интервалами времени T, не превышающими 1(2) с. Общая скорость передачи данных в используемых сетях первичных интерфейсов (RS232, RS485…) на порядок меньше скорости в ЛВС и согласуется с помощью специального программного обеспечения.

Для выравнивания скоростных потоков сети первичных интерфейсов и ЛВС предлагается метод повышения пропускной способности каналов низкоскоростной сети до уровня ЛВС. Импульсы передачи данных в каждом канале далее рассматриваются как перекрывающиеся во времени импульсы.

Особенностью использования метода сжатия данных является многократное сжатие исходных данных до минимально возможного значения, которое определяется минимальным количеством узловых точек для интерполятора. Например, для числа узловых точек 3, децимации между ними 64 и разрядности одного отсчета 32 минимальное значение составляет 96 Бит.

С целью снижения временных затрат на многократные процедуры компрессии/декомпрессии указанные схемы можно реализовать на быстродействующих процессорах, созданных на основе перепрограммируемых логических матриц Spartan 6 LX150 FPGA, можно заменить побитный ввод информации на вход компрессора алгоритмом параллельно-последовательного ввода. В результате производительность компрессора увеличивается как минимум в 64 раза.

УДК 004.94:378

Построение программного обеспечения на основе непрерывной интеграции

Дадыкин А.К., Ермолаев А.А.
Белорусский национальный технический университет

Существует множество различных техник, облегчающих разработку и сопровождение программного обеспечения (ПО) в промышленных масштабах: тестирование, система управления версиями, система отслеживания ошибок, автоматизированная система сборки и развертывания и т.д. Непрерывная интеграция (Continuous Integration, CI) объединяет все эти компоненты в единое целое.

Непрерывная интеграция – это практика разработки программного обеспечения, которая заключается в выполнении частых автоматизированных сборок проекта для скорейшего выявления и решения интеграционных проблем.

Термин CI был введен Мартином Фаулером и Кентом Беком. Данный термин был придуман ими для обозначения практики частой интеграции

проекта. В настоящее время CI одна из практик, применяемых в семействе гибких методологий разработки ПО.

Когда большое число разработчиков совместно работают над сложными программными проектами, интеграция разных частей кода может превратиться в длительный процесс с непредсказуемыми результатами. Однако на проектах, где процесс разработки строится на основе методик CI, проблемы и риски, связанные с интеграцией, сведены к минимуму.

В проекте, выполняемом одним человеком, интеграция ПО не является существенной проблемой, но при увеличении сложности проекта возникает необходимость в интеграции и проверке слаженной работы компонентов ПО. CI снижает трудоёмкость интеграции и делает её более предсказуемой, за счет наиболее раннего обнаружения и устранения ошибок и противоречий.

В сущности, CI гарантирует совместимость недавних изменений во вновь разработанном ПО с остальной частью ПО. На более высоком уровне CI повышает коллективную ответственность группы разработчиков и снижает трудоемкость проекта, уменьшая объем ручного труда, выполняемого при каждой интеграции.

Если описать данную методику в несколько фраз, то это: “всегда есть рабочая версия”, “автоматизированная сборка”, “всегда известно, в каком состоянии прибывает проект”.

УДК 621. 391. 25

Уязвимость алгоритма формирования общего ключа с помощью искусственных нейронных сетей к простой и геометрической атакам

Голиков В.Ф., Брич Н.В.
Белорусский национальный технический университет

Использование синхронизируемых искусственных нейронных сетей (ИНС) является одним из перспективных решений задачи формирования общего секретного ключа. Архитектура на стороне отправителя и получателя представляет собой двуслойный персептрон (TPM-архитектура), состоящий из K внутренних персептронов, каждый из которых имеет N входов. В случае простой атаки (атаки методом грубой силы) злоумышленник E знает значения входного вектора x_{ij}., выходов $Z^{\frac{A}{B}}$ на каждом шаге обучения; использует то же правило обучения, что и легитимный отправитель A (однако вместо значения Z^E использует Z^A). В случае геометрической атаки злоумышленник не пропускает итерацию,