Приближенное решение задачи оптимального распределения трафика в коммуникационной сети с топологией «многомерный тор»

А. В. Мукосей
АО «НИЦЭВТ»

Задача оптимального размешивания трафика в сети с топологией «многомерный тор» с отказами, на данный момент, не имеет аналитического решения, а число вариантов перебора растет экспоненциально от размера задачи. Для обеспечения отказоустойчивости многоузловых систем при возникновении отказов необходимо найти решение данной задачи за реальное время. В работе предложен алгоритм решения задачи, находящий приближенное решение за полиномиальное время. В работе проведено исследование предложенного алгоритма.

Ключевые слова: Отказоустойчивость, коммуникационные сети, многомерный тор, детерминированная маршрутизация, маршрутизация с порядком направлений, распределения трафика.

1. Введение

В АО «НИЦЭВТ» разработана высокоскоростная коммуникационная сеть Ангара [1,2] с топологией «многомерный тор». В маршрутизаторе сети реализована бездедлоковая, адаптивная маршрутизация, основанная на правилах «пузырька» (Bubble flow control, [5]) и «порядка направлений» (Direction ordered routing, DOR, [6, 7]) с использованием битов направлений [7]. Благодаря алгоритму First Step/Last Step «нестандартного первого и последнего шага» [7] аппаратно поддерживается обход отказавших узлов или линков. Эффективность этого метода по поддержанию связности в сети с отказами была показана в статье [3]. Применяемая маршрутизация позволяет избежать взаимных блокировок из-за циклических зависимостей пакетов в кольцах и между кольцами нескольких измерений, а также гарантирует сохранение порядка передачи пакетов между любыми двумя адресатами.

Для эффективного использования вычислительных узлов системы, необходимо уметь оптимально выделять ресурсы в зависимости от состояния кластера. Состояние кластера изменяется постоянно по различным причинам: занятость отдельных узлов и неисправность оборудования. В статье [4] автором предложен алгоритм выбора оптимального подмножества узлов в коммуникационной сети. Этот алгоритм разбит на этапы, в первом из этапов перебираются подмножества узлов, на втором этапе предлагается способ проверки множества на маршрутизируемость и построения таблицы маршрутизации при помощи алгоритма, основанного на анализе графа путей. Однако указнный алгоритм построения таблиц маршрутизации даже для очень простых случаев выдает плохие решения. В данной статье предложен генетический алгоритм построения таблицы маршрутизации, распределяющий трафик значительно лучше.

Статья организована следующим образом. В разделе 2 приводятся необходимые формальные определения. В разделе 3 описывается маршрутизация в сети Ангара. В разделе 4 представлена постановка задачи. В разделе 5 описан разработанный алгоритм. В разделе 6 проведено исследование построенных алгоритмов.

2. Определения

В данном разделе вводятся некоторые формальные определения, которые в дальнейшем будут использоваться в статье.

Рассмотрим коммуникационную сеть с топологией многомерный тор. Множество всех узлов сети обозначим N, размерности тора обозначим $(d_1, d_2, ..., d_n)$, а общее число узлов — $|N|$. Каждый узел u имеет координаты $(u_1, u_2, ..., u_n)$, где $0 \le u_i < d_i$. Соседними в рамках тороидальной топологии будем называть узлы $u = (u_1, u_2, ..., u_n)$ и $\tilde{u} = (u_1, ..., (u_j \pm 1) \mod d_j, ..., u_n)$ для любого индекса $1 \le j \le n$.

Легко заметить, что каждый узел имеет $2n$ соседей (в случае $d_j > 2$). Будем считать, что каждый из этих соседей находится в одном из *направлений* от узла u.

Множество направлений обозначим $\mathcal{D}$ и пронумеруем их числами $\overline{1, 2n}$:

$$\mathcal{D} = \{\Delta_j\}_{j=\overline{1,2n}}.$$

Направления с номерами $1, ..., n$ будем называть *положительными*:

$$\tilde{u} = (u_1, ..., (u_j + 1) \mod d_j, ..., u_n),$$
$$\Delta_j = (0, ..., \underbrace{1}_{\text{позиция } j}, ..., 0) \in \mathcal{D},$$

где $1 \le j \le n$.

Направления с номерами $n + 1, ..., 2n$ будем называть *отрицательными*:

$$\tilde{u} = (u_1, ..., (u_{j-n} - 1) \mod d_{j-n}, ..., u_n),$$
$$\Delta_j = (0, ..., \underbrace{-1}_{\text{позиция } j-n}, ..., 0) \in \mathcal{D},$$

где $n + 1 \le j \le 2n$. В такой формулировке выражение «узел u находится в направлении D от узла v записывается как $u = v + D$».

На множестве направлений $\mathcal{D}$ введем порядок в соответствии с указанной нумерацией: $\Delta_i < \Delta_j$, если $i < j$.

