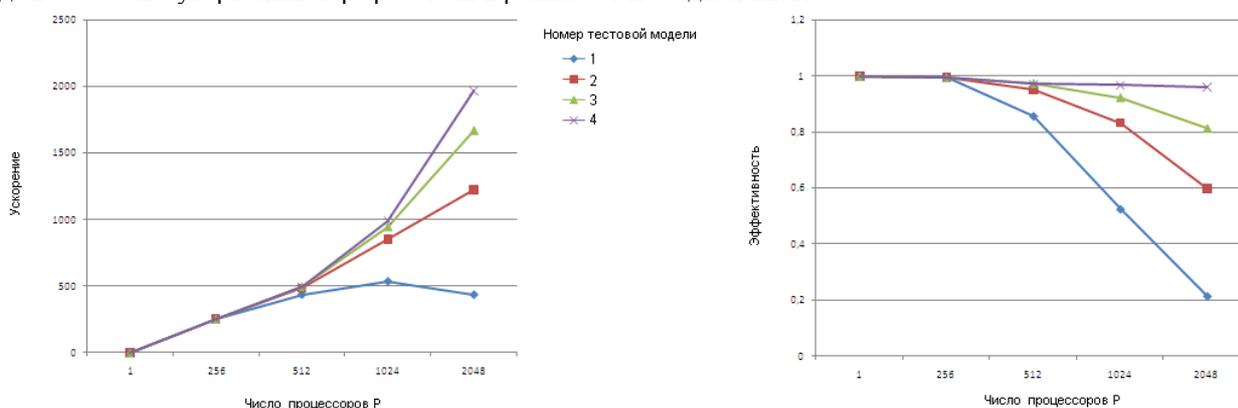


Рис 3. Ускорение при решении тестовых задач различной вычислительной сложности и эффективность параллельных вычислений для суперкомпьютера «Ломоносов»

На рисунке 3 показаны графики ускорения и эффективности параллельных вычислений. На графиках отражены результаты для тестовых моделей 1-4.

Как видно из графиков, на задачах с  $N2 > 32768$  (тестовая модель №4) эффективность параллельных вычислений близка к 1, что подтверждает хорошую сильную масштабируемость предложенного алгоритма.

IBM Blue Gene/P — массивно-параллельная вычислительная система, которая состоит из двух стоек, включающих 8192 процессорных ядер (2 x 1024 четырехядерных вычислительных узлов), с пиковой производительностью 27,9 терафлопс. Суперкомпьютер оснащён двумя коммуникационными сетями – «three-dimensional torus» для операций типа «точка-точка» и «global collective» для коммуникаций типа «один-многим».

Вычислительный эксперимент на IBM Blue Gene/P проводился аналогично эксперименту на системе «Ломоносов». Параметры тестовых моделей приведены в таблице 2. Ускорение и эффективность приведены на рисунках 4 и 5.

Таблица 2. Параметры тестовых моделей, сгенерированных для суперкомпьютера «BlueGene/P»

№	N1	N2	NL	NS	ND	NB	Q
1	63*109	8	11*103	130	130	130	3
2	63*109	32	11*103	130	130	130	5
3	63*109	512	11*103	130	130	130	9
4	63*109	2048	11*103	130	130	130	11
5	63*109	8192	11*103	130	130	130	13
6	63*109	32768	11*103	130	130	130	15

