УДК 004.75:004.896:004.021
DOI 10.17513/snt.39821

БАЛАНСИРОВКА РАСПРЕДЕЛЕННОЙ ВЫСОКОНАГРУЖЕННОЙ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМА КЕРНИГАНА-ЛИНА

Трамов И.Б., Ерёмин О.Ю., Степанова М.В., Шульман В.Д.

Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет), Москва,
e-mail: islam_tramov@mail.ru, ereminou@bmstu.ru, stepanova@bmstu.ru, vitalian42@mail.ru

Аннотация: в работе рассматривается вопрос балансировки распределенной высоконагруженной системы с использованием алгоритма Кернигана-Лина для разбиения графов. Рассмотрены основные аспекты и особенности процесса балансировки нагрузки в распределенных вычислительных системах. В работе проведена формализация задачи разрезания графа на подграфы, в результате которой сформирована графовая модель распределенной вычислительной системы как множество взаимосвязанных графов (подграфов). Разрезание графа производится при помощи алгоритма Кернигана-Лина, особенностью которого является высокая точность, быстродействие, а также возможность распараллеливания, использования метаэвристик и работы в режиме реального времени. Составлена схема алгоритма Кернигана-Лина, а также детально описан каждый из его шагов. Особенностью системы, разработанной для демонстрации функционирования алгоритма, является возможность настройки алгоритма разбиения графа путем изменения его параметра. Для оценки качества разбиения выделен ряд метрик, которые демонстрируют ключевые аспекты разбиения графа, такие как количество и средний вес внешних связей, а также плотность каждого подграфа. На основе полученных результатов можно оценить качество разбиения сети и определить, является ли данный вариант разбиения оптимальным или близким к оптимальному.

Ключевые слова: алгоритм Кернигана-Лина, балансировка нагрузки, облако, разбиение графа, графовая модель, метрики

BALANCING OF A DISTRIBUTED HIGH-LOAD SYSTEM USING THE KERNIGAN-LIN ALGORITHM

Tramov I.B., Eremin O.Yu., Stepanova M.V., Shulman V.D.

Bauman Moscow State Technical University (National Research University), Moscow,
e-mail: islam_tramov@mail.ru, ereminou@bmstu.ru, stepanova@bmstu.ru, vitalian42@mail.ru

Annotation: this paper considers the balancing of a distributed high-load system using the Kernighan-Lean algorithm for graph partitioning. The main aspects and peculiarities of load balancing in distributed computing information systems are considered. The paper formalizes the problem of graph cutting into subgraphs. As the result the graph model of distributed computing information system as a set of interconnected graphs (subgraphs) is formed. Graph cutting is performed by means of the Kernighan-Line algorithm which features high accuracy, performance, and capabilities of paralleling, metaheuristics and real-time operation. A flowchart of Kernighan-Lean algorithm is made, and each of its steps is described in detail. A feature of the system developed to demonstrate the functioning of the algorithm is the possibility of tuning the graph partitioning algorithm by changing its parameter. To evaluate the quality of partitioning, it was selected a number of metrics, which demonstrate key aspects of graph partitioning, such as the number and average weight of external links, as well as the density of each subgraph. Based on the results obtained, it is possible to evaluate the quality of network partitioning and determine whether a given partitioning variant is optimal or close to optimal.

Keywords: Kernighan-Lean algorithm, load balancing, graph partitioning, graph model, graph metrics

В наши дни количество интернет-пользователей стремительно увеличивается. В связи с этим растёт и объем мирового трафика. Согласно прогнозу компании Cisco, в 2023 году число интернет-пользователей превысит отметку в 5 миллиардов. В то же время прогнозируемый объём трафика достигнет отметки в 4.8 зеттабайта [1]. Такой рост активности в Интернете также влечёт за собой увеличение нагрузки на дата-центры. С каждым годом число облачных распределенных вычислительных систем растёт, т.к. крупные компании работают над обеспечением доступности своих продуктов при возникновении высоких нагрузок. Один из способов повышения степени доступности – это использование технологий балансировки нагрузки, что определяет актуальность данной работы. Целью представленного исследования является подход, позволяющий осуществлять балансировку нагрузки распределенной вычислительной системы, представленной графовой моделью, путем разрезания графа на подграфы при помощи алгоритма Кернигана-Лина. Программные системы балансировки нагрузки призваны решить данную проблему, обеспечив качественный доступ пользователей к ресурсам.