Каждый узел сети может выступать в роли: *активного* — узел выполняющий инжекцию/эжекцию пакетов в сеть и участвующий в вычислениях; *транзитного* — узел, не участвующий в вычислениях и необходимый для поддержания связности системы. Множество активных узлов будем помечать индексом "a", транзитных — "t".

Определение 1. *Каналом связи* (*линком*) будем называть пару (u, D), где $u \in N$, $D \in \mathcal{D}$. Множество всех каналов связи обозначим $\mathcal{E} = N \times \mathcal{D}$.

Определение 2. *Путем* $\mathcal{P}$, соединяющим два узла сети: u^0 и u^n, назовем последовательность вида $u^0, D_1, u^1, D_2, ..., D_l, u^l$, где u_i — узел сети, D_i — направления, связывающее узел u^{i-1} с узлом u^i, l — длина пути. При этом $u^1, ..., u^{n-1}$ — *транзитные узлы пути* $\mathcal{P}$. Так как транзитные узлы могут быть получены из соответствующих переходов, то их можно опустить, тогда подобный путь будет записываться в виде: $u^0, D_1, D_2, ..., D_n$.

Определение 3. Подмножество $M_{(a,t)} = M_a \cup M_t$ множества узлов N назовем *маршрутизируемым*, если для любых двух узлов u, v множества M_a существует путь $\mathcal{P}$ из u в v такой, что транзитные узлы этого пути принадлежат $M_{(a,t)}$.

Определение 4. *Таблицей маршрутизации* $\mathcal{R}$ маршрутизируемого множества $M_{(a,t)}$ назовем некоторый набор путей таких, что для любых двух узлов u, v множества M_a в $\mathcal{R}$ существует единственный путь из u в v такой, что транзитные узлы этого пути принадлежат $M_{(a,t)}$.

Определение 5. Пусть для некоторого маршрутизируемого множества $M_{(a,t)} \subset N$ построена таблица маршрутизации $\mathcal{R}$. *Загруженностью канала связи* $G_{(u,D)}$ маршрутизируемого множества $M_{(a,t)}$ будем называть количество путей, которым принадлежит данный канал связи: $G_{(u,D)} = \mid \{\mathcal{P}_{ij} \mid (u,D) \in \mathcal{P}_{ij}, \mathcal{P}_{ij} \in \mathcal{R}\} \mid$.

Определение 6. *Минимальной таблицей маршрутизации* R_{min} назовем такую таблицу маршрутизации, которая состоит из набора путей минимальных длин.

Определение 7. *Идеальной загруженностью линка* $G_{perfect}$ назовем среднюю загруженность линков в системе с минимальной таблицей маршрутизации R_{min}: $G_{perfect} = \frac{\sum G_{(u,D)}}{|G_{(u,D)}|}$.

Определение 8. *Отклонением* $\sigma(\eta, R)$ *степени* η системы с таблицей маршрутизации R от идеальной загрузки назовем: $\sigma(\eta, R) = \sqrt[\eta]{\frac{1}{|G_{(u,D)}|} \sum\limits_{\lambda \in G_{(u,D)}} (G_{perfect} - \lambda)^{\eta}}$.

3. Маршрутизация в сети Ангара

3.1. Правило порядка направлений с использование битов направлений

Среди алгоритмов маршрутизации для многомерных торов можно выделить класс алгоритмов, соблюдающих *правило порядка направлений*: маршрут между любой парой узлов включает движения в направлениях в определенном, заранее заданном, порядке. Эти алгоритмы обладают свойством отсутствия взаимных блокировок между кольцами нескольких измерений тора при любом количестве одновременных запросов на передачу данных по сети.

Во введенных обозначениях правило порядка направлений будет формулироваться следующим образом: $D_{j-1} \leq D_j, j = \overline{2, N}$, где N — *длина* пути.

Чтобы задать путь, удовлетворяющий правилу порядка направлений, необходимо задать стартовую вершину и количество шагов в каждом из направлений, т.е. набор $u^0, s_{\delta(1)}, s_{\delta(2)}, ..., s_{\delta(i)}$, где $u^0 \in N$ — стартовый узел, $s_{\delta(1)}, s_{\delta(2)}, ..., s_{\delta(i)} > 0$ — количество шагов в направлениях $\Delta_{\delta(1)}, \Delta_{\delta(2)}, ..., \Delta_{\delta(i)}$ таких, что $\Delta_{\delta(j)} < \Delta_{\delta(j+1)},\ j = \overline{1, i-1}$.

В сети Ангара реализована маршрутизация с использованием *битов направлений*, которая вносит некоторые ограничения на маршрутизацию с правилом порядка направлений. Аналогично правилу порядка направлений для задания пути, соответствующему маршрутизации с использованием битов направлений, необходимо задать стартовую вершину и количество шагов в выбранных направлениях, то есть следующий набор: $u^0, s_{\delta(1)}, s_{\delta(2)}, ..., s_{\delta(i)}$, где $u^0 \in N$ — стартовый узел, $s_{\delta(1)}, s_{\delta(2)}, ..., s_{\delta(i)} > 0$ — количество шагов в направлениях $\Delta_{\delta(1)}, \Delta_{\delta(2)}, ..., \Delta_{\delta(i)}$. При этом в наборе направлений $\Delta_{\delta(1)}, \Delta_{\delta(2)}, ..., \Delta_{\delta(i)}$ нет направлений с противоположными знаками и $\Delta_{\delta(j)} < \Delta_{\delta(j+1)},\ j = \overline{1, i-1}$. Обозначим такой набор направлений как D_{dirbit}. Путь, соответствующий маршрутизации с использованием битов направлений, обозначим P_{dirbit}.

