

## ЯЗЫКОВЫЕ КОНСТРУКЦИИ ДЛЯ ПАРАЛЛЕЛЬНЫХ АСИНХРОННЫХ ВЫЧИСЛЕНИЙ НА РЕКОНФИГУРИРУЕМЫХ МНОГОПРОЦЕССОРНЫХ СИСТЕМАХ

*Аннотация.* Предложен язык параллельной обработки для многопроцессорных вычислительных систем, позволяющий эффективно проводить вычисления при реконфигурации системы.

*Ключевые слова:* язык программирования, параллельные асинхронные вычисления.

**Введение.** Реконфигурируемые многопроцессорные вычислительные системы (МВС) представляют собой системы, у которых возможно изменение числа мультипроцессоров. Актуальной является разработка языка программирования для параллельных асинхронных вычислений на реконфигурируемых многопроцессорных системах.

**Постановка задачи.** Необходимо разработать языковые конструкции для эффективных параллельных асинхронных вычислений на реконфигурируемых многопроцессорных системах.

**Основная часть.** Модель обработки данных с учетом пространственного параллелизма и параллелизма задач для выполнения на МВС предложена в [1].

Управление выполнением задания базируется на методе асинхронного программирования [2]. Изначально все процедуры считаются взаимно-независимыми и параллельными. Каждое возможное упорядочивание выполнения процедур задается наложением определенных ограничений с помощью управляющих переменных. Управляющие взаимодействия между процедурами осуществляются путем изменения управляющих переменных, индивидуального отслеживания этих значений и индивидуальной реакции на их изменение.

Программист явно определяет информационные и управляющие связи между процедурами.

Каждой процедуре ставится в соответствие условие инициирования - спусковая функция, представляющая собой логическую функцию от управляющих переменных. Спусковая функция служит для указания используемых данных (информационные связи) и процедур, после которых может быть запущена данная процедура (управляющие взаимодействия).

В качестве управляющих переменных используются логические переменные процедур, наборов данных и параметров, имеющие соответствующие идентификаторы, логические переменные для синхронизации, а также специальные переменные, задаваемые и модифицируемые в процессе выполнения задания.

В разделе управления задания описываются управляющие переменные и определяются связи между процедурами. Для процедуры задается ее вызов, спусковая функция, выход и, если необходимо, модификация специальных переменных.

Вызов процедуры служит для активации процедуры. В выходе процедуры указываются общие наборы данных, которые использовались процедурой и освобождаются ею, ее выходные параметры, необходимые другим процедурам.

Каждой процедуре задания соответствует логическая переменная, значение которой до выполнения процедуры равно 0, а после – 1. Набору данных или параметру процедуры, указанному в спусковой функции, соответствует логическая переменная, которая принимает значение 1, если набор данных свободен, а параметр выдан процедурой, и 0, если набор данных занят, а параметр еще не выдан процедурой.

Специальные переменные и выходные параметры процедур используются для организации циклов и условных переходов в задании.

Инициализация управляющих переменных задает следующие их значения. Все логические переменные общих наборов данных устанавливаются в 1. Логические переменные выходных параметров устанавливаются в 0. Специальные переменные, управляющие логические переменные и параметр конфигурации принимают значения заданные в описании.

Для указания возможности параллельного выполнения одной процедуры над разными фрагментами данных она параметризуется

ся. Вместе с процедурами могут быть параметризованы наборы данных, параметры процедур, соответствующие им управляющие логические переменные, а также специальные переменные. Параметризация задается указанием идентификатора параметра в квадратных скобках после идентификатора процедуры или управляющей переменной. Состояние  $i$ -го экземпляра процедуры (набора данных, параметра) определяется  $i$ -м элементом массива соответствующих переменных.

