А. А. Бурак

NODE.JS

Node.js – это событийно-ориентированный фреймворк на языке Javascript для создания сетевых приложений. Основная идея в том, что в ходе выполнения кода ничто не блокируется, отсутствуют операции, которые чего-то ждут, например передачи данных, ввода пользователя или установки соединения. Все построено на событиях, которые происходят в момент наступления того, чего синхронные операции дожидаются. Это дает значительное, иногда в десятки раз, преимущество в производительности по сравнению со старыми синхронными системами.

Введение

Многие разработчики всегда считали JavaScript просто дополнением к браузеру, которое годится разве что для управления формами и манипулированиеи DOM-деревом веб-страницы. Действительно, язык очень простой. Впрочем, настоящие профессионалы давно научились творить с его помощью чудеса, предоставив пользователям потрясающе удобные онлайн-сервисы, которыми мы ежедневно пользуемся. Многие из таких разработчиков пошли дальше и, трезво посмотрев на сам язык и его возможности, особенно по части работы с событиями, решили: а что если на JavaScript написать сервер? Разработчики получают возможность написать на одном и том же языке все части сайта: что серверную часть, что клиентскую страницу. Кроме того, JS очень простой и одновременно гибкий язык прораммирования, что позволяет писать код в разных парадигмах: от обычного процедурного до ООП в смеси с функциональным стилем.

Тотальная асинхронность

Самым главным в Node.js является тотальная асинхронность. Это значит, что код скрипта будет выполняться не последовательно, как в случае с PHP, Perl-скриптами, а именно в тот момент, когда для него будут готовы все данные. Основной особенностью Node, кроме полной асинхронности, является его однопоточная модель. Другими словами, все операции выполняются в одном и том же потоке ОС, даже если у сервера тысяча одновременных пользователей. Правда, доступно создание дочерних процессов и низкоуровневое управление исполнением скриптов (загрузка, компиляция, работа с ассемблерным кодом, исполнение). Для реализации многопоточности и задействования всех ядер современных процессоров рекомендуется просто загружать несколько копий приложения, но можно взять на вооружение WebWorker из стандарта HTML5 и распределить работу приложения по нескольким дочерним процессам. Второй особенностью архитектуры Node является событийность [1]. Почти каждая операция имеет коллбэки, генерирует событие, а пользователю доступен объект EventEmiter, через который можно буквально одной строкой генерировать свои события. Сам по себе Node построен вокруг EventLoop [2], глобального цикла обработки событий, который на каждом тике проверяет, готовы ли данные для какого-либо из определенных пользователем коллбэков. В дополнение к этому в сервер встроен очень эффективный сборщик мусора (GC), поэтому даже тысячи подключений не вызывают переполнения памяти и падения сервера.

1. The Node Beginner Book [Electronic resource] / M. Kiessling. – A Node.js tutorial by Manuel Kiessling, 2012. – Mode of access: http://www.nodebeginner.org/. – Date of access: 01.07.2012.
2. Up and Running with Node.js [Electronic resource] / T. H. Croucher. – O'Reilly, 2012. – Mode of access: http://www.amazon.com/Node-Running-Scalable-Server-Side-JavaScript/dp/1449398588. – Date of access: 07.05.2012.

Бурак Александр Анатольевич, студент 4 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлекстроники, alexander.burak@outlook.com.

Научный руководитель: Герман Олег Витольдович, доцент кафедры информационных технологий автоматизированных систем Белорусского государственного университета, кандидат технических наук, доцент, ovgerman@tut.by.