3.2. First Step/Last Step

Метод First Step/Last Step [7] используется в сети Ангара как механизм обхода отказавших узлов. Он расширяет маршрутизацию с использованием битов направлений путем добавления первого и последнего нестандартного шага.

Путь с использованием первого и последнего нестандартного шага будет записываться следующим образом: $u^0, D_{FS}, P_{dirbit}, D_{LS}$, где u^0 — стартовый узел, D_{FS} — первое положительное нестандартное направление, D_{LS} — последнее отрицательное нестандартное направление. При этом набор направлений $D_{FS}, D_{dirbit}, D_{LS}$ удовлетворяет правилу порядка направлений.

Таким образом, для однозначного задания пути в сети Ангара необходимо задать набор $D_{FS}, P_{dirbit}, D_{LS}$.

4. Постановка задачи

Во время работы разделяемого пользователями вычислительного кластера необходимо при любом состоянии системы уметь предоставлять требуемое число узлов, которые должны быть маршрутизируемы между собой, транзитный трафик не должен затрагивать узлы вне этого набора, если это возможно. Состояние системы определяется набором отказавших линков и/или узлов и наличием занятых узлов. Занятый или отказавший узел можно интерпретировать как узел, у которого линки сломаны во всех направлениях.

Обозначим множество сломанных линков $F \subset \mathcal{E}$.

Так как физический канал связи между двумя узлами v и u представляет собой линки от узла v к узлу u и наоборот, то разумно предположить, что при неисправности одного из линков — второй так же неисправен. Таким образом, множество F будет включать в себя отказавшие каналы связи попарно.

Во введенных определениях задача будет формулироваться следующим образом. Пусть задан тор с размерностями $(d_1, ..., d_n)$ и набором отказавших линков F. Пусть заданно исследуемое маршрутизируемое множество узлов $M_{(a,t)}$. Необходимо построить алгоритм поиска таблицы маршрутизации с минимальным значением отклонения $\sigma(\eta, R)$.

5. Алгоритмы решения задачи

Допустим, что число путей между двумя узлами ограничено некоторым числом N_{paths}, тогда существует $N_{paths}^{(|M_a|-1)|M_a|}$ различных таблиц маршрутизаций. Даже при небольшом числе узлов сети и вариантов путей число различных таблиц маршрутизации очень велико, и требуется специальный алгоритм для выбора таблиц маршрутизации. Дополнительную сложность добавляет несимметричность алгоритмов маршрутизации и несимметричность системы ввиду наличия отказов.

Для решения поставленной задачи разработано несколько алгоритмов. Один из алгоритмов — генетический алгоритм.

5.1. Генетический алгоритм

Генетический алгоритм — это эвристический алгоритм поиска, основанный на идеях эволюционных теорий. Переменные, характеризующие решение задачи, представлены в виде генов в хромосоме индивидуума. Генетический алгоритм оперирует конечным множеством решений (популяцией) и генерирует новые решения как различные комбинации генов индивидуумов, используя такие операции как отбор, рекомбинация (кроссинговер) и мутация.

Сведем нашу задачу к понятиям генетического алгоритма:

- *Ген* — функциональная единица индивидуума, в текущей задаче это путь $\mathcal{P}_{ij}$ между двумя узлами. Обозначим ген $\mathcal{G}_k^l$, где k — позиция гена в индивидууме, характеризующая пару узлов u_i и u_j (так как число узлов конечно, то всевозможные пары узлов можно однозначно отобразить на набор натуральных чисел), l — вариант пути между узлами u_i и u_j;

- *Индивидуум* — набор генов, которые определяют основные свойства организма. В данном случае это таблица маршрутизации с набором путей (генов). Каждый индивидуум можно представить как вектор генов:

$$\mathcal{I} = (\mathcal{G}_1^{\lambda(1)}, \mathcal{G}_2^{\lambda(2)}, ..., \mathcal{G}_{L-1}^{\lambda(L-1)}, \mathcal{G}_L^{\lambda(L)}),$$

  где $L = (|M_a|-1)|M_a|$ длина вектора, равная числу путей, $\lambda(k)$ — функция вариантов путей, выбранных для индивидуума $\mathcal{I}$;

- *Популяция* — набор различных индивидуумов: $\mathcal{P} = (\mathcal{I}_1, ..., \mathcal{I}_S)$, где S — размер популяции;
- *Кроссинговер* — операция, при которой индивидуумы обмениваются частью генов не меняя их позицию в индивидууме;
- *Мутация* — эвристическая операция изменения индивидуума путем случайного изменение гена или набора генов. Под изменением гена подразумевается замена его вида на другой возможный;
- *Пригодность* — функция качества индивидуума, экстремум которой необходимо найти. В нашем случае это минимум отклонения η-степени $\sigma(\eta, R)$.

