

VERT.X КАК НОВОЕ СРЕДСТВО РАЗРАБОТКИ МИКРОСЕРВИСОВ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Садовников И.В.

Искра Н.А. – старший преподаватель

Vert.x - это асинхронный, событийно-ориентированный фреймворк работающий на JVM. Данный фреймворк показывает высокую производительность, легко конфигурируется и поддерживает асинхронную модель программирования. Первый релиз Vert.x состоялся в 2012 году [1].

Основными преимуществами Vert.x являются:

1. мультиязычность - компоненты приложения могут быть разработаны на Java, JavaScript, Groovy, Ruby, Ceylon, Scala или Kotlin;
2. асинхронность - очень простая схема асинхронного взаимодействия без блокировок;
3. параллелизм - простая модель параллельного выполнения задач, основанная на event-loops [2];
4. распределенная шина событий - данная шина позволяет распределять нагрузку между всеми вертикалями [3] автоматически;
5. модульная структура - модули позволяют использовать только те компоненты, что необходимы для проекта;
6. Open Source - полностью открытый исходный код продукта, активно развиваемого сообществом.

Vert.x поддерживает механизмы кластеризации прямо из коробки (out of the box). По умолчанию для кластеризации используется Hazelcast, но также есть возможность использовать такие инструменты как Infinispan, Apache Ignite, Apache Zookeeper [4].

Пример команды запускающей вертикаль в кластере: `vertx run MyVerticle.java -cluster`

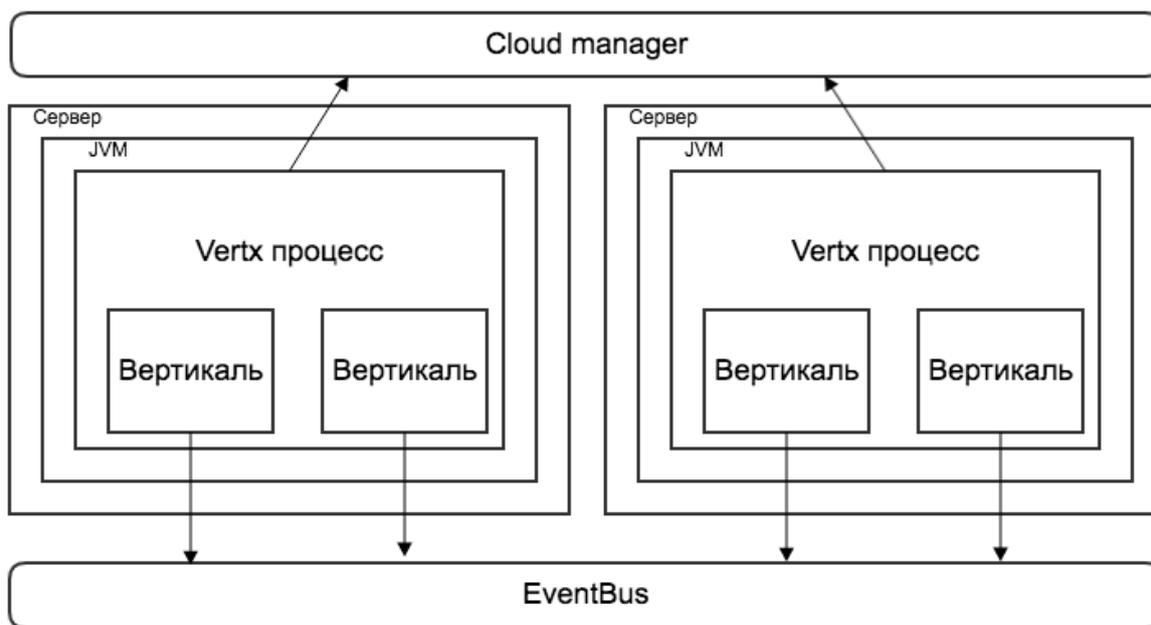


Рис 1. – Схема работы вертикалей Vert.x в кластере[3]

Vert.x поддерживает механизм отказоустойчивости своих компонентов, работающих в одном процессе vertx или в одном кластере, таким образом если часть компонентов выходит из строя, другая часть берет их функциональность на себя, пока решается проблема с остановившимися компонентами [5]. Для того чтобы воспользоваться данной функцией, в команду строку надо всего лишь добавить параметр запуска -ha. При этом запуск будет автоматически подниматься в кластере, поэтому необходимости добавлять параметр -cluster нет.

Пример команды запускающей вертикаль с механизмом отказоустойчивости:
`vertxrunMyVerticle.java-ha`

Список использованных источников:

1. Официальный сайт Vert.x [Электронный ресурс]. - Режим доступа: <https://vertx.io/>
2. Официальная документация основного модуля Vert.x [Электронный ресурс]. - Режим доступа: <https://vertx.io/docs/vertx-core/java/>
3. Официальная документация о Verticle [Электронный ресурс]. - Режим доступа: https://vertx.io/docs/vertx-core/java/#_verticles
4. Julien Ponge, Thomas Sigismont, Julien Viet, A gentle guide to asynchronous programming with Eclipse Vertx for Java

developers.