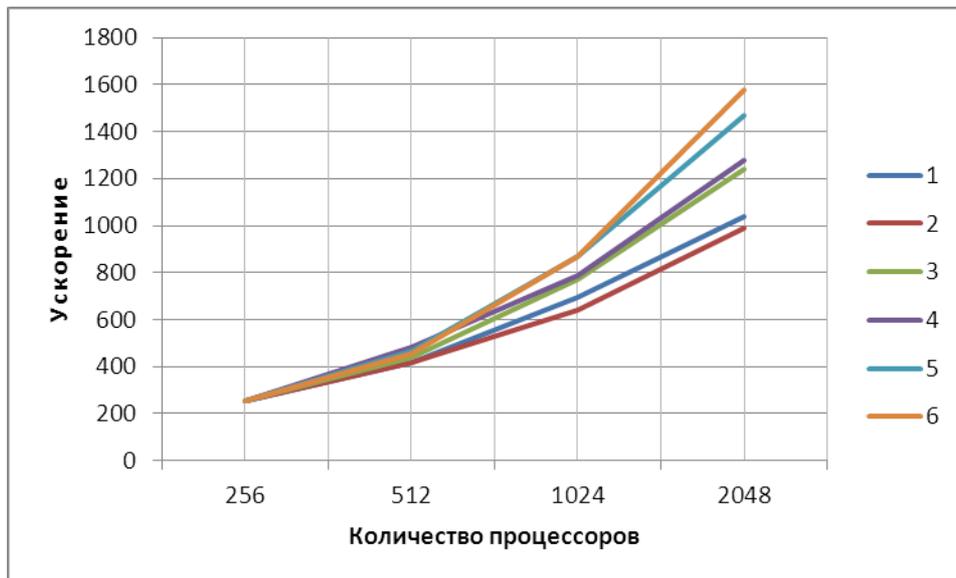


Рис 4. Ускорение при решении тестовых задач различной вычислительной сложности для суперкомпьютера «BlueGene/P»

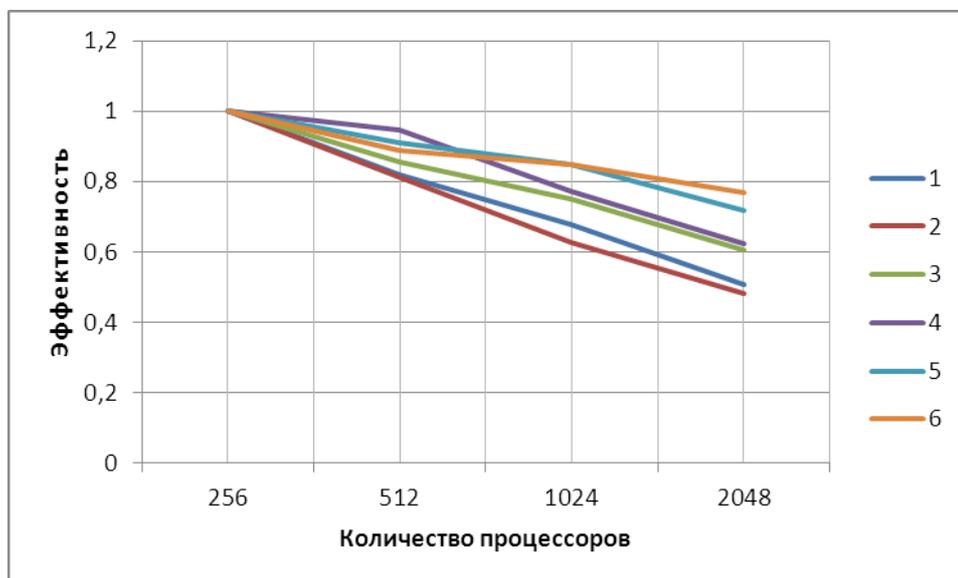


Рис 5. Эффективность при решении тестовых задач различной вычислительной сложности для суперкомпьютера «BlueGene/P»

Применение алгоритма. В качестве иллюстрации использования разработанного программного комплекса опишем решение модельной задачи составления плана регламентных работ на нефтепроводной сети. Сеть показана на рисунке 6.

