

Алгоритмы размещения файлов в компьютерных сетях

Д.В. Бельков

Донецкий национальный технический университет
belkov65@list.ru

Бельков Д.В. Алгоритмы размещения файлов в компьютерных сетях. В работе предлагаются новые алгоритмы для решения актуальной научной задачи оптимизации размещения файлов в компьютерных сетях. Они могут быть использованы для повышения эффективности компьютерной сети на этапах ее проектирования и эксплуатации. Критерием оптимальности размещения файлов является суммарный поток локальных запросов, инициированных в узлах сети в единицу времени. С увеличением потока локальных запросов эффективность функционирования сети увеличивается за счет уменьшения времени отклика на запросы к файлам. В работе приведены результаты вычислительного эксперимента.

Ключевые слова: Файлы, узлы компьютерной сети, методы размещения файлов.

Введение

В связи с интенсивным развитием средств вычислительной техники широкое распространение получили компьютерные сети с распределенной обработкой информации. Большой вклад в решение задач повышения эффективности компьютерных сетей внесли В.М. Глушков, А.А. Стогний, Е.Л. Ющенко, В.Н. Редько. Однако в связи с массовым распространением распределенных систем, работы в этом направлении не утрачивают актуальности и требуют дальнейшего развития. Одним из способов повышения эффективности функционирования компьютерных сетей является оптимизация размещения файлов. Поэтому задача оптимального размещения файлов по узлам сети имеет важное практическое значение. Эта задача относится к классу NP - трудных. Точные методы можно применять только для решения задач малой размерности. В случае большой размерности необходимо использовать приближенные методы. Поэтому актуальной является разработка новых методов размещения файлов [1-6]. Целью данной работы является повышение эффективности функционирования компьютерной сети за счет оптимизации размещения файлов. Задачи работы:

1. Разработать алгоритмы размещения файлов в компьютерной сети, которые позволяют уменьшить время решения этой задачи за счет исключения полного перебора вариантов решений.
2. Исследовать точность решений задачи распределения файлов, разработанными алгоритмами.

Анализ работ по размещению файлов в компьютерных сетях свидетельствует, что для

обеспечения эффективного функционирования распределенной системы необходимо рациональное размещение файлов по узлам компьютерной сети в зависимости от интенсивности запросов к ним. Наиболее рационально размещать файл в тот узел, который чаще других выдает запрос на его использование. Это позволяет существенно повысить эффективность функционирования сети за счет уменьшения нагрузки на каналы связи. Для статического размещения файлов по узлам компьютерной сети необходимо при фиксированных значениях интенсивностей запросов к файлам так распределить файлы по узлам компьютерной сети, чтобы время отклика сети было минимальным.

Задача размещения файлов

В данной работе для уменьшения времени отклика оптимизируется интенсивность информационного трафика. При функционировании сети в каждом узле образуются два типа запросов: сетевой запрос, для обработки которого необходим файл, не содержащийся в том узле, где возник запрос и локальный запрос, для обработки которого необходим файл, содержащийся в том узле, где возник запрос. Критерием оптимальности размещения файлов является суммарный поток локальных запросов, инициированных в узлах в единицу времени. Чем больше суммарный поток локальных запросов, тем меньше время отклика сети.

Обозначим: F_{ij} - количество запросов к файлу i из узла j в единицу времени; $X_{ij} = 1$, если файл i расположен в узле j , иначе $X_{ij} = 0$;

V_i - объем файла i ; B_j - объем узла j , $i=1...m$, $j=1...n$.

Для рационального использования памяти узлов целесообразно минимизировать объем их свободной памяти. Поэтому коэффициент заполнения узлов, равный отношению V_i / B_j необходимо максимизировать. Задача размещения файлов по узлам компьютерной сети имеет вид:

Целевая функция

$$L = \sum_{i=1}^m \sum_{j=1}^n F_{ij} V_i X_{ij} / B_j = \sum_{i=1}^m \sum_{j=1}^n L_{ij} X_{ij} \rightarrow \max \quad (1)$$

Ограничения:

$$X_{ij} \in \{0,1\}, \sum_{j=1}^n X_{ij} = 1, i=1...m \quad (2)$$

$$\sum_{i=1}^m V_i X_{ij} \leq B_j \quad (3)$$

В задаче (1)-(3) необходимо найти матрицу размещений файлов X . В задаче максимизируется суммарный поток локальных запросов [6]. Увеличение потока локальных запросов связано с увеличением эффективности функционирования сети следующим образом.