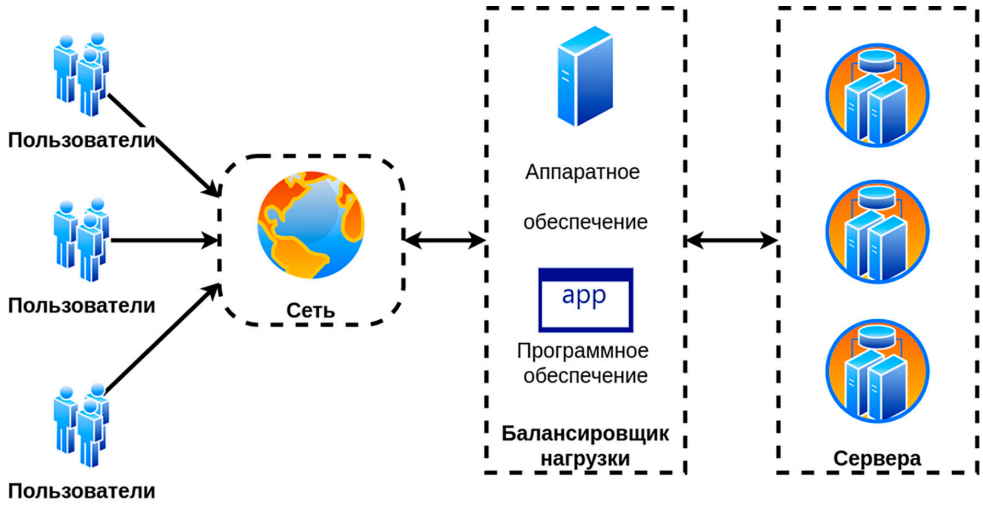

Рис. 1. Общая архитектура балансировщика нагрузки

Балансировщик нагрузки

Процесс балансировки нагрузки заключается в распределении задач между несколькими устройствами (например, серверами [2; 3]) с целью оптимизации использования ресурсов, отказоустойчивости, а также сокращения времени обслуживания запросов [4]. В облачных распределенных вычислительных системах балансировщик нагрузки распределяет нагрузку между узлами сети (облака) [5; 6]. На рисунке 1 представлена общая архитектура балансировщика нагрузки.

Согласно сетевой модели OSI, различные сетевые устройства могут взаимодействовать друг с другом на семи уровнях. В свою очередь, балансировка нагрузки происходит на уровнях 4 и 7: транспортный и прикладной уровни соответственно [7].

На транспортном уровне балансировщик нагрузки занимается отслеживанием сетевой информации о портах и протоколах (UDP и TCP). Доставка трафика происходит при помощи использования ряда алгоритмов балансировки. На прикладном уровне балансировщик нагрузки активно используется в связке с протоколами HTTP/HTTPS. На этом уровне происходит проверка всего трафика [8].

Для улучшения качества доступа используются методы и алгоритмы балансировки нагрузки:

- хэш-подход;
- алгоритм наименьшего времени;
- липкий метод;
- метод наименьшего количества подключений;
- метод циклического перебора.

Представленные алгоритмы определяют наиболее подходящие серверы для приема входящих клиентский запросов.

Целью исследования является экспериментальная проверка возможности балансировки распределенной высоконагруженной системы с использованием алгоритма Кернигана-Лина.

Графовая модель распределенной системы

Программная подсистема балансировки нагрузки должна обрабатывать входную схему сети и представлять её в виде модели графа. В таком случае проблема балансировки рассматривается как задача разрезания графа на подграфы.

Разрезание графа на подграфы – представление исходного графа $G = \{X, U\}$ в виде множества подграфов, таких что

$$\bigcup_{i=1}^{N} X_i = X,$$

где N – количество вершин в исходном графе. Таким образом, все вершины исходного графа должны быть распределены по подграфам. Также следует определить два условия:

– $\bigcup_{i=1}^{N} X_i : X_i \neq \varnothing$: ни один подграф не должен быть пустым;

– $\forall i \neq j : X_i \cap X_j$: одна и та же вершина не может входить в разные подграфы.