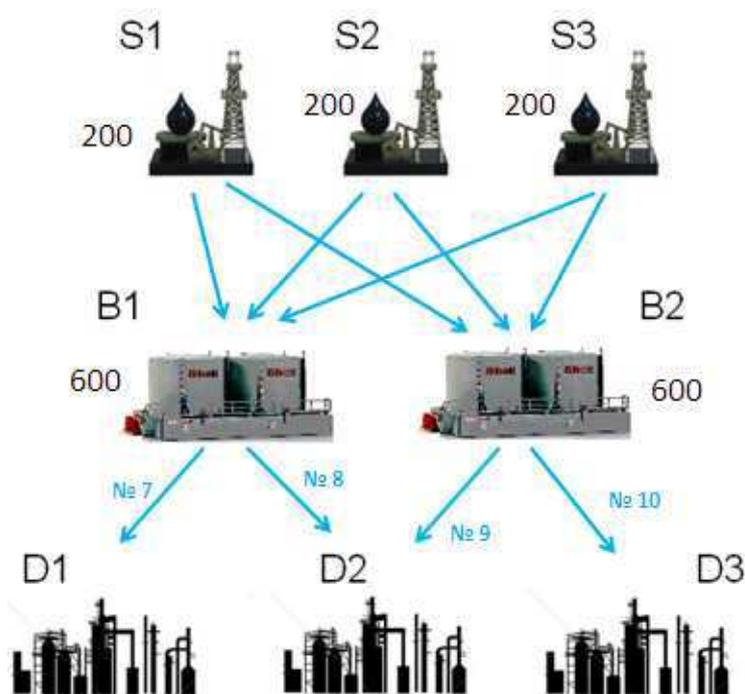


Рис 6. Схема нефтепроводной сети

Сеть состоит из трёх источников S1...S3, двух распределителей B1,B2, трёх потребителей D1...D3. Небольшое число узлов сети выбрано только с целью детальной демонстрации всех шагов алгоритма.

Сеть может находиться в одном из трёх состояний, параметры сети в этих состояниях приведены в таблице 3.

Таблица 3. Состояния нефтепроводной сети

№ состояния	S1	S2	S3	B1	B2	D1	D2	D3
1	200	200	200	600	600	50	1000	50
2	200	200	200	600	600	1000	50	50
3	200	200	200	600	600	50	50	1000

Количество состояний обусловлено сезонными изменениями спроса и потребления нефти в транспортной сети.

Начальные пропускные способности нефтепроводов  $C_i=200$ ,  $i=1...3$ . Имеется три регламентных мероприятия, связанных с увеличением пропускной способности нефтепроводов:

1. Увеличение пропускной способности трубопровода №7,  $C_7=400$
2. Увеличение пропускной способности трубопроводов №8 и №9,  $C_8=400$  и  $C_9=400$
3. Увеличение пропускной способности трубопровода №10,  $C_{10}=100$

Файл графа в формате Dot приведён в таблице 4.

Таблица 4. Листинг файла графа нефтепроводной сети в формате Dot

```

digraph G{
  s1->b1[label="1"];
  s1->b2[label="2"];
  s2->b1[label="3"];
  s2->b2[label="4"];
  s3->b1[label="5"];
  s3->b2[label="6"];
  b1->d1[label="7"];
  b1->d2[label="8"];
  b2->d2[label="9"];
  b2->d3[label="10"];
}

```

Обозначим последовательность выполнения мероприятий  $q=(q_1,q_2,q_3)$ , где  $q_i$  - номер мероприятия, выполняемого в отрезок времени  $[T_{i-1} ; T_i]$

Возможно 6 последовательностей выполнения мероприятий:

- $q_1=1$   $q_2=2$   $q_3=3$

- $q_1=1$   $q_2=3$   $q_3=2$
- $q_1=2$   $q_2=1$   $q_3=3$
- $q_1=2$   $q_2=3$   $q_3=1$
- $q_1=3$   $q_2=1$   $q_3=2$
- $q_1=3$   $q_2=2$   $q_3=1$

В соответствии с предложенным алгоритмом, программный комплекс построил оптимальную последовательность развития сети  $q = (2,1,3)$ . Суммарная разность спроса и предложения (критерий качества) за модельное время  $V_0=1400$ . Графическое представление развития транспортной сети, полученное с помощью инструмента визуализации модуля FrontEnd, показано на рисунке 7.

