

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ NINJA И АЛЬТЕРНАТИВНЫХ СИСТЕМ СБОРКИ В ЗАДАЧАХ ПОЛНОЙ И ИНКРЕМЕНТАЛЬНОЙ СБОРКИ

Лобанов И.В. (ИТМО)

Научный руководитель – кандидат педагогических наук Валитова Ю.О. (ИТМО)

Введение. Системы сборки играют ключевую роль в процессах разработки, влияя на производительность, масштабируемость и расход системных ресурсов при работе над проектами различной сложности. На фоне существующих систем сборки можно выделить Ninja, легковесный и высокопроизводительный инструмент, разработанный для повышения скорости работы и обеспечения простоты использования [2], в сравнении с популярными альтернативами, такими как Make, Gradle и Bazel. Ninja, известная своим минималистичным подходом и чрезвычайно высокой скоростью выполнения сборки. Она часто используется для работы с языками скомпилированного типа, такими как C++, а также совместно с генераторами систем сборки, например CMake. Следует отметить, что Ninja, заточенная на скорость сборки и особенно эффективная в больших проектах [2], имеет более упрощенную функциональность в сравнении с более богатыми возможностями инструментов, таких как Gradle и Bazel, которые акцентированы на многоязычных и зависимость-ориентированных сценариях разработки. Цель исследования - сравнение производительности системы сборки Ninja с такими инструментами, как Make, Bazel и Gradle.

Основная часть. В целях объективности сравнения в рамках исследования был создан ряд тестовых проектов, отличающихся по размеру и сложности (от одноязычных до многоязычных сред), которые имитируют реальные сценарии разработки.

Методология исследования включала измерение ключевых метрик сборки: полного времени сборки (компиляция проекта с нуля), времени на итеративную (инкрементальную) сборку после внесения изменений в код, а также расхода системных ресурсов (использование CPU и оперативной памяти). Также оценивались масштабируемость системы сборки Ninja и ее эффективность в сравнении с расширенными механизмами управления зависимостями и поддержкой многоязычных проектов, которые предоставляют Gradle и Bazel.

При исследовании особое внимание уделялось большим проектам, которые являются наиболее наглядными при сравнении производительности между отдельными системами сборки. Тут Ninja достигает значительного выигрыша благодаря своему минималистичному подходу к управлению графами зависимостей. В отличие от Make, Ninja не пересобирает лишние таргеты при инкрементальных изменениях кода, что приводит к существенной экономии времени. Выделяет Ninja и поддержка параллельной сборки, позволяя максимально эффективно использовать ресурсы современных многоядерных процессоров [1].

Bazel проявила себя как более гибкая и функциональная система в условиях сложных многоязычных проектов, где может потребоваться распределённая сборка или глубокое управление зависимостями. Gradle, в свою очередь, оказался сильным инструментом для JVM-проектов (Java/Kotlin), благодаря встроенной поддержке инструментов для управления зависимостями и широкой экосистеме плагинов.

Исследование показало, что Make, несмотря на свою простоту, менее эффективен на крупных проектах из-за замедленного процесса проверки зависимостей и отсутствия должной оптимизации по сравнению с Ninja или Bazel.

Выводы: проведено сравнение системы сборки Ninja с системами сборки Make, Gradle, Bazel. Получены метрики времени полной сборки, итеративной сборки, расхода системных ресурсов для проектов различной сложности. Следует отметить, что у каждой из

рассмотренных систем сборки есть свои достоинства и недостатки, которые необходимо учитывать при выборе ее для реализации проектов.

Список использованных источников:

1. Смирнов В.М. Как сделать из Ninja систему распределенной сборки? – URL: <https://habr.com/ru/articles/321660/> (дата обращения: 16.02.2024).
2. Antreas Antoniou., Bala Priya C Using Ninja Build to Build Projects Faster – URL: <https://earthly.dev/blog/> (дата обращения: 16.02.2024).
3. Жиров А., Арсеньева Т.П. Система мета-сборки GN: краткий обзор и подходы – URL: <https://habr.com/ru/companies/vk/articles/793854/> (дата обращения: 16.02.2024).

Лобанов И. В. (автор)
Валитова Ю.О. (научный руководитель)

Подпись
Подпись