Общая схема используемого генетического алгоритма представлена на рисунке 1. Начальная популяции генерируется случайным образом.

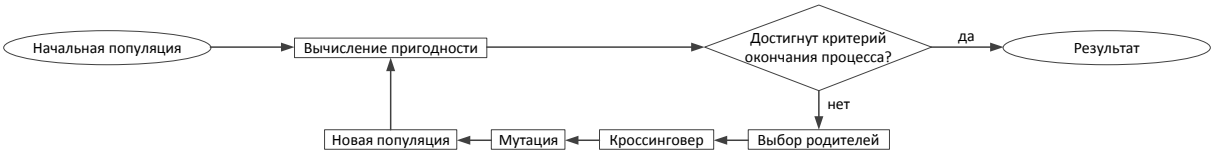


Рис. 1. Общая схема генетического алгоритма.

В разработанном генетическом алгоритме реализована операция выбора родителя — *панмиксия*. В соответствии с ним каждому члену популяции сопоставляется случайное целое число на отрезке $[1, S]$, где S — размер популяции. Это число — номер партнера из популяции. При таком выборе некоторые члены популяции не будут участвовать в процессе размножения, так как образуют пару сами с собой. Какие-то члены примут участие в процессе размножения неоднократно.

После выбора партнеров происходит размножение путем двухточечного кроссинговера. Для этого, в индивидууме выбираются случайно две точки и происходит обмен между особями наборами генов, ограниченных этими точками. Таким образом получается два новых потомка. Обозначим точки кроссинговера соответственно $\mathcal{K}_1$ и $\mathcal{K}_2$. Тогда из двух родителей $\mathcal{I}_{parent_1}$ и $\mathcal{I}_{parent_2}$:

$$\mathcal{I}_{parent_1} = (\mathcal{G}_1^{\lambda(1)}, \mathcal{G}_2^{\lambda(2)}, ..., \mathcal{G}_{L-1}^{\lambda(L-1)}, \mathcal{G}_L^{\lambda(L)})$$
$$\mathcal{I}_{parent_2} = (\mathcal{G}_1^{\tau(1)}, \mathcal{G}_2^{\tau(2)}, ..., \mathcal{G}_{L-1}^{\tau(L-1)}, \mathcal{G}_L^{\tau(L)})$$

получим двух потомков:

$$\mathcal{I}_{child_1} = (\mathcal{G}_1^{\lambda(1)}, ..., \mathcal{G}_{\mathcal{K}_1-1}^{\lambda(\mathcal{K}_1-1)}, \mathcal{G}_{\mathcal{K}_1}^{\tau(\mathcal{K}_1)}, ..., \mathcal{G}_{\mathcal{K}_2-1}^{\tau(\mathcal{K}_2-1)}, \mathcal{G}_{\mathcal{K}_2}^{\lambda(\mathcal{K}_2)}, ..., \mathcal{G}_L^{\lambda(L)})$$
$$\mathcal{I}_{child_2} = (\mathcal{G}_1^{\tau(1)}, ..., \mathcal{G}_{\mathcal{K}_1-1}^{\tau(\mathcal{K}_1-1)}, \mathcal{G}_{\mathcal{K}_1}^{\lambda(\mathcal{K}_1)}, ..., \mathcal{G}_{\mathcal{K}_2-1}^{\lambda(\mathcal{K}_2-1)}, \mathcal{G}_{\mathcal{K}_2}^{\tau(\mathcal{K}_2)}, ..., \mathcal{G}_L^{\tau(L)}).$$

После процедуры размножения к потомкам применяется операция мутации. Каждый ген с вероятностью q_m может мутировать. Партнеры и потомки образуют новую популяцию, к которой применяется операция селекции: для каждого индивидуума вычисляется функция качества $\sigma(\eta, \mathcal{I})$, в новой популяции останутся индивидуумы с наименьшим значением качества. Наилучшую сходимость алгоритм показал на отклонение степени 4.

Критерии окончания процесса выбраны следующим образом:

1. Максимальное число поколений (итераций алгоритма) — 30 поколений;
2. Скорость сходимости функции качества — $0,05$, т.е. алгоритм завершается, если наилучшее значение качества отличается от наилучшего значения качества следующего поколения менее, чем на $0,05$.

Оценка сложности алгоритма складывается из оценки каждого этапа. Обозначим размер исходной популяции как S.

Операцию панмиксия можно оценить как $T_{parent\_selection} = O(S)$ операций бросания кости. В результате работы алгоритма образуется максимум S пар.

Оценка сложности операции размножения $T_{selection} = O(2L)$ операций размещения генов в потомках. Эту операцию необходимо проделать для каждой пары, то есть не более S раз.