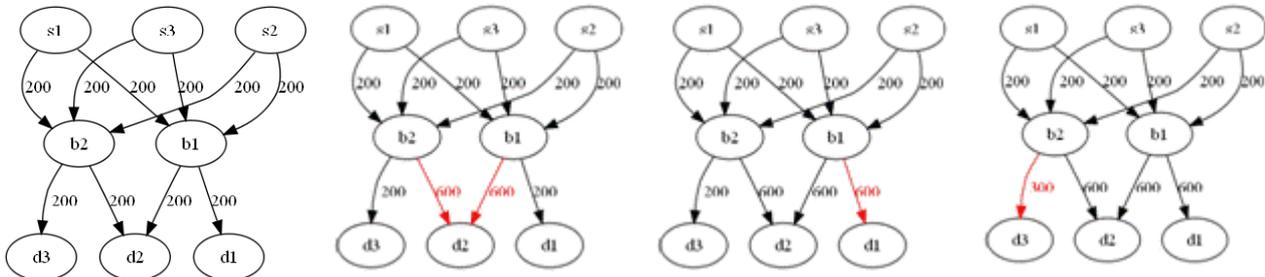


Рис. 7. Графическое представление развития транспортной сети, полученное с помощью инструмента визуализации модуля FrontEnd.

Расчет решения этой модели выполнялся на небольшом числе процессоров и приводится в данной статье для иллюстрации подхода. Такой подход, связанный с быстрым расчетом небольших моделей, создаёт возможность одновременного проведения многих экспериментов для одной и той же транспортной сети с различными параметрами количества моментов времени и комбинаций одновременно проводимых мероприятий по развитию сети.

**Заключение.** В заключение следует отметить, что в работе описан программный комплекс для проведения моделирования развития транспортной сети. Проведено исследование разработанного параллельного алгоритма оптимального управления динамической транспортной сетью. Проведено экспериментальное исследование эффективности алгоритма на суперкомпьютере «Ломоносов».

Определены параметры входных данных, влияющие на время ее решения:  $N_1$  – оценка сложности поиска максимального потока алгоритмом Эдмондса-Карпа,  $N_2$  – число экстремальных задач. Проведенный вычислительный эксперимент показал высокую (близкую к 1) эффективность алгоритма на задачах с большим количеством моментов модельного времени и состояний системы  $N_2 > 215$  и размером графа транспортной сети. Рассмотрено решение реальной задачи построения плана регламентных работ для нефтепроводной сети. Применение суперкомпьютерных вычислений позволяет решать реальные транспортные сети с большим количеством узлов и дуг, состояний и моментов модельного времени.

Так как в настоящее время число узлов суперкомпьютеров растёт, в дальнейшем планируется провести исследование эффективности алгоритма, положенного в основу программного комплекса, при количестве задействованных узлов более 214 и возможности усовершенствования схемы балансировки загрузки для минимизации накладных расходов на передачу данных.

Работа выполнена при поддержке грантов РФФИ № 11-07-00756 и №11-07-00614.

#### ЛИТЕРАТУРА:

1. А. В. Жарков, Д. Г. Пивоварчук. Разработка и исследование параллельного алгоритма решения задачи оптимизации развития инфраструктуры типа поставщик-потребитель // Труды Пятой Международной конференции «Параллельные вычисления и задачи управления» PACO'2010. Институт проблем управления им. В.А. Трапезникова РАН, 2010, С. 433-446.
2. Jack Edmonds, Richard M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems // Journal of the ACM 19 (2), 1972, pp. 248–264.
3. Ravindra K. Ahuja, Murali Kodialam, Ajay K. Mishra, James B. Orlin. Computational investigations of maximum flow algorithms // European Journal of Operational Research 97, 1997, pp. 509-542