

сборе и извлечении конфиденциальной информации о роботизированных системах и их текущих операторах.

Атака «человек посередине» с перехватом данных при обмене информацией между РТС или узлами, изменять информацию и внедрять.

Атаки с помощью скомпрометированных роботов, размещенных в сети для генерации ложных данных.

## **5. Противодействие**

При проектировании роботов производители должны учитывать безопасность в качестве ключевого компонента при разработке любой прошивки, операционной системы, оборудования и приложений. Использовать постоянный мониторинг уязвимостей для обеспечения безопасности. Мониторинг активности позволит отследить аномалии поведения робототехнических систем. Внедрение улучшенных способов управления доступом, для предотвращения атак, направленных на перехват управления. Внедрение усовершенствованных криптографических решений и протоколов для защиты данных и каналов коммуникации, также внедрение безопасных каналов связи на физическом уровне.

Внедрение механизмов мгновенного отключения робота при обнаружении угрозы безопасности, в том числе через реализацию закладки для саморазрушения управляющего контроллера или процессора.

---

**Сомов С.К.**

### **Способы сокращения вычислительной сложности алгоритмов поиска оптимального размещения массивов данных в распределенных системах обработки данных**

**Аннотация:** В работе приведены несколько утверждений теорем, использование которых при решении задач поиска оптимального размещения копий массивов данных в распределенных системах позволяет сократить вычислительную сложность данных задач. Представлено несколько методов, которые способствуют уменьшению

вычислительной сложности задач размещения копий массивов данных или их реплик в распределенных системах обработки данных различного назначения.

**Ключевые слова:** распределенные системы обработки данных, задача размещения копий массивов данных в узлах распределенных систем, вычислительная сложность задач размещения данных

Для решения задачи увеличения производительности и обеспечения высокой надежности функционирования распределенных систем обработки данных используются различные подходы. В первую очередь, это создание таких систем на базе быстродействующего и надежного оборудования, системного и прикладного программного обеспечения. Другим часто применяемым в распределенных системах обработки данных (РСОД) методом является использование информационной избыточности в виде некоторого количества копий массивов данных или реплик массивов, которые размещаются в нескольких узлах распределенной системы [1]. Такое размещение, во-первых, позволяет адресовать запрос к данным в ближайший узел с требуемым массивом данных, что ускоряет получение ответа на запрос и, соответственно, увеличивает производительность всей системы. Во-вторых, если запрос не был успешно обработан в ближайшем узле, он может быть переадресован к другому работоспособному узлу системы. Эта возможность повышает надежность работы РСОД.

Чтобы получить существенное улучшение надежностных и временных характеристик системы путем размещения в узлах системы нескольких копий массивов или реплик необходимо разместить их по узлам системы оптимальным образом.

Множество узлов, в которых размещаются копии массивов или реплики массивов данных, определяются при использовании алгоритма решения задачи поиска оптимального распределения массивов или реплик данных. Данный класс задач обладает большой вычислительной сложностью, особенно для крупномасштабных распределенных систем с большим количеством узлов и сложной топологией. Поэтому, как правило, для решения таких задач используются эвристические алгоритмы,

обеспечивающие решение задачи, близкое к оптимальному. Для этого формулируются и решаются задачи поиска оптимального размещения копий массивов или их реплик. Задачи такого класса имеют большую вычислительную сложностью, особенно для крупномасштабных систем с большим количеством узлов и сложной топологией. Для решения таких задач, как правило, используются эвристические алгоритмы [2].

Для иллюстрации вычислительной сложности данного класса задач размещения рассмотрим следующий пример. Допустим данную задачу необходимо решить для РСОД, состоящей из  $K = 10$  узлов. В РСОД хранится и используется для обработки поступающих запросов  $M = 6$  разных массивов данных. В некотором подмножестве узлов системы требуется разместить по несколько копий различных массивов данных. Сделаем допущение о том, что в каждом  $i$ -м узле системы можно разместить от 1-го до 3-х наборов копий разных массивов. Тогда для каждого из 6 узлов сети возможны следующие варианты размещения наборов копий массивов:

- $C_6^1$  – один набор для одного из 6 разных массивов данных,
- $C_6^2$  – два набора копий двух из 6 различных массивов данных,
- $C_6^3$  – три набора копий двух из 6 различных массивов данных.

Тогда получим, что для каждого одного из 6-ти узлов системы число возможных вариантов размещения наборов копий из 6 разных массивов данных, равно:

$$W' = (C_6^1 + C_6^2 + C_6^3) = (6 + 15 + 20) = 41. \quad (1)$$

Следовательно, при использовании метода полного перебора для решения задачи поиска оптимального размещения наборов копий массивов данных для рассматриваемой системы, количество разных вариантов размещения, которые необходимо рассмотреть, будет равно:

$$W = (W')^{10} = (41)^{10} = 13\,422\,659\,310\,152\,401 \approx 13 * 10^{15} \quad (2)$$

$W = 10^{15} = 13\,422\,659\,310\,152\,401$  – это огромное число, более 13-ти квадриллионов вариантов решения задачи.