Для задачи разрезания графа не важна структура графа: граф может быть неориентированный или ориентированный, взвешенный или невзвешенный.

Для решения полученной задачи разбиения графа предлагается воспользоваться алгоритмом Кернигана-Лина [9]. Данный алгоритм отлично подходит для обработки

взвешенных неориентированных графов. Помимо этого, данный подход позволяет нам напрямую управлять процессом разбиения, оперируя параметром C. Данный параметр отвечает за максимальное количество обменов пар вершин для получения новых вариантов разбиения.

На рисунке 2 представлена схема алгоритма Кернигана-Лина.

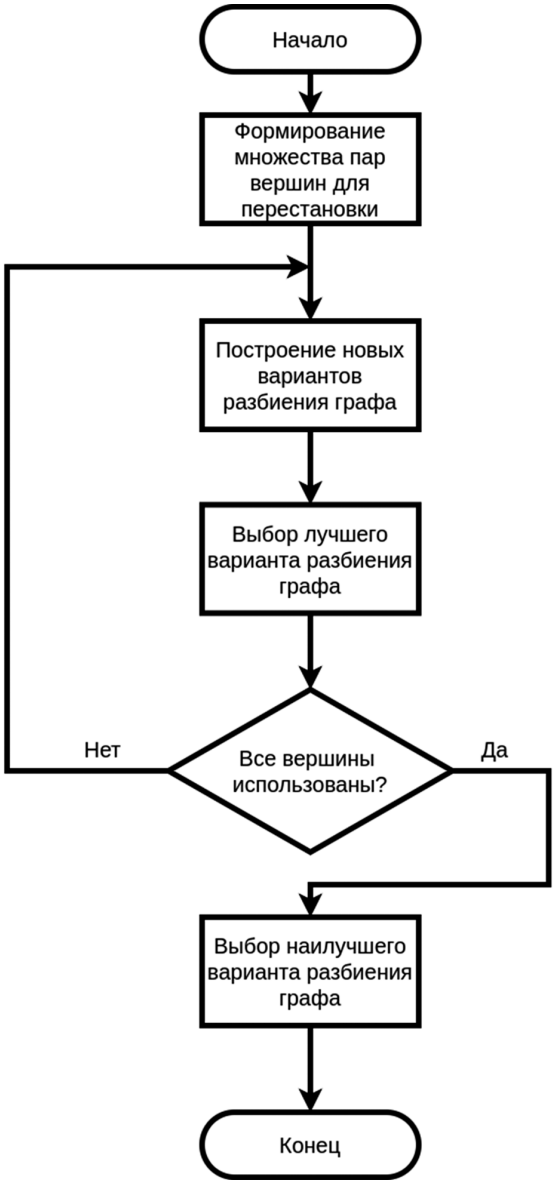

Рис. 2. Схема алгоритма Кернигана-Лина

Алгоритм Кернигана-Лина:

Шаг 1. Формирование множества пар вершин для перестановки. Из незадействованных на данной итерации вершин формируются все возможные пары (при этом в каждой паре должно присутствовать по одной вершине из каждой части имеющегося разбиения).

Шаг 2. Построение новых вариантов разбиения графа. Для получения множества новых вариантов деления производится поочередный обмен пар вершин между частями имеющегося разбиения графа.

Шаг 3. Выбор лучшего варианта разбиения графа. Выбирается лучший вариант среди множества новых делений графа (сформированного на шаге 2). Подходящий вариант далее используется как новое текущее разбиение графа. Соответствующая выбранному варианту пара вершин считается использованной на текущей итерации алгоритма.

Шаг 4. Проверка использования всех вершин. При наличии в графе не использованных вершин (не участвовавших в перестановках) выполнение итерации алгоритма снова продолжается с шага 1. В противном случае, если перебор графа завершен, то следует приступить к шагу 5.

Шаг 5. Выбор наилучшего варианта разбиения графа. Среди всех разбиений графа выбирается наилучший вариант разбиения.

Шаг 6. Конец алгоритма.

Результат работы алгоритма

Для оценки качества разбиения графа необходимо выделить ряд графовых метрик. В таблице представлены метрики для анализа [10].