Приведем описание синтаксиса раздела управления задания с помощью модифицированных форм Бэкуса-Наура.

```

раздел_управления ::= control инициализация_переменных
                        задание_параметра_конфигурации
                        определение_связей
задание_параметра_конфигурации ::= идентификатор = диапа-
зон ;
инициализация_переменных ::= { идентификатор [пара-
метр_конфигурации] = выражение ; }
определение_связей ::= оператор_связи { ; оператор_связи }
оператор_связи ::= спусковая_функция / [пара-
метр_конфигурации]
                        вызов_процедуры / [выход] /
[ : модификация_спец_перем]
вызов_процедуры ::= имя_процедуры (спи-
сок_фактических_параметров)
выход ::= идентификатор [параметр_конфигурации]
        { , идентификатор [параметр_конфигурации] }
модификация_спец_перем ::= спец_перем = выражение
спусковая_функция ::= логическое_выражение

```

В определенные моменты времени проверяются значения спусковых функций всех процедур. Те процедуры, значения спусковых функций которых «истинны», объявляются готовыми к выполнению. Это значит, что данные необходимые для этих процедур уже вычислены и/или управление передано на эти процедуры.

В случае указания выходного параметра процедуры в спусковой функции некоторой процедуры, ее запуск задерживается до окончания процедуры, выдающей этот параметр. Затем выполняется пересылка параметра к данной процедуре. При использовании общих

данных процедура считается готовой к выполнению, если никакая другая процедура в данный момент времени не работает с указанными в спусковой функции общими данными.

После выполнения процедуры значения управляющих логических переменных, указанных в выходе, устанавливаются в 1 и производится модификация специальных переменных.

При описании переменных задаются значения параметров конфигурации МВС. Параметризация процедур в разделе управления дает возможность выполнять задание на реконфигурируемой МВС с любым числом мультипроцессоров.

Для автоматического выполнения задачи над всем изображением в спецификации процедуры описываются системные параметры, такие как размеры фрагмента и число мультипроцессоров. Число фрагментов набора данных является одной из специальных управляющих переменных. В спусковой функции перехода, соответствующей прикладной процедуре, присутствует проверка на неравенство нулю этой специальной переменной. После обработки одного фрагмента она уменьшается на единицу в процедуре модификации управляющих переменных перехода.

Пример описания раздела управления задания. Прикладная процедура PSG выполняет обработку входного изображения из файла А, результирующее изображение записывается в файл В.

```
control
i=1..nmp;
nfr; S[i]=1; IB[i]=nbi; OB[i]=nbo;
A and S[i] and nfr>0 and IB[i]>0 / [i] INP() / AM,A,S[i], nfr--,
IB[i]--
AM[i] and OB[i]>0 / [i] PSG() / BM[i],OB[i]--,IB[i]++
BM[i] and B and S[i] / [i] OUT() / B, S[i], OB[i]++
nmp – число мультипроцессоров в МВС,
nfr – число фрагментов изображения,
INP(), OUT() – процедуры ввода и вывода соответственно,
AM, BM – входной и выходной фрагменты изображения,
i, IB, OB – параметры конфигурации, соответствующие мультипроцессорам, входным и выходным буферам,
nbi, nbo – число входных и выходных буферов соответственно,
S – управляющая переменная.
```

Возможно моделирование выполнения задания, описанного на приведенном языке, параметризированной сетью Петри, описанной в [3].

**Выводы.** Предложены языковые конструкции раздела управления языка параллельных вычислений на реконфигурируемых МВС, которые предоставляют возможность описания задания на обработку. Эффективное выполнение обработки системой определенной архитектуры достигается за счет задания системных параметров конфигурации.

### ЛИТЕРАТУРА

1. Пономарев И.В., Федотов А.Г. Структурные свойства данных и методы построения программного обеспечения реконфигурируемых многопроцессорных систем с общим управлением.- М.: Вопросы кибернетики, 170, 1991, с. 132-158.
2. Элементы параллельного программирования/ Под ред. В.Е. Котова. М.: Радио и связь, 1983. 240 с.
3. Пономарев И.В. Моделирование параллельных вычислений для эффективной организации процесса обработки изображений на МВС // Системные технологии. Региональный межвузовский сборник научных работ. Выпуск 1(72). – Днепропетровск, 2011, - с. 45-50.