Оценка сложности операции мутации реализуется за $T_{mutation} = O(L)$ операций бросания кости. Эту операцию необходимо проделать для каждого нового индивидуума, которых можно оценить как $2S$.

Для того, чтобы вычислить функцию пригодности, необходимо вычислить загруженность каждого линка в системе. Каждый путь проходит через некоторый набор линков, число которых можно оценить сверху как максимальная длина пути в сети: $L_{max} = \sum_{i=1}^{n} d_i$. Для вычисления загруженности всей системы необходимо $T_{loading} = O(LL_{max})$ операций. Для вычисления отклонения степени η необходимо еще $O(|\{(u, D)\}|) = O(2n|M_{(a,t)}|)$ операций. Итого $T_{fitness} = O(|M_a|^2 + 2n|M_{(a,t)}|)$. Необходимо вычислить функцию пригодности для каждого нового индивидуума.

На последней итерации нам необходимо отсеять все плохие организмы и оставить S наилучших. Размер текущей популяции (после этапа размножения) можно оценить как $3S$, где одна часть — это старые особи, и две части — это новое поколение. Алгоритм сортировки оценивается как $T_{evolution} = O(3S\log(3S))$.

В результате на одну итерацию алгоритма необходимо $T_{GA} = T_{parent\_selection} + ST_{selection} + 2ST_{mutation} + 2ST_{fitness} + T_{evolution} = O(S + 2S|M_a|^2 + 2S|M_a|^2 + 2S(|M_a|^2 + 2n|M_{(a,t)}|) + 3S\log(3S)) = O(6S|M_a|^2)$ операций.

5.2. Алгоритм на основе анализа графа путей

При построении маршрута между двумя узлами ограничение на принятие решения о следующем шаге вносит предыстория пути. Рассмотрим движение по некоторому пути в торе в направлениях: $\Delta_{\delta(1)}, ..., \Delta_{\delta(i)}$. Этот путь можно продолжить только в таком направлении $\Delta_{\delta(i+1)}$, что набор направлений $\Delta_{\delta(0)}, ..., \Delta_{\delta(i)}, \Delta_{\delta(i+1)}$ удовлетворяет правилу с использованием битов направлений или $\Delta_{\delta(i)} \leq \Delta_{\delta(i+1)}$ в случае, если $\Delta_{\delta(i+1)}$ является последним нестандартным шагом.

Поэтому для описания вычислительного узла u^i в графе построим множество U^i вершин, которые будут характеризовать предысторию путей, которые проходят через вычислительный узел u^i. Во множество U^i входят вершины U^i_{begin} и U^j_{end}, характеризующие начало и конец пути P_{ij} (инжекцию и эжекцию пакета). Вершины U^i соединяются ребрами соответственно проходящими путям через узел u_i из других узлов.

В статье [4] приводиться подробное описание графа. Главное свойство графа состоит в том, что из узла $u^i \in M_{(a)}$ в узел $u^j \in M_{(a)}$ существует путь $\mathcal{P}$ тогда и только тогда, когда в графе $G(V, E)$ существует путь из вершины U^i_{begin} в вершину U^j_{end}.

Предложенный алгоритм построения таблицы маршрутизации устроен следующим образом. Предположим, что все линки узлов множества $M_{(a,t)}$ имеют нулевую загруженность. Для каждого узла u маршрутизируемого множества M_a в графе $G(V, E)$ запускается алгоритм поиска вширь — BFS. После окончания поиска из каждого узла множества M_a необходимо подняться по построенному дереву обратно вверх к узлу u, увеличивая при этом загруженность $G_{u,D}$ проходимых линков сети. Эвристически выяснено, что относительно сбалансированная таблица маршрутизации получается, если в качестве следующего узла для запуска поиска вширь выбирать максимально удаленный узел от узла u. Вторая эвристика, введенная для получения более равномерной загрузки линков, заключается в

сортировке вершин на каждом новом слое поиска вширь по возрастанию загруженности линков, соответствующих вершинам.

Алгоритмическая сложность построения таблицы маршрутизации составляет $T_R = O(|M_{(a,t)}|^2)$.

6. Исследование

На данный момент наиболее актуальным с практической точки зрения является решение задачи для систем с числом узлов до 80. Исследование проводилось в два этапа: на первом этапе рассматривались системы без сломанных линков, на втором этапе рассматривались системы со случайно сломаными линками, но при сохраненной связности.

Для исследования на первом этапе рассмотрены всевозможные системы с максимальной размерностью тора $\max\limits_{i\in\overline{1,n}} d_i \leq 5$, степенью $2 \leq n \leq 4$ и числом узлов $\prod\limits_{i\in\overline{1,n}} d_i \leq 80$.