Пусть $\sum_{i=1}^m \sum_{j=1}^n F_{ij}$ - суммарная интенсивность всех запросов, $\sum_{i=1}^m \sum_{j=1}^n F_{ij} X_{ij}$ - суммарная интенсивность

локальных запросов, $(\sum_{i=1}^m \sum_{j=1}^n F_{ij} - \sum_{i=1}^m \sum_{j=1}^n F_{ij} X_{ij})$ -

суммарная интенсивность сетевых запросов, R_0 - среднее время выполнения сетевого запроса, X_0 - максимальная пропускная способность сети, R - среднее время отклика сети. Среднее время отклика на один запрос совпадает со средним временем ожидания обслуживания сетевого запроса. Оно состоит из среднего времени доступа к каналу связи и среднего времени выполнения запроса. Значение R определяется по формуле

$$R = R_0 \cdot (\sum_{i=1}^m \sum_{j=1}^n F_{ij} - \sum_{i=1}^m \sum_{j=1}^n F_{ij} X_{ij}) / X_0. \quad \text{Критерий}$$

$\sum_{i=1}^m \sum_{j=1}^n F_{ij} X_{ij} V_i / B_j \rightarrow \max$ задачи размещения файлов обеспечивает максимизацию значений $\sum_{i=1}^m \sum_{j=1}^n F_{ij} X_{ij}$ и, следовательно, минимизацию значения R .

Таким образом, максимизация интенсивности локальных запросов к файлам приводит к повышению эффективности функционирования сети за счет уменьшения среднего времени отклика на запросы к файлам.

Известно [7], что интенсивность запросов к файлам в Internet меняется в пределах от 0 до 1000 запросов в секунду и подчиняется вероятностному распределению Парето

$$F(x) = P(X \leq x) = \begin{cases} 0, & x < k \\ 1 - (k/x)^\alpha, & x \geq k, \alpha < 2 \end{cases}, \quad \text{где}$$

$P(X \leq x)$ - вероятность того, что значение случайной величины X не превысит заданное число x . Параметр $\alpha = \alpha_1 = 1,5$. Большинство файлов находится в диапазоне 100–100000 байт. Распределение объемов файлов подчиняется вероятностному распределению Парето с параметром $\alpha_2 = 1,1$.

Вычислительные эксперименты по размещению файлов среди узлов сети проведены для случая $E_1 = 500$ запросов в секунду, $k_1 = (\alpha_1 - 1) \cdot E_1 / \alpha_1 = 166,67$ запросов в секунду. Интенсивность запросов к i -му файлу из j -го узла вычислялась по формуле: $F_{ij} = k_1 / (1 - \gamma)^{1/\alpha_1}$ (запросов в секунду), где γ - случайная величина, равномерно распределенная на интервале $(0;1)$, $E_2 = 50000$ байт, $k_2 = (\alpha_2 - 1) \cdot E_2 / \alpha_2 = 4545,46$ (байт). Объем i -го файла вычислялся по формуле: $V_i = k_2 / (1 - \gamma)^{1/\alpha_2}$ (байт), $i=1,2,\dots,8$; $j=1,2,3$. В ходе вычислительных экспериментов предлагаемыми в работе методами и методом полного перебора решается задача распределения m файлов среди n узлов сети, $m=8$, $n=3$. Для полного перебора требуется $n^m = 3^8 = 6561$ итерация, предлагаемые методы выполняют 1000 итераций. На каждой итерации формируется матрица размещения файлов, и вычисляется целевая функция (ЦФ) по формуле (1). Программы разработаны в среде Delphi.

Алгоритм имитации отжига

В данном разделе для решения задачи размещения файлов предлагается алгоритм „Машина Коши” (МК), относящийся к классу методов имитации отжига. Алгоритм работает следующим образом. „Температура” T изменяется по формуле $T := T_0 / \tau^\alpha$, $T_0=1000$, $\alpha = 1,5$, τ - номер итерации. Вероятность выбора узла j для размещения i -го файла вычисляется по формуле $P_j := C_j / \sum_{j=1}^n C_j$, где

$$C_j = \frac{1}{1 + \exp\{-F_{ij} / (r \cdot T)\}}, \quad r - \text{максимальное}$$

текущее значение ЦФ на очередной итерации. Первоначально $r=1000$. Узел выбирается по правилу рулетки. Если найденное решение не

удовлетворяет условию (3) задачи, то оно отвергается. Результаты вычислительного эксперимента показаны на рис. 1. Получено строго оптимальное решение на 984 итерации.