Метрики оценки качества разбиения графа

Метрика	Формула
Среднее количество внешних рёбер	-
Средний вес внешних рёбер	-
Плотность	$d = \frac{2m}{n(n-1)}$
Транзитивность	$T = 3\frac{\#triangles}{\#triads}$
Средний коэффициент кластеризации	$C = \frac{1}{n}\sum_{v \in G} c_v$
Обозначения: n – количество узлов, m – количество ребер, *#triangles* – количество треугольников в графе, *#triads* – количество триад в графе, c_v – коэффициент кластеризации для узла v	

Для демонстрации работы алгоритма Кернигана-Лина проведен эксперимент, в ходе которого случайный граф был разрезан на 128 частей. Значение параметра C равно 16 (таким образом, потребуется 16 перестановок пар вершин для определения нового разбиения). Рассматриваемый

алгоритм и модель сети были реализованы на языке программирования Python.

Перед проведением эксперимента был выдвинут ряд предположений о качестве полученного разбиения:

- при увеличении количества разбиений количество внешних рёбер уменьшается;
- при увеличении количества разбиений средний вес внешних рёбер уменьшается;
- при увеличении количества разбиений плотность подграфов растёт.

На рисунке 3 продемонстрировано изменение количества внешних связей при увеличении количества разбиений. Исходя из результатов, показанных на рисунке 3, можно сделать вывод о правильности первого предположения о качестве разбиений графа.

Аналогичная ситуация наблюдается и при измерении зависимости среднего веса внешних рёбер от количества разбиений, что наглядно представлено на рисунке 4.