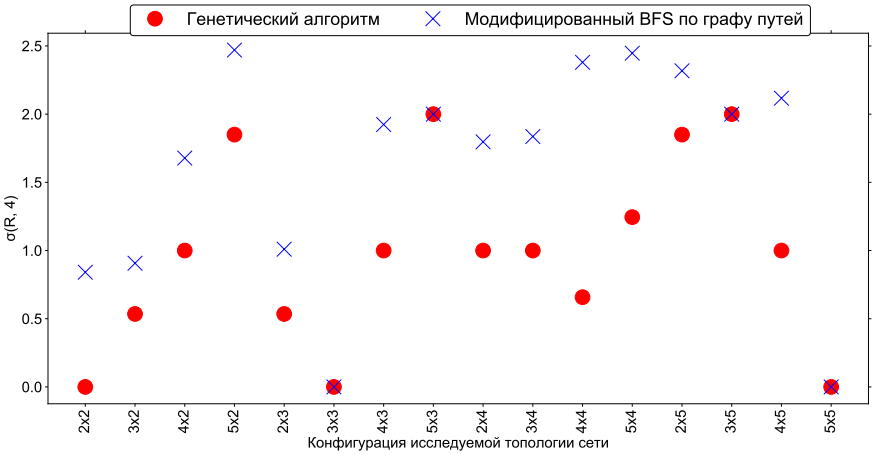


Рис. 2. Значения характеристик $\sigma(4, R)$, полученной для таблиц маршрутизации, построенных разными алгоритмами на различных двухмерных торах.

На рисунках 2, 3, 4 представлены значения характеристик $\sigma(4, K)$ таблиц маршрутизации построенных на различных конфигурациях торов с помощью генетического алгоритма и алгоритма, основанного на BFS. На рисунках видно, что генетический алгоритм в каждом из вариантов построил таблицу маршрутизации со значением характеристики меньше либо равную, чем алгоритм на основе BFS. На рисунках присутствуют точки в которых характеристика достигла своего наилучшего значения 0. Это системы 2x2 и 2x2x2x2, которые характерны короткими измерениями тора, вследствие чего возможно равномерное распределение. Системы 3x3, 5x5, 3x3x3 обладают нечетными измерениями, распределить трафик в такой системе не составляет труда. Заметим, что на графиках приведены не все системы, а только некоторая выборка, но аналогичные рассуждения можно применить и к остальным точкам, которые были опущены для удобства восприятия.

В процентном соотношение генетический алгоритм предложил на двухмерных, трехмерных и четырехмерных торах таблицу маршрутизации на которой характеристика $\sigma(4, R)$ лучше, чем алгоритм, основанный на BFS соответственно на 36,61%, 16,63% и 15,45%. Необходимо отметить, что для простых случаев предложенный генетический алгоритм выдает значительно лучшие решения, чем алгоритм, основанный на BFS.

Для исследования на втором этапе, выбраны системы с конфигурацией 3x3x4 и 4x4x5. Для каждой из систем рассматривалось различное число сломанных линков, и исследовались только маршрутизируемые случаи. Для каждого значения числа сломанных линков

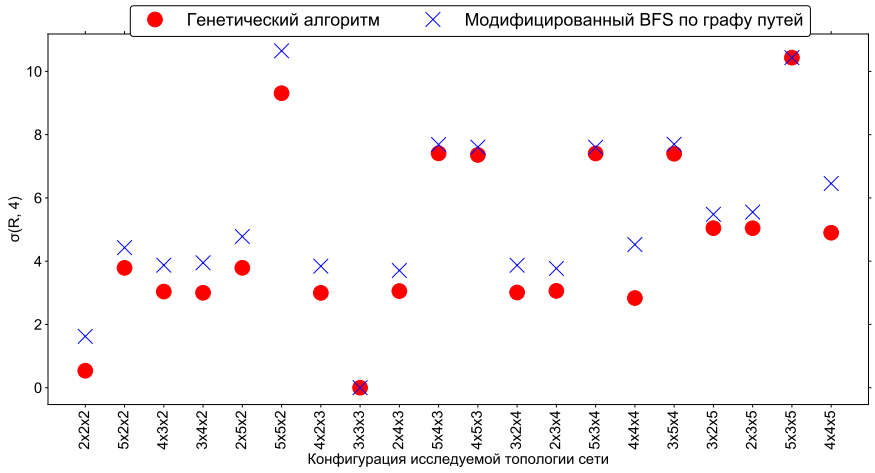


Рис. 3. Значения характеристик $\sigma(4, R)$, полученной для таблиц маршрутизации, построенных разными алгоритмами на различных трехмерных торах.