5. Clement Escoffier. Building Reactive Microservices in Java.

МАТЕМАТИЧЕСКИЙ КАЛЬКУЛЯТОР

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Ковбаса Г.А.

Ковальчук А.М. – старший преподаватель

Проводя обзор аналогов, наиболее широко распространенными программными продуктами, занимающимися символьными и численными вычислениями, являются СКА – системы компьютерной алгебры, такие как AXIOM, Maxima, Reduce и более простые Derive, MathCAD, Maple, Mathematica-2.1, WolframAlpha.

Преимущество СКА – в полноте представленных операций, возможности составления пользователем своих собственных алгоритмов большой вычислительной мощности, подходящей для выполнения расчетов в высокоточных исследованиях и широкой представленности на рынке.

С другой стороны, недостатками таких программ являются их высокая стоимость, необходимость соединения с интернетом, повышенные требования к системе компьютера (процессор: Intel Pentium Dual-Core или эквивалентный; дисковое пространство: 14 GB; оперативная память (RAM): рекомендовано 2 GB+ - для Mathematica, процессор: частота 1 GHz или выше; дисковое пространство: 4 GB; оперативная память (RAM): 4 GB+ - для Maple 2018), длительное усвоение навыков работы и т.п.

Системы компьютерной алгебры, в большинстве своем, базируются на древовидных структурах данных, традиционно характерных для языков, реализующих парадигму функционального программирования, каким, к примеру, является Lisp. В моей программе на основе введенного пользователем выражения, путем последовательного считывания элементов строки, создается объект класса дерева выражений. Данное дерево представляет собой усовершенствованное В-дерево (сбалансированное, сильноветвистое дерево), узлы которого могут быть двух видов: узлы суммы и узлы произведения (рисунок 1).

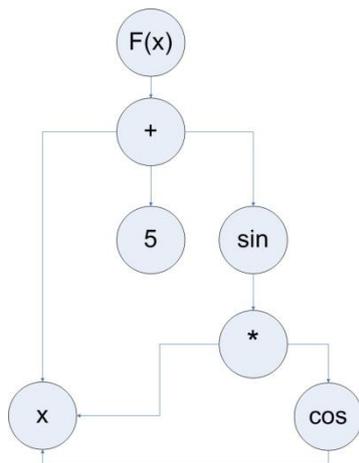


Рисунок 1 – Схема дерева выражения

Корень дерева представляет собой сумму всех последующих операций. Данное представление обосновано тем, что любую математическую операцию можно представить в виде суперпозиции этих двух операций над некоторыми величинами. К примеру, операция вычитания является операцией сложения двух величин, одна из которых имеет отрицательный коэффициент.

Поиск в дереве и его анализ на зависимость функций от некоторой другой функции или переменной проводятся за меньшее время, чем в стандартном бинарном дереве поиска, т.к. доступ к любому элементу суперпозиции выражения можно получить из корневого узла, и у каждого узла-потомка может быть несколько узлов-предков. Таким образом, алгоритмы интегрирования, дифференцирования и вычисления значения будут производиться быстрее.

Поэтому, в сравнении с рассмотренными аналогами, достоинствами моей разработки являются низкие системные требования (процессор: 0.233 GHz или выше; свободное пространство на диске: 100 MB; оперативная память (RAM): 64 Мб RAM или выше), портативность, отсутствие необходимости подключения к сети интернет, наличие базовых возможностей символьных вычислений, достаточных для использования в качестве вспомогательного средства для учебы и домашнего пользования.

Программа имеет удобный пользовательский, с необходимыми пунктами меню, интерфейс на базе Windowsforms. В программе выполняются следующие действия: работа с дифференциалами, основными пределами и интегралами, а также построение графиков функций одной переменной. Предусмотрена операция отмены последних действий и сохранение промежуточных пользовательских вычислений.

Для реализации приложения используется объектно-ориентированный язык программирования C++, среда разработки MicrosoftVisualStudio 2015, ОС Windows7.

Список использованных источников:

1. Страуструп, Б. Программирование: принципы и практика использования C++ / Б. Страуструп : пер. с англ. — М. : ООО "И.Д. Вильямс", 2011. — 1248 с.: ил.
2. Тан, К.Ш. Символьный C++: Введение в компьютерную алгебру с использованием объектно-ориентированного программирования / К. Тан, В. Стиб, И. Харди : пер. со 2-го англ. изд. — М.: Мир, 2001. — 622с, ил.
3. Дэвенпорт, Дж. Компьютерная алгебра / Дж. Дэвенпорт, И. Сирэ, Э. Турнье. - М.: Мир, 1991. - 352 с.
4. Wikipedia[электронный ресурс]– Режим доступа https://ru.wikipedia.org/wiki/К-мерное_дерево – Дата доступа 05.11.2017