

Модель памяти в технологии графосимволического программирования

П.В. Аболмасов, В.В. Жидченко

Самарский государственный аэрокосмический университет имени академика С.П. Королёва (национальный исследовательский университет)

На кафедре программных систем Самарского государственного аэрокосмического университета разработана и развивается технология графосимволического программирования (технология ГСП), предназначенная для визуального проектирования и разработки параллельных программ. Технология использует графическую форму представления программ, семантически близкую к блок-схемам, дополненную механизмами описания параллелизма и синхронизации. Пример представления программы в технологии ГСП приведен на рисунке 1.

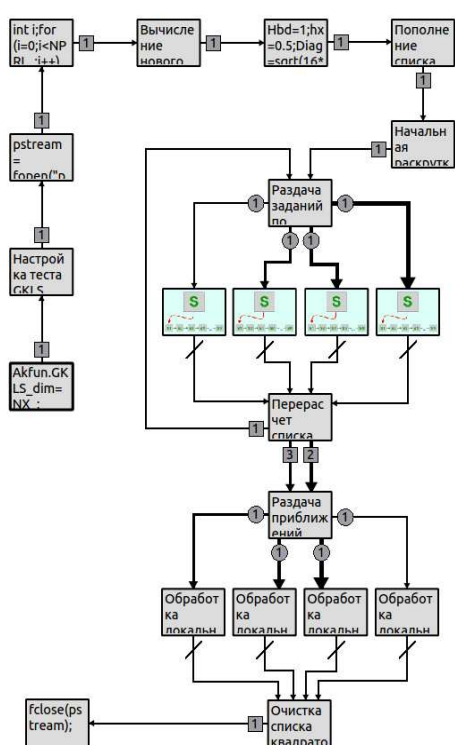


Рис 1. Граф-модель параллельного алгоритма нахождения глобального экстремума

Вершинами графа служат подпрограммы на языке С или другие графы. Направленные дуги графа определяют передачу управления между вершинами. По графическому описанию программы специализированный компилятор технологии ГСП строит исходный текст на языке С. Исполняемый файл формируется из этого исходного текста обычным компилятором С с использованием библиотеки MPI.

Изначально технология создавалась для разработки последовательных программ, поэтому она ориентирована на модель общей памяти. Для исполнения разрабатываемых программ в распределенных системах необходимо расширение модели памяти, используемой в технологии ГСП.

Все переменные программы разделяются на два подмножества: глобальные и локальные. Глобальные хранятся и изменяются единственным процессом – диспетчером данных. Для локальных переменных в каждом использующем их процессе создается отдельная копия переменной. Обмен значениями локальных переменных между процессами осуществляется с помощью сообщений. Сообщения изображаются графически в виде направленной дуги от вершины – источника сообщения, в вершину – получатель сообщения. Дуга помечается именами передаваемых в сообщении переменных.

Компилятор технологии ГСП генерирует исходный текст программы, в котором все переменные инкапсулируются объектами класса TPOData. Доступ к переменным предоставляется через поля-свойства. Свойство - это способ доступа к внутреннему состоянию объекта, имитирующий переменную некоторого типа. Обращение к свойству выглядит так же, как и обращение к полю объекта, но в действительности оно реализовано через вызов функции. При попытке задать значение свойства вызывается один метод (setter), а при попытке получить - другой (getter). Объекты класса TPOData создаются в каждом процессе параллельной программы. Поля-свойства для локальных переменных, а также для глобальных переменных в диспетчере данных обращаются к переменным по указателям. При доступе к глобальным переменным из остальных процессов поля-свойства работают с переменными, формируя диалог с диспетчером данных с помощью сообщений MPI.

Класс TPOData напрямую недоступен для пользователя технологии ГСП. Пользователь работает непосредственно с переменными программы и не обязан знать об их реализации.