

ОБЛАЧНАЯ СРЕДА ФУНКЦИОНАЛЬНОГО ПРОГРАММИРОВАНИЯ ДЛЯ НАУЧНЫХ ВЫЧИСЛЕНИЙ И ОБРАЗОВАНИЯ

Р.И. Идрисов

Институт систем информатики им. А.П. Ершова СО РАН, Новосибирск

Доклад посвящен проекту браузерной среды программирования, предназначенной для функциональных языков программирования и использующей в качестве вычислителя рабочую станцию и удалённую систему нефиксированной структуры.

Введение

На сегодняшний день всё большую популярность набирают облачные сервисы, которые предоставляют различные возможности. Конечно, под словом «облачный» подчас скрываются обычные вещи, которые просто были названы по-новому. Согласно последней редакции российской Википедии (<http://ru.wikipedia.org>) на момент написания статьи, облачный сервис – это просто некоторый доступный ресурс в сети, который может быть использован без знания его внутренней структуры.

Статистика поисковых запросов в некотором смысле отражает интересы общества. Если проследить статистику google (<http://www.google.com/insights/search/>) по двум терминам «parallel» и «cloud», можно увидеть, что популярность «cloud» с 2004 года выросла вдвое и продолжает расти, а популярность «parallel», наоборот, неуклонно падает и с 2004 года изменилась вдвое. Рост популярности термина «cloud» обусловлен тем, что облачность представляет пользовательский интерфейс для использования какого-то ресурса. Сам термин «облачный» по одной из версий образовался из того, что Интернет изображался в виде облака, сокрытая от пользователя структура, находящаяся в Интернете, и является этим самым облаком. Концепция, конечно, была высказана уже давно, но, как и с функциональными языками программирования, которые были описаны задолго до их популярности, развитие технологий достигло уровня, дающего возможность для массового применения облачности.

Продолжают набирать популярность облачные хостинги amazon, cloud9, которые предоставляют вычислительные ресурсы. В частности, они позволяют выполнять код на V8 (JavaScript от Mozilla), этот язык во многом совместим с браузерным JavaScript, что открывает возможность для создания переносимого кода. Одна и та же программа может быть исполнена как на персональной рабочей станции, так и на потенциально мощном облачном вычислителе. При этом здесь переносимость совсем не такая, как для Java. Программное обеспечение не требует дополнительной установки библиотек поддержки времени исполнения, а выполняется внутри браузера. Кроме того, это исполнение предполагается безопасным для пользователя. Растущее количество сервисов в сети Интернет, предоставляющих различные услуги, прямо или косвенно связанные с вычислениями, говорит о популярности такого подхода. Очень привлекательно, не имея компилятора, иметь возможность в любой момент зайти на соответствующую страницу в сети и выполнить интересующий код. Подобные сервисы варьируются от совершенно аскетичных, рассчитанных на исключительно короткие программы (<http://codepad.org>), до предоставляющих среду разработки с группировкой по проектам, подсветкой синтаксиса (<http://c9.io> – Cloud 9) и т.д.

Несмотря на большое количество таких ресурсов, всё же некоторая ниша остаётся незаполненной. В частности, нет таких, которые предоставляли бы возможность отладить и запустить научную задачу с массивным параллелизмом.

Следует отметить, что такой подход не является чем-то абсолютно новым, из-за большого количества разных параллельных систем разработчики стараются использовать переносимые варианты, которые могут быть исполнены в различных условиях, в тех случаях, когда, конечно, речь не идёт о специальном программном обеспечении для конкретного вычислителя.

Разрабатываемая система ставит перед собой две цели: научную и образовательную. Для научной цели более важна масштабируемость, а для образовательной – доступность. Кроме того, было бы неправильно ориентироваться только на один язык программирования, поскольку не существует единого мнения о наилучшем учебном языке. Для масштабируемости требуется универсальность описания параллелизма, это значит, что программа не должна быть адаптирована для структуры конкретной вычислительной системы. Согласно работам А.П. Ершова, это достигается, если язык программирования приближается к языку описания задач, а не к языку описания алгоритмов. В Институте систем информатики СО РАН мы продолжаем разработку потокового языка программирования Sisal [1, 2], эта работа ставит перед собой именно такие цели.

Литература

1. McGraw, J. R. et. al. Sisal: Streams and iterations in a single assignment language, Language Reference Manual Version 1.2 // Lawrence Livermore Nat. Lab. Manual M-146 (Rev.1). – Livermore, CA 1985.
2. Касьянов В.Н., Бирюкова Ю.В., Евстигнеев В.А. Функциональный язык SISAL 3.0 // Поддержка супервычислений и Интернет-ориентированные технологии. – Новосибирск, 2001. – С. 54-67.