

Квазар-Оффлайн: распределенный вычислительный комплекс для решения задач распознавания образов

Качалков А. В., Хачай М. Ю.

kachalkov@gmail.com, mkhachay@imm.uran.ru

Екатеринбург, Институт математики и механики УрО РАН

Работа посвящена анализу и моделированию предметной области в рамках унифицированного процесса, с использованием универсального языка моделирования UML, построению универсальной модели для описания задачи распознавания образов, описанию нескольких конкретных алгоритмов распознавания в терминах модели, и разработке распределенного вычислительного портала «КВАЗАР-Оффлайн» для решения задач распознавания образов и анализа эмпирических данных.

Основные результаты работы

1. Разработана новая методология создания распределенных вычислительных порталов для различных предметных областей. В рамках разработанной методологии построена универсальная модель подсистемы размещения и мониторинга заданий, с использованием унифицированного процесса разработки и языка моделирования UML.
2. В соответствии с предложенной методологией, реализован оригинальный распределенный вычислительный комплекс для решения задач распознавания образов, который включает в себя:
 - (a) подсистему размещения и мониторинга заданий, реализованную на базе системы управления контентом **DotNetNuke**:
 - оригинальный специализированный модуль **Quasar**, реализующий доступ к системе;
 - база данных **Quasar** в соответствии с построенной концептуальной моделью, хранимые процедуры и функции;
 - компоненты доступа к базе данных **Quasar**.
 - (b) вычислительную подсистему:
 - оригинальный базовый асинхронный вычислительный компонент, в рамках подхода, основанного на событиях, облегчающий доступ к инфраструктуре вычислительного портала;
 - вычислительная служба, отвечающая за инициализацию и запуск текущих заданий на счет с использованием подключаемой технологии загрузки вычислительных алгоритмов.
3. Разработаны новая концепция и рекомендации по реализации вычислительных алгоритмов, использующих вычислительную библиотеку



Рис. 1. Архитектура базового асинхронного компонента `QuasarEngine`.

теку численного анализа `dnAnalytics` на базе платформы `.NET Framework 2.0`¹.

Базовый асинхронный компонент `QuasarEngine`

Для того, чтобы вычислительная служба `Quasar` могла загружать и запускать вычислительный алгоритм на счет, необходимо, чтобы вычислительный компонент `.NET` был унаследован от служебного компонента `QuasarEngine` (см. рис. 1).

Компонент `QuasarEngine` предназначен для того, чтобы предоставить разработчикам вычислительных компонентов необходимую функциональность для взаимодействия с инфраструктурой системы `Quasar` и существенно упростить процедуру разработки.

Компонент `QuasarEngine` реализует шаблон проектирования асинхронных вычислений, основанный на событиях². Реализация асинхронных вычислений необходима для того, чтобы иметь возможность передать запрос от пользователя об отмене вычислений непосредственно в вычислительный алгоритм. Также, данная реализация позволяет в перспективе реализовать поддержку сохранения промежуточного состояния

¹<http://msdn.microsoft.com/netframework>

²<http://msdn2.microsoft.com/en-us/library/wewwczdw.aspx>

вычислений с последующим продолжением вычислений с момента сохранения. Это может быть полезно для недетерминированного распределения вычислительного времени между заданиями и для приостановки вычислений от одного задания к другому.

Ключевым методом компонента является виртуальный метод `CalculateWorker`, в переопределенной версии которого разработчик и реализует свой вычислительный алгоритм, используя библиотеку численного анализа `dnAnalytics`. Остальные методы, свойства и события компонента `QuasarEngine` предназначены для поддержки шаблона асинхронных вычислений, основанного на событиях.

Так, метод `CalculateAsync` создает новую асинхронную операцию с помощью метода `CreateOperation` объекта `AsyncOperationManager` для задания с идентификатором `taskID`, и вызывает метод `CalculateWorker` в отдельном потоке, передавая входные параметры — `TaskParameters`, идентификатор вновь созданной асинхронной операции `AsyncOperation`, а также делегат `SendOrPostCallback`, представляющий из себя метод обратного вызова для передачи сообщения контексту синхронизации, который будет вызван по окончании асинхронной операции.

Метод `CancelAsync` предназначен для остановки текущих асинхронных вычислений с помощью метода `PostOperationCompleted` класса `AsyncOperation`.

Метод `ReportProgress` используется для публикации промежуточных и окончательных результатов в базе данных `Quasar`.

Работа выполнена на основе исследований, проводимых в отделе математического программирования Института математики и механики УрО РАН по государственной бюджетной теме № 01.2.00102387, проектам РФФИ № 04-01-00108, № 04-01-96104 и президента РФ по поддержке ведущих научных школ, гранты № НШ-792.2003.1, № НШ-5595.2006.1.