В данной работе представлено несколько методов сокращения вычислительной сложности задач размещения, которые можно использовать до начала работы алгоритмов поиска решения задач. Данные методы позволяют исключить некоторые узлы системы из области решения задач.

В общем виде задача поиска размещения резерва копий массива данных или его реплик в узлах распределенной системы обработки данных, оптимальной по величине затрат на функционирование системы, формулируется следующим образом:

$$F(I) \rightarrow \min \quad (3)$$

В задаче используются следующие ограничения:

$$\rho_j \geq \bar{\rho}; E_j \geq \bar{E}; x_j \geq \bar{x}_j; \quad j = \overline{1, N} \quad (4)$$

В качестве аргумента в функционале  $F(I)$  задачи используется переменная  $I$ , которая является массивом, содержащим в качестве своих элементов номера узлов системы, в которых размещены экземпляры копий массива данных. Ниже приведена формула расчета функционала задачи:

$$S(I) = \sum_{j=1}^N s_j x_j + \sum_{j=1}^N \left[ \sum_{i \in I} V_j ZP_{ji}(x_i) + U_i \min_{i \in N_j \cap I} ZP_{ji}(x_i) \right] \quad (5)$$

В формуле расчета функционала задачи применяются обозначения:

$ZP_{ji}(x_i) = 2d_{ji} + E_i(x_i)h_i$  – стоимость (затраты) обработки одного запроса, который сгенерирован в узле  $j$ , а для обработки и получения ответа на него адресован узел  $i$  системы;

$E_i(x_i)$  – это среднее время использования ресурсов узла  $i$ , в процессе обработки одного запроса. При этом в данном узле размещено  $x_i$  копий массива;

$d_{ji}$  – стоимость использования ресурсов каналов связи системы для передачи из узла  $j$  в узел  $i$  одного запроса или ответа на запрос;

$I$  – множество номеров узлов системы, в которых размещены копиями массивов данных;

$s_j$  – стоимость хранения одной копии массива данных в узле  $j$ .

Информационный запрос, возникший в одном из узлов системы, пересылается для обработки в ближайший узел системы, в котором размещена копия необходимого массива данных. Возникший в системе запрос на модификацию данных адресуется из узла-источника запроса во все узлы системы, в которых размещены копии модифицируемого массива данных.

Обработка запросов в системе характеризуется следующими параметрами:

- $\rho_j$  – вероятность успешной обработки запроса, сгенерированного узле  $j$  системы, и получения на него ответа;
- $E_j$  – среднее время, затраченное на получение ответа на запрос, сгенерированный в узле  $j$ . Значения этих параметров:

$$\rho_j = \sum_{i=1}^N \rho_{ji}(x_i)y_{ji}; \quad E_j = \sum_{i=1}^N E_{ji}(x_i)y_{ji}. \quad (6)$$

Булева переменная  $y_{ji}$  ( $y_{ji} \in \{0,1\}$ ), используемая в формулах, имеет значение  $y_{ji} = 1$ , в случае, если запрос из узла  $j$  отправляется в узел  $i$  для последующей его обработки, адресуется для обработки в узел  $i$ . И переменная равна 0 в противном случае.

В приведенных выше формулах используются параметры:

$\rho_{ji}(x_i)$  – значение вероятности успешной обработки в узле  $i$  с  $x_i$  копиями массива данных, запроса к данным, отправленного из узла  $j$ ;

$E_{ji}(x_i)$  – среднее время получения ответа на запрос из узла  $j$  к данным копии массива, размещенным в узле  $i$ .

Значения данных параметров рассчитываются по формулам:

$$\rho_{ji}(x_i) = r_{ji}P_i(x_i)r_{ij} \quad (7)$$

$$E_{ji}(x_i) = \begin{cases} 2t_3 + E_i(x_i) & \text{при } j \neq i \\ E_i(x_i) & \text{при } j = i \end{cases} \quad (8)$$

В данных формулах:

$P_i(x_i)$  – вероятность успешной обработки запроса в узле  $i$  системы;

$t_3$  – среднее время задержки передачи сообщения (запроса или ответа на него) по каналам связи в системе.

Для каждого узла  $i$  ( $i = \overline{1, N}$ ) системы сформируем множество  $M_i$ , которое содержит множество таких узлов  $j$  системы, что если в таком узле генерируется запрос к данным, то он адресуется в узел  $i$  для последующей обработки и получения ответа:

$$M_i = \{j / \rho_{ji}(x_{ji}) \geq \bar{\rho}; E_{ji}(x_{ji}) \leq \bar{E}; x_{ji} \leq \bar{x}_i\}. \quad (9)$$

Дополнительно для каждого узла  $i$  ( $i = \overline{1, N}$ ) системы сформируем вектор  $X_i = (x_{ji}), j = \overline{1, N}$ . Каждый элемент  $x_{ji}$  данного вектора это такое минимальное количество копий массива данных, размещаемых в узле  $i$ . Эти копии массива будут использоваться для обработки запросов, поступающих из узла  $j$ . При этом должны учитываться ограничения (4).

