
ОПТИМИЗАЦИЯ МУЛЬТИПРОЦЕССОРНОЙ СИСТЕМЫ НА ОСНОВЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

О. И. Кутузов, Т. М. Татарникова (Санкт-Петербург)

Межсетевые устройства (МУ), такие как центры коммутации пакетов, концентраторы, маршрутизаторы, шлюзы, серверы, абонентские пункты, число которых в распределенной системе может достигать несколько сотен, выполняются в виде многопроцессорных вычислительных систем (МВС), обеспечивающих необходимую производительность за счет параллельной работы элементов МПС и эффективность системы в результате автоматизации и рационализации распределения нагрузки между ее элементами.

Одно из преимуществ параллельных вычислительных комплексов – их способность адаптироваться к внешним условиям, в частности, за счет динамического перераспределения вычислительной мощности в зависимости от приоритета решаемых задач.

Новая технология построения высокопроизводительных систем, какими являются МУ, предлагаемая в данной работе, основана на модульной технологии построения многопроцессорных устройств из базовых "кирпичиков", что позволит пользователям расширять свои системы и добавлять новые технологии по мере необходимости. Система, построенная на базе такой технологии, состоит из "кирпичиков" различных типов:

- процессорный модуль, содержащий процессоры обработки пакетов (ПОП) с локальной памятью для хранения программ (протоколов) обработки пакетов;
- общее поле памяти (ОПП), разбитое на секции (СОП) для обеспечения параллельного доступа нескольких ПОП к разным СОП;
- модуль базового ввода–вывода, представляющий собой набор процессоров портов (ПП) межсетевого устройства;
- коммутационный модуль, который содержит центральный блок комплексирования всех модулей устройства между собой;
- блок расширения, содержащий слоты горячей замены.

Новые типы модулей могут быть добавлены в систему для построения специализированных конфигураций.

Процессорный модуль состоит из набора процессоров обработки пакетов (ПОП). Каждый ПОП выполняет строго определенные протоколы: ПОП 2 реализует протокол 2-го уровня, ПОП – протокол 3-го уровня и т.д. до 7-го уровня ЭМВОС. Здесь возможна параллельная обработка пакетов на разных ПОП как одного и того же уровня, так и разных уровней. Причем количество ПОП разных уровней не обязательно должно быть одинаковым. Например, если на выполнение процессов в ПОП $(n+1)$ -го уровня необходимо больше времени, чем в ПОП n -го уровня, то для исключения неограниченного роста очереди к ПОП $(n+1)$ в работу включается еще один (или несколько) ПОП $(n+1)$ -го уровня.

Каждый пакет, записанный процессором порта в ОПП, генерирует задание, которое выполняется в ПОП по некоторому алгоритму, определяющему последовательность обработки пакетов процессами, и задается управляющей информацией, содержащейся в их заголовках. Информационная связь процессов, выполняемых в ПОП, представлена на рис. 1.

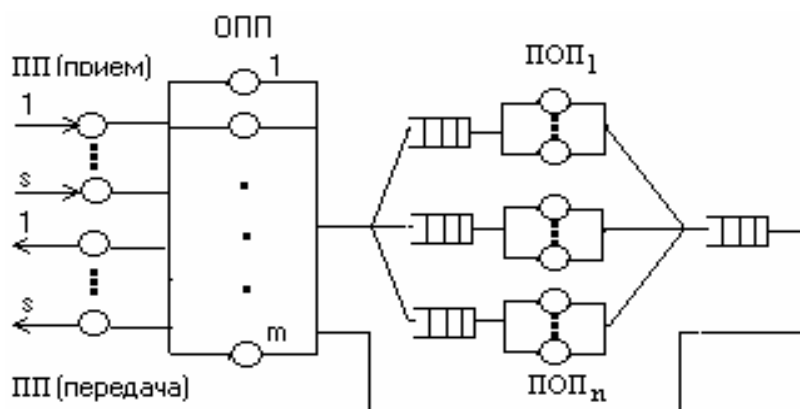


Рис. 1. Информационной связи процессов в межсетевом устройстве

В устройстве в реальном масштабе времени выполняется ряд процессов, реализующих заданные функциональные алгоритмы. Каждый пакет, записанный ПП в ОПП, генерирует задание, которое обрабатывается прикладными процессами по некоторому последовательному алгоритму (технологической цепочке [нити]).