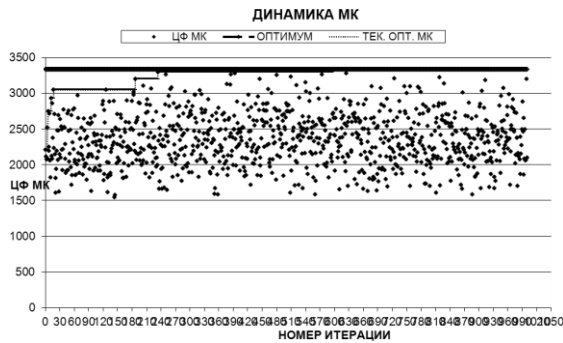


Рисунок 1 – Динамика поиска решений методом имитации отжига

Муравьиный алгоритм

В данном разделе для решения задачи размещения файлов предлагается алгоритм, относящийся к классу алгоритмов муравьиной колонии. Этот класс алгоритмов появился в результате исследований поведения живых муравьев. Муравей, двигаясь по определенному маршруту, оставляет за собой след пахнущего вещества (феромона). Такое вещество влияет на выбор маршрута: выбирается то направление движения, где уровень феромона больше. Самоорганизация муравьев обеспечивается взаимодействием следующих компонентов: случайность, многократность, положительная обратная связь, отрицательная обратная связь. Центральной идеей муравьиного алгоритма является накопление и использование статистических данных, собираемых искусственными муравьями [8].

Для решения задачи (1)-(3) в данной работе предлагается следующий алгоритм:

- Шаг 1. Присвоить переменной record первоначальное (небольшое) значение;
- Шаг 2. Присвоить переменной t значение 1;
- Шаг 3. While $t \leq t_{\max}$ Do

Begin

Шаг 4. For i:=1 to m do

Begin

Шаг 5. Сформировать вектор P(n) вероятностей размещения i-го файла в j-й узел.

Шаг 6. Сгенерировать случайную величину g, распределенную по закону P(n);

Шаг 7. Если условие $j \leq g < j + 1$ и условие (3) задачи выполняются, то назначить i-й файл в j-й узел, иначе сгенерировать новое значение случайной величины g.

End;

Шаг 8. Вычислить значение целевой функции по формуле (1) и присвоить его переменной c_f;

Шаг 9. If c_f > record then

Begin

Шаг 10. Присвоить переменной record значение переменной c_f и сохранить рекордное решение; End;

Шаг 11. Присвоить переменной t значение t+1; End;

Шаг 12. Вывести наилучшее решение и завершить алгоритм.

Пусть τ_j - количество феромона, накопленное очередным муравьем на маршруте j, ρ - коэффициент испарения феромона, $\rho = 0,01$, τ_0 - первоначальный уровень феромона, $\tau_0 = 0,01$, τ_j - уровень феромона на маршруте j. Вероятность размещения файла в узел j вычисляется по формуле (4):

$$P_j = \frac{\tau_j F_j}{\sum_{j=1}^n \tau_j F_j} \quad (4)$$

При выборе узла p для размещения файла происходит увеличение значения τ_p :

$\tau_p := (1 - \rho)\tau_p + \tau_p / ideal$, значение переменной ideal должно быть одного порядка с оптимальным значением целевой функции. При переходе к очередной итерации алгоритма уровень феромона накапливается:

$$\tau_j := (1 - \rho)\tau_j + \sum_{k=1}^n \tau_k / ideal.$$

Результаты эксперимента показаны на рис. 2. Получено строго оптимальное решение на 311 итерации.

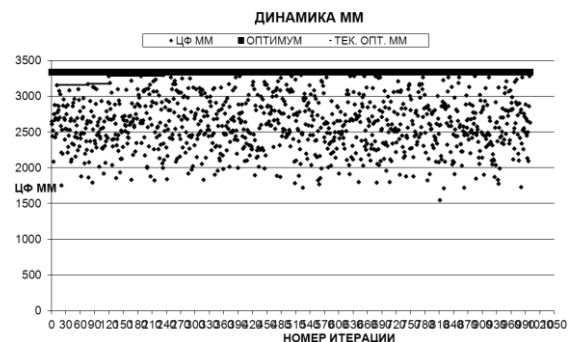


Рисунок 2 – Динамика поиска решений муравьиным методом

Генетический алгоритм

В данном разделе для решения задачи размещения файлов предлагается алгоритм, относящийся к классу генетических алгоритмов (ГА). Это стохастические эвристические оптимизационные методы. Они работают с совокупностью "особей" - популяцией, каждая из которых представляет возможное решение данной проблемы [9,10].

