

Использование асинхронных вычислений в функциональном языке потоково-параллельного программирования

И.В. Матковский

Сибирский Федеральный Университет

Одной из актуальных проблем организации параллельных вычислений является обработка асинхронных вычислений. Различная скорость работы отдельных модулей может привести к тому, что необходимые для работы системы в целом данные будут поступать с задержками. Асинхронная модель вычисления должна эффективно обрабатывать данные вне зависимости от того, с какими задержками и в каком порядке они поступают.

Предлагаемая модель асинхронных списков [1] предназначена для функционального языка потоково-параллельного программирования “Пифагор” [2] и построена на использовании асинхронных списков. Асинхронный список хранит поступающие в него фрагменты данных в порядке поступления. Список может находиться в одном из двух состояний – либо он пуст, либо содержит как минимум один элемент.

```
Async_Sum<< funcdef A {
  x1<< A:1;
  tail_1<< A:-1;
  [(tail_1:[IsEmptyDataList, IsNotEmptyDataList]:?)^
  ( x1,
  { block {
    x2<< tail_1:1; s<< (x1,x2):+; tail_2<< tail_1:-1;
    [(tail_2:[IsEmptyDataList, IsNotEmptyDataList]:?)^
    ( s,
    { asynch( tail_2:[], s):A_VecSum }
    ):. >>break;
    }):. >>return;}
  ])
```

Рис. 1. Алгоритм суммирования асинхронного списка

Выделяется первый (головной) элемент асинхронного списка x_1 (получаемый командой выборки $A:1$); после этого проверяется хвост списка $tail_1$ (получаемый командой выборки-исключения $A:-1$). Если хвост списка $tail_1$ пуст, то список A состоит из одного элемента x_1 , который и следует вернуть в качестве результата функции. Если хвост списка $tail_1$ не пуст, то выделяем его головной элемент x_2 и суммируем его с x_1 ; после этого проверяем “хвост хвоста” $tail_2$. Если $tail_2$ пуст, то A содержал только элементы x_1 и x_2 . Возвращаем их сумму в качестве суммы списка. Если $tail_2$ не пуст, создаем новый асинхронный список, в который помещаем сумму x_1 и x_2 и $tail_2$; вызываем для него рекурсивно функцию суммирования.

Литература

1. Легалов А.И., Редькин А.В. Расширение асинхронного управления по готовности данных / Труды III Международной конференции «Параллельные вычисления и задачи управления» РАСО’2006. — ISBN 5-201-14990-1 / М.: Институт проблем управления им. В.А. Трапезникова РАН, 2006. С 1272-1281. (Электронное издание)
2. Легалов А.И. Функциональный язык для создания архитектурно-независимых параллельных программ // Вычислительные технологии. 2005. № 1 (10). С.71-89.