Одновременно, если позволяет число ПОП, может обрабатываться несколько заданий (пакетов). Если задание, приписанное некоторому прикладному процессу, не находит свободного ПОП, то оно ставится в очередь заданий. По освобождении необходимого ПОП, он приписывается первому ожидающему его заданию.

При выполнении процессов может возникнуть необходимость доступа к ресурсам, находящимся в ОПП. Если при запросе некоторого ресурса он оказывается занятым, то процесс ставится в очередь к данному ресурсу и его дальнейшее выполнение блокируется. ПОП, на котором выполнялась обработка пакета, освобождается. При освобождении ресурса он приписывается первому ожидающему его заблокированному процессу, и обработка пакета продолжается с того места алгоритма обработки, где она была прервана.

Будем считать, что суммарное время ожидания предоставления ресурсов и их использования не зависит от порядка обращения к ресурсам.

Назовем подпроцессом фазу процесса, выполняющуюся между двумя последовательными запросами прикладных ресурсов. Таким образом, процесс, имеющий ω запросов к ресурсам, состоит из $\omega+1$ подпроцессов.

Передача задания от одного процесса к другому, распределение активных процессов по ПОП, регулирование доступа процессов к ресурсам осуществляется ОС. Последнее состоит из набора макрокоманд, позволяющих выполнять все необходимые системные операции. Обращение прикладного процесса к любой макрокоманде, инициирует в ПОП соответствующий системный процесс. В свою очередь, при выполнении любого системного процесса осуществляется доступ к системным ресурсам, находящимся в ОПП. Если при запросе некоторого системного ресурса он оказывается занятым, то запрос ставится в очередь к системным ресурсам. При этом ПОП, выполняющий системный процесс, освобождается.

Для определения оптимальной структуры межсетевого устройства предлагается привлечь очень популярные в настоящее время генетические алгоритмы (ГА), основанные на использовании эволюционных принципов для поиска наилучшего решения.

Параметры задачи являются генетическим материалом – генами. Совокупность генов составляет хромосому. Каждая особь обладает своей хромосомой, а следовательно, своим набором параметров. Подставив параметры в целевую функцию, можно по-

лучить какое-то значение. То, насколько это значение удовлетворяет поставленным условиям, определяет характеристику особи, которая называется приспособленностью (fitness). Функция, определяющая приспособленность должна удовлетворять следующему условию: чем "лучше" особь, тем выше приспособленность. ГА работают с популяцией, как правило, фиксированного размера, состоящей из особей, заданных способом, описанным выше. Особи "скрещиваются" между собой с помощью генетических операторов, получается потомство, причем часть потомков заменяют представителей более старого поколения в соответствии со стратегией формирования нового поколения. Выбор особей для скрещивания проводится согласно селективной стратегии (selection strategy). Заново сформированная популяция снова оценивается, затем выбираются наиболее достойные для скрещивания особи, которые также участвуют в процессе эволюции и т.д. Эта процедура продолжается до тех пор, пока не найдется особь, гены которой представляют оптимальный набор параметров, при которых значение целевой функции близко к максимуму или минимуму либо равно ему. Останов работы ГА может произойти также в случае, если популяция вырождается, т.е. практически нет разнообразия в генах особей популяции либо просто вышел лимит времени. Вырождение популяции называют преждевременной сходимостью (premature convergence).

Может создаться впечатление, что ГА являются просто модификацией случайного поиска. Но приспособленность была введена совсем не зря. Дело в том, что она непосредственно влияет на шанс особи принять участие в селекции. Выбирая каждый раз для скрещивания наиболее приспособленных особей, можно с определенной степенью уверенности утверждать, что потомки будут либо ненамного хуже, чем родители, либо лучше их. Приблизительно эту величину уверенности можно оценить с помощью теоремы шаблонов (теоремы шим).

В соответствии с концептуальной моделью МУ к множеству оптимизируемых параметров относятся:

1. Число процессоров обработки пакетов N ;
2. Число секций общего поля памяти M ;
3. Матрица распределения ресурсов по секциям памяти $B^P = [B_{mr}^P]_{m=1, \dots, M; r=1, \dots, R}$,

где R – число общих ресурсов.

Значение B_{mr}^P есть число элементов r -го ресурса, распределенных в m -ю СОП.