Для решения задачи (1)-(3) предлагается генетический алгоритм ГА, который выполняет поиск оптимального решения задачи с помощью постоянной эволюции предыдущего решения. При переходе от решения к решению применяются генетические операторы селекции, кроссингвера и мутации. В данной работе используется мультихромосомное представление решений задачи. Очередным решением является особь, мультихромосома которой показана в табл. 1. Она состоит из m хромосом и mn генов, где m – число файлов, n – число узлов, $X_{ij} \in \{0;1\}$

Таблица 1 - Мультихромосома особи

X11	X12	X13	X14	X15	...	X1n
X21	X22	X23	X24	X25	...	X2n
...
Xm1	Xm2	Xm3	Xm4	Xm5	...	Xmn

В ГА используются генетические операторы кроссингвера и мутации. Оператор кроссингвера производит обмен хромосомами между особями. Оператор мутации позволяет дать хромосомам-потомкам свойства, отсутствующие у родителей. Это позволяет алгоритму выходить из зоны локального экстремума. Выполнение кроссингвера показано в табл. 2а, 2б. Выполнение мутации показано в табл. 3.

Таблица 2а - Родительские особи (S и D)

S11	S12	S13	S14	S15	...	S1n
S21	S22	S23	S24	S25	...	S2n
S31	S32	S33	S34	S35	...	S3n
S41	S42	S43	S44	S45	...	S4n
...
Sm1	Sm2	Sm3	Sm4	Sm5	...	Smn
D11	D12	D13	D14	D15	...	D1n
D21	D22	D23	D24	D25	...	D2n
D31	D32	D33	D34	D35	...	D3n
D41	D42	D43	D44	D45	...	D4n
...
Dm1	Dm2	Dm3	Dm4	Dm5	...	Dmn

Таблица 2б – Результат кроссингвера

S11	S12	S13	S14	S15	...	S1n
D21	D22	D23	D24	D25	...	D2n
S31	S32	S33	S34	S35	...	S3n
S41	S42	S43	S44	S45	...	S4n
...
Sm1	Sm2	Sm3	Sm4	Sm5	...	Smn
D11	D12	D13	D14	D15	...	D1n
S21	S22	S23	S24	S25	...	S2n
D31	D32	D33	D34	D35	...	D3n
D41	D42	D43	D44	D45	...	D4n
...
Dm1	Dm2	Dm3	Dm4	Dm5	...	Dmn

Таблица 3 – Результат мутации

S11	S12	S13	S14	S15	...	S1n
S21	S22	S23	S24	S25	...	S2n
S41	S42	S43	S44	S45	...	S4n
S31	S32	S33	S34	S35	...	S3n
...
Sm1	Sm2	Sm3	Sm4	Sm5	...	Smn

В предлагаемом ГА оператор селекции имеет три разновидности: оператор селекции генов используется при формировании генов хромосом; оператор селекции хромосом используется при выборе хромосом для выполнения кроссингвера и мутации; оператор селекции особей используется при выборе родительских особей из популяции.

При формировании генов i -й хромосомы используется оператор селекции генов. Вероятность π_{ij} выбора j -го гена i -й хромосомы вычисляется, как отношение генной и хромосомной приспособленности по формуле (5):

$$\pi_{ij} = \frac{L_{ij}Z_{ij}}{\sum_{q=1}^n L_{iq}Z_{iq}} \quad (5)$$

При формировании каждого гена i -й хромосомы разыгрывается случайная величина, подчиняющаяся распределению π . Для этого используется следующая процедура:

- Шаг 1. $\Psi_1 := \pi_1$
- Шаг 2. For $q:=1$ to $n-1$ do
- Шаг 3. Begin $\Psi_{q+1} := \pi_q + \pi_{q+1}$ End
- Шаг 4. $\gamma := random$
- Шаг 5. For $q:=1$ to $n-1$ do
- Шаг 6. Begin If $\Psi_q \leq \gamma < \Psi_{q+1}$ then $X_{iq} := 1$ else $X_{iq} := 0$
- Шаг 7. End