Значения элементов  $x_{ji}$  вектора  $X_i$  рассчитываются на основе формул, приведенных в [2-3]. Например, количество копий массива данных в узле  $i$  будет равно:

$$x_{ji} = \left\lfloor \ln \left\{ 1 - \bar{\rho} (r_{ji} r_{ij})^{-1} \right\} (\ln q_i)^{-1} \right\rfloor + 1. \quad (10)$$

Если одно или оба ограничения, приведенные ниже, не выполняются:

$$\rho_{ji}(x_{ji}) \geq \bar{\rho}; E_{ji}(x_{ji}) \leq \bar{E} \text{ или } x_{ji} \leq \bar{x}_i, \quad (11)$$

то минимальное количество копий массива данных  $x_{ji}$ , размещаемых в узле  $i$  равно нулю:  $x_{ji} = 0$ .

Определение (9) множества  $M_i$  позволяет утверждать справедливость следующего утверждения:

*Утверждение 1.* Если множество  $M_i$  пусто, т.е.  $M_i = \emptyset$ , то в соответствующем узле  $i$  не требуется размещение копий массивов данных, поскольку в этот узел системы не будут направляться запросы.

Для всех узлов  $j = \overline{1, N}$  распределенной системы на основе множества  $M_i$ , сформированного ранее, получим соответствующее количество множеств  $N_j = \{i / j \in M_i\}$ , каждое из которых содержит перечень номеров узлов системы, в которые можно пересылать для обработки запросы, сгенерированные в узле  $j$ . На рисунке 1 показан пример этих множеств.

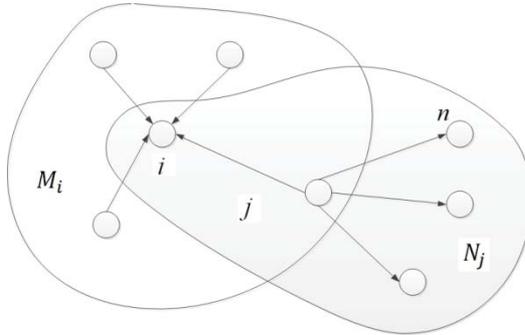


Рисунок 1 – Пример множества  $M_i$  и множества  $N_j$

Решение сформулированной ранее задачи (3) – (4) ищется на следующем множестве  $\hat{U}$ :

$$\hat{U} = \bigcup_{j=1}^N N_j. \quad (12)$$

Тогда, можно утверждать справедливость утверждений 2 и 3.

*Утверждение 2.* Если  $\hat{U} = \emptyset$ , то задача (3) – (4) поиска размещения копий массивов данных не имеет решения, так как нарушаются ограничения (4).

*Утверждение 3.* Если некоторый узел  $j$  системы, в котором генерируются запросы к массивам данных ( $j \in J_3$ ) справедливо равенство;  $N_j = \{i\}$ , то в этом узле  $j$  размещаются копии массивов данных, используемые для обработки запросов, генерируемых только в данном узле системы.

Из Утверждения 3 очевидно получается следствие 1.

*Следствие 1.* Если для всех узлов  $j$  ( $j \in J_3$ ) выполняется равенство  $N_j = \{i\}$ , тогда из Утверждения 3 следует, что распределенную систему можно рассматривать как множество автономных компьютеров. А задача оптимального резервирования для этих автономных компьютеров решается известными методами [3].

Несколько теорем, которые помогают сократить вычислительную сложность задачи оптимального распределения копий массивов данных в узлах распределенных систем, доказано в работе [3]. Уменьшение вычислительной сложности задачи достигается за счет того, что эти теоремы позволяют сократить количество вариантов адресации запросов для их обработки, т.е. сокращается мощность множеств  $M_i$  и  $N_j$ .

### **Заключение**

В работе представлено несколько способов сокращения вычислительной сложности задач оптимального размещения избыточных данных в узлах распределенных систем обработки данных. Представленные материалы целесообразно использовать при анализе и синтезе крупномасштабных РСОД.

### **Литература:**

1. *Кульба В.В., Сомов С.К., Шелков А.Б.* Анализ влияния использования информационной избыточности на показатели надежности распределенных информационных систем // Надежность. – 2022. – Том 22, № 1. – С. 4-12.
  2. *Сомов С.К.* Сохранность информации в распределенных системах обработки данных. – М.: ИПУ РАН, 2019. – 254 с.
  3. *Кульба В.В., Сомов С.К., Шелков А.Б.* Резервирование данных в сетях ЭВМ. – Казань: Издательство казанского университета, 1987. – 175 с.
-