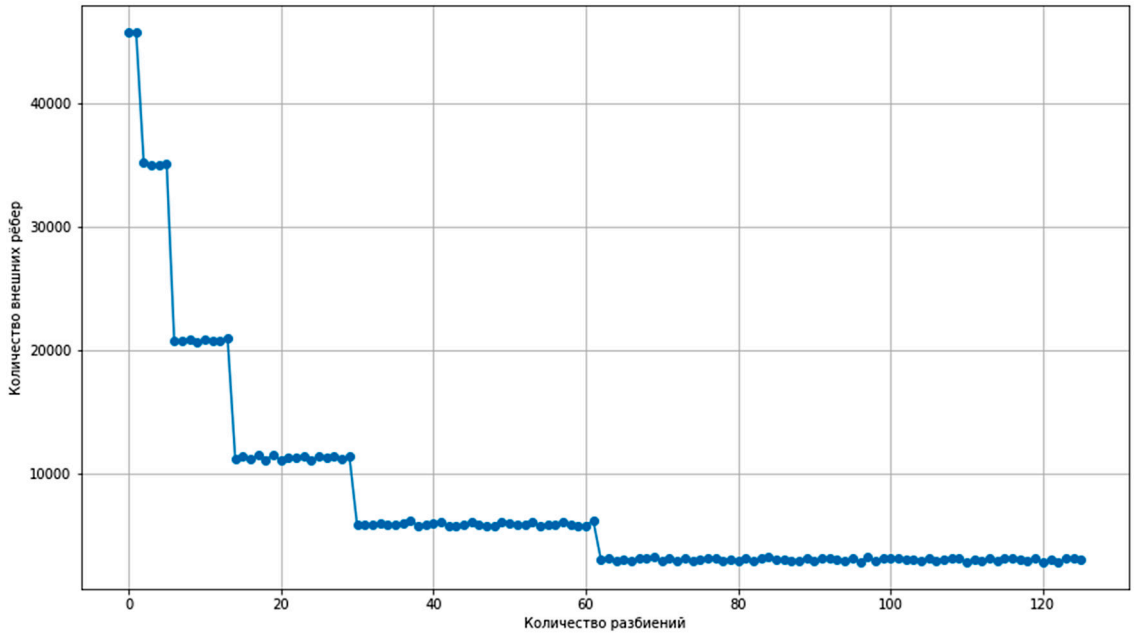

Рис. 3. График зависимости количества внешних связей от количества разбиений

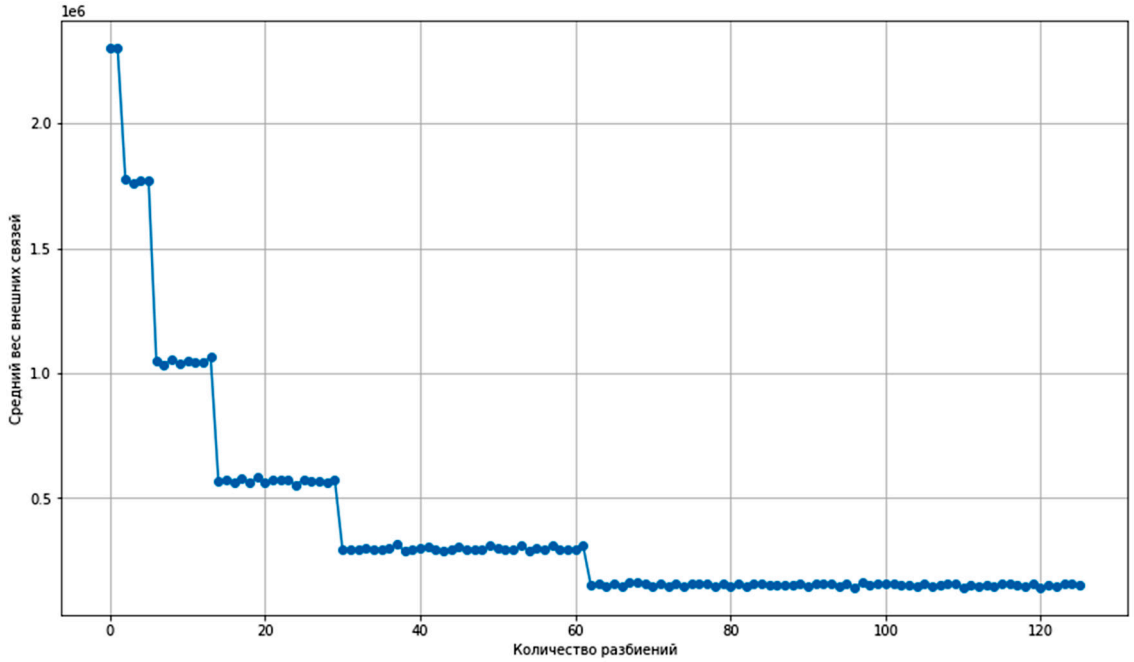

Рис. 4. График зависимости среднего веса внешних рёбер от количества разбиений

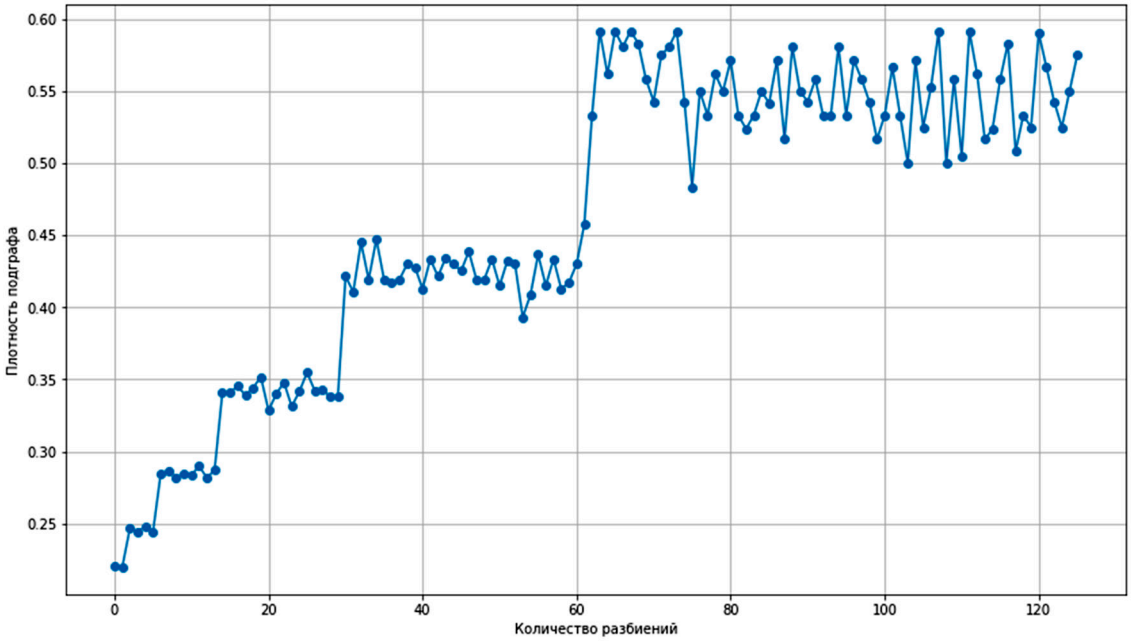

Рис. 5. График зависимости плотности разбиений от количества разбиений

Действительно, при увеличении количества подграфов средний вес внешних связей уменьшается, что и подтверждает второе предположение о качестве разбиения.

Иная картина наблюдается относительно плотности графов. График зависимости плотности разбиений от количества разбиений представлен на рисунке 5.

Исходя из результатов, представленных на рисунке 5, можно судить о том, что средняя плотность подграфов растет пропорционально количеству разбиений.

По результатам эксперимента видно, что алгоритм Кернигана-Лина для разбиения графа позволяет обеспечить качество разбиения, как и ожидалось. Все предположения, выдвинутые перед экспериментом, оказались верны.

Заключение

В работе представлена графовая модель распределенной системы, а также алгоритм, позволяющий решить поставленную задачу разрезания графа.

Предложенный подход позволяет разбить сеть на подсети с учётом представленных ограничений (например, минимальное количество узлов в сети). Также особенностью предложенного подхода является возможность тонкой настройки процесса разбиения при помощи параметра алгоритма Кернигана-Лина. Данный факт позволяет проводить серию разбиений с различными параметрами с целью выявления оптимального разбиения сети на подсети.

Список литературы

1. Cisco Annual Internet Report (2018-2023) White Paper. 2020. [Электронный ресурс]. URL: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html (дата обращения: 30.10.2023).