При выборе хромосом для выполнения кроссингвера и мутации используется оператор селекции хромосом. Вероятность Π_i выбора i -й хромосомы вычисляется, как отношение хромосомной и мультихромосомной приспособленности по формуле (6):

$$\Pi_i = \frac{\sum_{j=1}^n L_{ij}X_{ij}}{\sum_{r=1}^m \sum_{j=1}^n L_{rj}X_{rj}} \quad (6)$$

Пусть $Y_r = 1$, если выбрана хромосома r , иначе $Y_r = 0$. При формировании значений Y_r генерируются случайные величины, подчиняющиеся распределению Π . Для этого

используется процедура, аналогичная процедуре оператора селекции генов.

При выборе из популяции родительских особей используется оператор селекции особей. Вероятность P_k выбора особи на k -й итерации вычисляется, как отношение индивидуальной и наилучшей приспособленности. Пусть CF_k - значение целевой функции на k -й итерации, RCF - максимальное значение среди CF_s , где $s=1,2,\dots,k$. На каждой итерации целевая функция вычисляется по формуле: $CF = \sum_{i=1}^m \sum_{j=1}^n L_{ij} X_{ij}$.

Вероятность P_k вычисляется по формуле (7). Лучше приспособленные особи имеют большую вероятность стать родительскими.

$$P_k = CF_k / RCF_k \quad (7)$$

При выборе особи генерируется случайная величина, подчиняющаяся распределению P_k . Для этого используется такая же процедура, как и в операторе селекции генов.

Реализация ГА предусматривает выполнение следующих действий:

{Подготовительная часть}

{Формирование первоначальной особи}

Шаг 1. $k:=1$;

Шаг 2. Случайным образом сгенерировать мультихромосому особи $MH(k)$;

Шаг 3. Вычислить значение CF_k целевой функции особи $MH(k)$;

Шаг 4. $RCF := CF_k$;

{Основная часть}

{Цикл формирования популяции}

Шаг 5. While $k \leq k_{\max}$ Do

Begin

Шаг 6. Выполнить мутацию особи $MH(k)$ и получить особь $MH(k+1)$;

Шаг 7. Вычислить значение CF_{k+1} целевой функции особи $MH(k+1)$;

Шаг 8. If $CF_{k+1} > RCF$ then $RCF := CF_{k+1}$;

Шаг 9. $k:=k+1$;

Шаг 10. Выполнить кроссинговер особей $MH(k-1)$, $MH(k)$ и получить особи $MH(k+1)$ и $MH(k+2)$;

Шаг 11. Если условие (3) задачи выполняется, выполнить мутацию особей $MH(k+1)$ и $MH(k+2)$;

Шаг 12. Вычислить значение CF_{k+1} целевой функции особи $MH(k+1)$;

Шаг 13. If $CF_{k+1} > RCF$ then $RCF := CF_{k+1}$;

Шаг 14. Вычислить значение CF_{k+2} целевой функции особи $MH(k+2)$;

Шаг 15. If $CF_{k+2} > RCF$ then $RCF := CF_{k+2}$;

Шаг 16. $k:=k+2$;

Шаг 17. Выполнить оператор репродукции;

End;

Шаг 18. Вывести значения генов мультихромосомы с максимальной целевой функцией и завершить ГА.

Результаты эксперимента показаны на рис. 3. Получено строго оптимальное решение на 58 итерации.

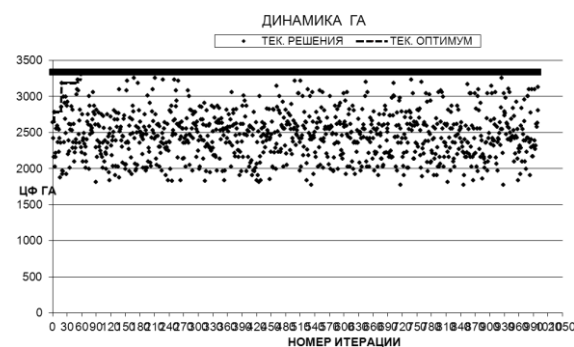


Рисунок 3 – Динамика поиска решений генетическим алгоритмом

Выводы

В работе предложены алгоритмы имитации отжига, муравьиный и генетический алгоритмы решения задачи размещения файлов в компьютерных сетях. Динамика работы алгоритмов показана на рис. 4.