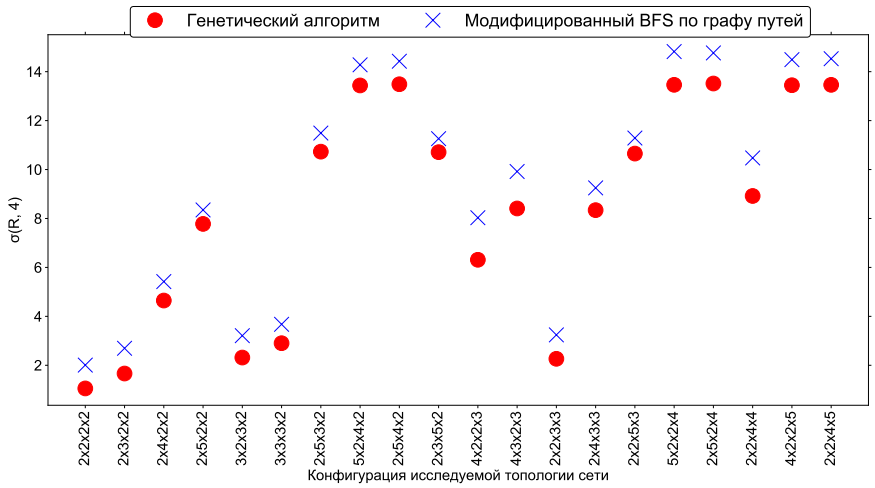


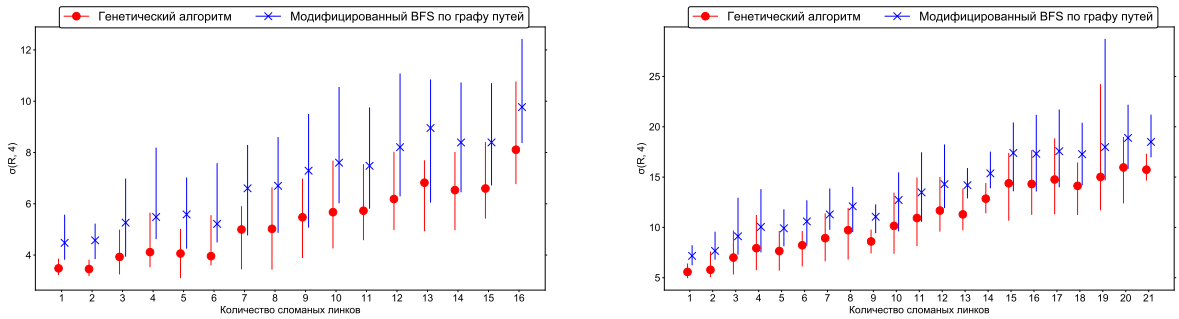
Рис. 4. Значения характеристик $\sigma(4, R)$, полученной для таблиц маршрутизации, построенных разными алгоритмами на различных четырехмерных торах.

случайно генерировалось 10 различных систем.

На рисунке 5a представлен разброс получаемых значений характеристики $\sigma(4, R)$ различными алгоритмами на различных системах с фиксированным числом сломанных линков на системе 3×3×4. Нижняя точка столбика соответствует наилучшему найденному решению в одном из 10 вариантов при фиксированном числе сломанных линков. Верхняя точка, соответственно, — наихудшему, а маркер на столбике соответствует среднему значению по всем случаям. На рисунке видно, что генетический алгоритм во всех случаях позволил получить лучшее решение по сравнению с алгоритмом, основанном на BFS. В среднем выигрыш $\sigma(4, R)$ составляет в среднем 23,78%.

На рисунке 5b рассматривается система 4×4×5. Здесь генетический алгоритм также позволил предложить всегда лучшее решение, в среднем — на 19,48%.

Время работы генетического алгоритма на системе 3×3×4 составило в среднем 0,2 секунды, время работы алгоритма, основанного на BFS — 0,001 секунды. ДляF системы 4×4×5 эти времена составляют соответственно: 1,24 и 0,01 секунды. Для рассматриваемых вычисли-



(a) Система 3x3x4

(b) Система 4x4x5

Рис. 5. Разброс значений характеристик, полученных на таблицах маршрутизации, найденных с помощью исследуемых алгоритмов.

тельных систем с числом узлов не более 80 данное время работы можно признать удовлетворительным, при необходимости ускорения работы генетического алгоритма возможна его алгоритмическая оптимизация, а также распараллеливание реализации при помощи технологии OpenMP.

Заключение

В данной работе описано два полиномиальных алгоритма построения сбалансированных таблиц маршрутизации: генетический алгоритм и алгоритм, основанный на BFS.

Предложенный генетический алгоритм показал лучшие результаты на всех рассмотренных случаях, в среднем по системам улучшение характеристики составило от 16,6% до 36,6%. Для простых случаев с небольшим числом узлов предложенный генетический алгоритм выдает значительно лучшие решения, чем алгоритм, основанный на BFS. Для более сложных случаев, по предварительным оценкам, данное улучшение является существенным, однако его роль будет оценена на различных коммуникационных паттернах в дальнейших исследованиях. Также в дальнейших исследованиях будет произведена попытка научиться выяснять, является ли построенная таблица маршрутизации оптимальной. В условиях, когда оптимальность таблицы маршрутизации под вопросом, разработанный генетический алгоритм является лишь приближенным.

Работа выполнена под руководством к.т.н. А.С. Семенова.