Теперь задачу определения структуры мультипроцессорной системы можно сформулировать следующим образом.

При заданных значениях Π – производительности процессоров обработки пакетов, объемов локальных памятей ПОП, стоимости модулей, составляющих структуру МУ, – определить такую структуру МУ, чтобы выполнялись следующие условия:

$$\Lambda(\Pi) \geq \frac{1}{L} \sum_{s=1}^S v_s \delta_s, \quad (1)$$

где L – средняя длина пакета, бит; v_s и δ_s – скорость, бит/с, и коэффициент использования (загрузки) s -го канала связи:

$$\bar{t}_3(\Pi) \leq t_3^0, \quad (2)$$

где t_3^0 – допустимое среднее время задержки пакета в МУ, с;

для каждого входящего канала $s=1, \dots, S$

$$P_6^S(\Pi) \leq P_6^D, \quad (3)$$

где P_6^D – допустимая вероятность блокировки и искажения пакета:

$$\Phi(\Pi) \rightarrow \min_{\Pi}, \quad (4)$$

где Φ представляет собой аддитивную функцию стоимости всех модулей, входящих в состав МУ.

Решение сформулированной задачи распадается на два этапа. На первом этапе на соответствующих моделях находятся зависимости $\Lambda(\Pi); \bar{t}_3(\Pi); P_6^S(b_s)$.

На втором этапе решается оптимизационная задача, определяемая выражениями (1)–(4).

Исходными данными для процедуры оптимизации являются параметры генетического алгоритма:

Z – число особей в популяции;

$\chi_1, \chi_2, \chi_3, \chi_4$ – порядок операции кроссовера для разных хромосом;

q – вероятность мутации гена;

$\varepsilon, K^{\text{доп}}$ – критерии останова.

Каждая особь популяции описывается набором из трех хромосом:

1) N ;

2) $B_{11}^P, \dots, B_{MR}^P$;

3) M .

Алгоритм оптимизации включает следующие шаги.

Шаг 1. Выбрать начальную популяцию S_0 , включающую Z особей. Особь $s \in S_0$ включить в популяцию только при выполнении ограничений (1)–(3). Положить номер популяции $k: = 0$.

Шаг 2. Вычислить приспособленность каждой особи популяции $\Phi(s)$, $s=1, \dots, Z$, и приспособленность популяции в целом

$$\Phi_k^* = \min\{\Phi(s) \mid s = 1, 2, \dots, Z\},$$

где s – номер особи в популяции, который позволяет однозначно определить соответствующий этой особи генотип; $\Phi(s)$ – целевая функция задачи оптимизации.

Шаг 3. Селекция. Случайным образом выбрать родителей s_1, s_2 из популяции I_k в соответствии с распределением вероятностей $P_1, \dots, P_s, \dots, P_Z$, где

$$P_s = \frac{\Phi(s)}{\sum_{s \in S_k} \Phi(s)}.$$

Шаг 4. Скрещивание. Построить s' по s_1, s_2 при помощи оператора χ_i -точечного кроссовера применительно к i -й хромосоме. Одну из двух вновь образующихся хромосом выбрать в качестве хромосомы потомка с вероятностью 0,5.

Шаг 5. Мутация. Модифицировать s' , с вероятностью q заменив значение каждого гена потомка на противоположное.

Шаг 6. Формирование новой популяции, которая объединяет решения следующего поколения. Для этого Z раз выполнить операции алгоритма, начиная с шага 3, по-

мещающую каждую вновь полученную особь в следующую популяцию S_{k+1} . Затем положить номер новой популяции $k := k + 1$ и перейти к шагу 7.

Шаг 7. Останов. Если приспособленность L последних популяций увеличилась на величину меньшую, чем ε , или если $k = K^{\text{доп}}$, то остановить процесс эволюции. В качестве решения задачи оптимизации взять наилучшую из найденных особей последней популяции.

Литература

1. **Кутузов О.И., Сергеев В.Г., Татарникова Т.М.** Коммутаторы в корпоративных сетях. Моделирование и расчет. –СПб.: Судостроение, 2003.
2. **Golberg D.E.** Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, Mass., 1989.
3. **Костенко В.А.** Принципы построения генетических алгоритмов и их использование для решения задач оптимизации//Труды IV Международной конференции "Дискретные модели в теории управляющих систем" (19–25 июня 2000 г.). – С. 49–55.