Рисунок 4 – Динамика поиска решений

Предложенные алгоритмы могут быть использованы для повышения эффективности компьютерной сети на этапах ее проектирования и эксплуатации. Критерием оптимальности размещения файлов является суммарный поток локальных запросов, инициированных в узлах сети в единицу времени. В ходе вычислительных экспериментов предлагаемыми в работе алгоритмами и методом полного перебора решена задача распределения m файлов среди n узлов сети, $m=8$, $n=3$. Для полного перебора требуется $n^m = 3^8 = 6561$ итерация, предложенными алгоритмами выполнено 1000

итераций. Получены следующие результаты. Строго оптимальное решение найдено алгоритмом имитации отжига на 984 итерации, муравьиным алгоритмом на 311 итерации, генетическим алгоритмом на 58 итерации.

Литература

1. Upadhyaya S., Lata S. Task allocation in Distributed computing VS distributed database systems: A Comparative study. // International Journal of Computer Science and Network Security, Vol. 8, N 3, 2008. – p. 338 – 346.
2. Singh A., Dutta K. Apply AHP for Resource Allocation Problem in Cloud. // International Journal of Computer Science and Network Security, Vol. 8, N 3, 2015. – p. 13 – 21.
3. Abdalla H.I. A New Data Re-Allocation Model for Distributed Database Systems. // International Journal of Database Theory and Application, Vol. 5, N 2, 2012. – p. 45 – 60.
4. Javed S., Bakhshi Z. H., Khalid M. M. Optimum allocation in Stratified Sampling with Random Costs. // International Review of Pure and Applied Mathematics, Vol. 5, N 2, 2009. – p. 363-370.
5. Khan M. F., Ali I., Ahmad Q.S. Chebyshev Approximate Solution to Allocation Problem in Multiple Objective Surveys with Random Costs. // American Journal of Computational Mathematics, N 1, 2011. – p. 247-251.
6. Kaur M., Singh S., Kaur R. Directory Structure and File Allocation Methods. // International Journal of Computer Science and Information Technologies, Vol. 7, N 2, 2016. – p. 577–582.
7. Зинкин С.А., Белецкий П.А. Оптимизация размещения данных по узлам информационно-вычислительной сети. В кн.: Современные тенденции технических наук: Материалы международной заочной научной конференции. – Уфа: Лето, 2013. – с. 29–31.
8. Курейчик В.В., Запорожец Д.Ю. Роевой алгоритм в задачах оптимизации. // Известия Южного федерального университета. Технические науки, Т. 108, № 7, 2010. – С. 28-32.
9. Abd El-Aziz M. H. Optimizing the Average Distributed Program Throughput (ADPT) by Using Genetic Algorithms. // International Journal of Intelligent Computing and Information Science, Vol. 10, N 1, 2010. –p.1-12.
10. Abd El-Aziz M. H., Younes A., Hassan M. R., Abdo H. Solving the file allocation problem in the distributed networks by using genetic algorithms. // International Journal of Computer Science and Network Security, Vol. 2, N 1, 2013. – p. 109 – 117.

Бельков Д.В. Алгоритмы размещения файлов в компьютерных сетях. В работе предлагаются новые методы для решения актуальной научной задачи оптимизации размещения файлов в компьютерных сетях (метод имитации отжига, муравьиный и генетический методы). Они могут быть использованы для повышения эффективности компьютерной сети на этапах ее проектирования и эксплуатации. Критерием оптимальности размещения файлов является суммарный поток локальных запросов, инициированных в узлах сети в единицу времени. С увеличением потока локальных запросов эффективность функционирования сети увеличивается за счет уменьшения времени отклика на запросы к файлам. В работе приведены результаты вычислительного эксперимента.

Ключевые слова: Файлы, узлы компьютерной сети, методы размещения файлов.

Belkov D.V. File allocation algorithms in computer networks. New algorithms for the decision of actual scientific task of optimization of file allocation in computer networks are offered in work. They can be used for the increase of efficiency of computer network on the stages of its design and exploitation. The total thread of the local queries initiated in the nodes of network in time unit is the criterion of optimum of file allocation. With the increase of thread of local queries is multiplied efficiency of functioning of network due to diminishment of response time upon requests to the files. The results of calculable experiment are presenting in work.

Keywords: Files, nodes of computer network, file allocation algorithms.

Статья поступила в редакцию 7.09.2017

Рекомендована к публикации доктором технических наук В.Н. Павлышом