Литература

1. Корж А.А., Макагон Д.В., Жабин И.А., Сыромятников Е.Л. Отечественная коммуникационная сеть 3D-тор с поддержкой глобально адресуемой памяти для суперкомпьютеров транспетафлопсного уровня производительности // Параллельные вычислительные технологии (ПаВТ'2010): Труды международной конференции (Уфа, 29 марта 2 апреля 2010 г.). Челябинск: Издательский центр ЮУрГУ, 2010. С. 227–237.

2. Жабин И., Макагон Д., Симонов А. и др. Кристалл для Ангары // Суперкомпьютеры. — 2013. — Т. зима-2013. — С. 46–49.

3. Пожилов И.А., Семенов А.С., Макагон Д.В. Алгоритм определения связности сети с топологией «многомерный тор» с отказами для детерминированной маршрутизации // Программная инженерия. — 2015. — № 3. — С. 13–19.

4. Мукосей А.В., Семенов А.С., Макагон Д.В. Приближенный алгоритм выбора оптимального подмножества узлов в коммуникационной сети Ангара с отказами //

Параллельные вычислительные технологии (ПаВТ'2016): Труды международной конференции (Архангельск, 28 марта – 1 апреля 2016 г.).

5. Puente V., Beivide R., Gregorio J.A., Prellezo J.M., Duato J., Izu C. "Adaptive bubble router: a design to improve performance in torus networks,"// Parallel Processing, 1999. Proceedings. 1999 International Conference on , vol., no., pp.58,67, 1999.

6. Adiga N.R., Blumrich M., Chen D. et al. Blue Gene/L torus interconnection network // IBM Journal of Research and Development. 2005. — Vol. 49, no. 2.3. — P. 265–276.

7. Scott S.L., et al. The Cray T3E Network: Adaptive Routing in a High // Performance 3D Torus. — 1996.

An approximate solution of optimal traffic load balancing problem in «multi-dimensional» torus topology network

A.V. Mukosey

JSC «NICEVT» (Moscow)

The problem of interconnect load balancing for multidimensional torus topology network with failures doesn't have an analytical solution. The complete enumeration method requires too much time, but fault tolerance providing for multi-node systems with failures requires a small time of finding the problem solution. The paper presents an algorithm for solving the problem of finding an approximate solution in polynomial time.

Keywords: Fault tolerance, communication networks, multidimensional torus, deterministic routing, direction ordered routing, network load balancing.

References

1. Korzh A.A., Makagon D.V., Zhabin I.A., Syromyatnikov E.L. Otechestvennaya kommunikatsionnaya set' 3D-tor s podderzhkoy global'no adresuyemoy pamyati dlya superkomp'yuterov transpetaflopsnogo urovnya proizvoditel'nosti [Russian 3D-torus Interconnect with Support of Global Address Space Memory]. Parallelnye vychislitelnye tekhnologii (PaVT'2010): Trudy mezhdunarodnoj nauchnoj konferentsii (Ufa, 29 marta – 2 aprelya 2010) [Parallel Computational Technologies (PCT'2010): Proceedings of the International Scientific Conference (Ufa, Russia, March, 29 – April, 2, 2010)]. Chelyabinsk, Publishing of the South Ural State University, 2010. P. 527–237.

2. Zhabin, I.A. Kristall dlya Angary [Angara Chip] / I.A. Zhabin, D.V. Makagon, A.S. Simonov // Superkomp'yutery [Supercomputers]. —Winter-2013. — P. 46–49.

3. Pozhilov I.A., Semenov A.S., Makagon D.V. Algoritm opredeleniya svyaznosti seti s topologiyey "mnogomernyy tor"s otkazami dlya determinirovannoy marshrutizatsii [Connectivity problem solution for direction ordered deterministic routing in nD torus]. // Software Engineering. — 2015. — № 3. — С. 13–19.

4. Mukosey A.V., Semenov A.S., Makagon D.V. Priblizhennyy algoritm vybora optimal'nogo podmnozhestva uzlov v kommunikatsionnoy seti Angara s otkazami [Approximate algorithm for choosing the best subset of nodes in the «Angara» interconnect with failures]. Parallelnye vychislitelnye tekhnologii (PaVT'2016) : Trudy mezhdunarodnoy konferentsii (Arkhangel'sk , 28 marta - 1 aprelya 2016) [Parallel Computational Technologies (PCT'2016): Proceedings of the International Scientific Conference (Arkhangelsk, 28 March – 1 April 2016 г.).

5. Puente V., Beivide R., Gregorio J.A., Prellezo J.M., Duato J., Izu C. "Adaptive bubble router: a design to improve performance in torus networks,"// Parallel Processing, 1999. Proceedings. 1999 International Conference on , vol., no., pp.58,67, 1999.

6. Adiga N.R., Blumrich M., Chen D. et al. Blue Gene/L torus interconnection network // IBM Journal of Research and Development. 2005. — Vol. 49, no. 2.3. — P. 265–276.

7. Scott S.L., et al. The Cray T3E Network: Adaptive Routing in a High // Performance 3D Torus. — 1996.