2. Степанова М.В., Ерёмин О.Ю. Назначение заданий узлам распределенной системы платформы Интернета вещей на основе машинного обучения с подкреплением // Автоматизация процессов управления. 2021. №1 (63). С. 27-33. DOI: 10.35752/1991-2927-2021-1-63-27-33.

3. Eremin O., Stepanova M. A Reinforcement Learning Approach for Task Assignment in IoT Distributed Platform // Cyber-Physical Systems. Studies in Systems, Decision and Control. 2021. Vol. 350. P. 385-394. DOI: 10.1007/978-3-030-67892-0_31.

4. Stepanova M., Eremin O., Proletarsky A. Self-regulation Management in IoT Infrastructure Using Machine Learning. // Recent Innovations in Computing. Lecture Notes in Electrical Engineering. 2022. Vol. 832. P. 3-15. DOI: 10.1007/978-981-16-8248-3_1.

5. Berenberg A., Calder B. Deployment Archetypes for Cloud Applications. ACM Computing Surveys (CSUR) // ACM Computing Surveys. 2022. Vol. 55. No. 3. URL: https://dl.acm.org/doi/full/10.1145/3498336 (дата обращения: 30.10.2023). DOI: 10.1145/3498336.

6. Afzal S., Kavitha G. Load balancing in cloud computing – A hierarchical taxonomical classification // Journal of Cloud Computing. 2019. Vol. 8. Is. 1. P. 22. DOI: 10.1186/s13677-019-0146-7.

7. Aghdai A., Wang M. I.-C., Xu Y., Wen C. H.-P., Chao H. J. In-network Congestion-aware Load Balancing at Transport Layer // 2019 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN). 2019. P. 1-6. DOI: 10.1109/NFV-SDN47374.2019.9040109.

8. Де Йонге Д. Nginx. Книга рецептов. Продвинутые рецепты высокопроизводительной балансировки нагрузки. М.: ДМК, 2020. 177 с.

9. Kernighan B.W. An efficient heuristic procedure for partitioning graphs // Bell System Technical Journal. 1970. Vol. 49. P. 291-307.

10. Бадеха И.А., Ролдугин П.В. О плотности графов, в которых каждое ребро входит хотя бы в две максимальные клики // Дискретная математика. 2013. Т. 25. Вып. 3. URL: https://www.mathnet.ru/rus/dm/v25/i3/p7 (дата обращения: 30.10.2023). DOI: 10.4213/dm1244.