

# ПОСТРОЕНИЕ И СИСТЕМНАЯ РЕАЛИЗАЦИЯ МОДЕЛИ АСИНХРОННОЙ ОБРАБОТКИ ДАННЫХ

## CREATION AND SYSTEM PROVIDED IMPLEMENTATION OF MODEL OF ASYNCHRONOUS DATA PROCESSING

*D. Udin*

*Summary.* Development of modern administrative technologies and the organization of functioning of different control objects, beginning from the industrial enterprises, holdings, the transport systems and finishing with spheres of financial services and ensuring activity of the state and public institutes, generates essential number of tasks which solution demands application of the concept of distributed computing. First of all the tasks connected with providing information services, such as, interdepartmental electronic interaction, electronic public and municipal services, services of electronic transactions and payments and also scientific problems, problems of control and object management belong here. At the same time for the solution the listed tasks, as a rule, demand a real-time mode.

*Keywords:* information technologies, asynchronous data processing, distributed systems of information processing.

**Юдин Дмитрий Сергеевич**

Аспирант, Национальный исследовательский университет, «МЭИ» (Московский энергетический институт)  
order@echogroup.ru

*Аннотация.* Развитие современных управленческих технологий и организация функционирования различных объектов управления, начиная от промышленных предприятий, холдингов, транспортных систем и заканчивая сферами финансовых услуг и обеспечения деятельности государственных и общественных институтов, порождает существенное число задач, решение которых требует применения концепции распределенных вычислений. В первую очередь сюда относятся задачи, связанные с предоставлением информационных услуг, таких, как например, межведомственное электронное взаимодействие, цифровизованные государственные и муниципальные услуги, услуги электронных сделок и платежей, а также научные задачи, задачи контроля и управления объектами. При этом для своего решения перечисленные задачи, как правило, требуют режима реального времени.

*Ключевые слова:* информационные технологии, асинхронная обработка данных, распределенные системы обработки информации.

**С**инхронная программная модель подразумевает назначение потоку одной задачи вычисления. Когда завершается ее выполнение, тогда появляется возможность заняться другой задачей. Иными словами, в данной модели невозможно остановить выполнение задачи, чтобы в промежутке выполнить другую задачу.

Асинхронная модель программирования в отличие от синхронной предусматривает, что поток начав выполнение какой-либо одной задачи может приостановить ее выполнение и, сохранив ее текущее состояние, начать выполнение другой задачи.

Асинхронные вычислительные модели — класс моделей параллельных вычислений, в которых алгоритмы задач разбиваются на функционально обособленные части, представляемые компонентами, а потоки данных между компонентами буферизируются. Компоненты модели — автоматы специального вида с конечным набором состояний и фиксированной диаграммой переходов [1].

ИТ-инфраструктура предприятия всегда состоит из многих компонентов, включающих системное и прикладное программное обеспечение, которое создано в разное время, на разных платформах и с применени-

ем различных архитектурных подходов. Это приводит к тому, что задача интеграции этих составляющих частей оказывается гораздо сложнее, чем задача разработки их функционала [2].

Эти же условия актуальны и для случаев развертывания и реализации систем оказания государственных и муниципальных услуг в электронном виде и особенно для обеспечения межведомственного электронного взаимодействия, когда обмен данными, которые к тому же должны как правило не только иметь достаточно надежную защиту, но и быть заверенными электронными подписями, при этом разные ведомства в нашей стране изначально оснащались абсолютно разными системами внутреннего электронного документооборота и ведения баз данных.

При этом для каждого случая реализации электронных систем всегда весьма актуальными остаются: удобство использования интерфейса и производительность.

На практике получается, что пользователь системы запускает какую-либо функцию, например элементарную функцию сохранения данных, запускает процесс, который в свою очередь требует выполнения множества промежуточных задач, таких как чтение и заполнение

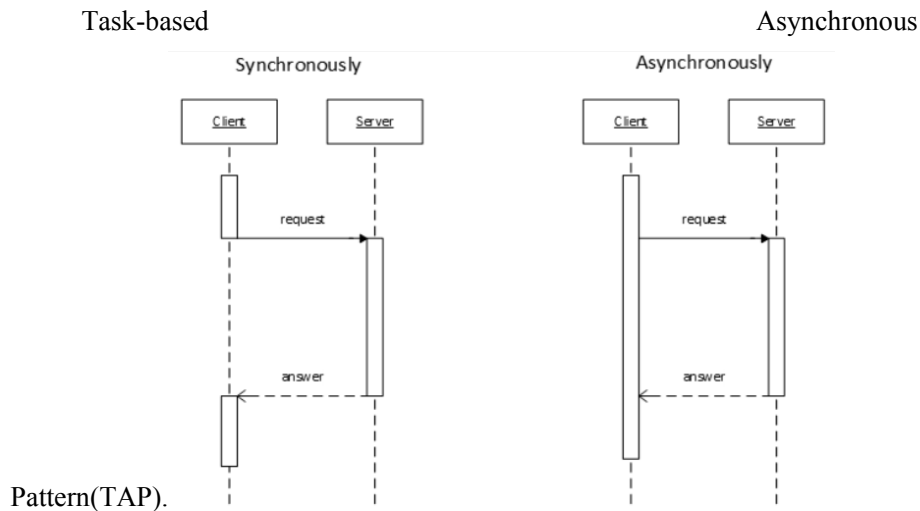


Рис. 1. Диаграмма реализации синхронного и асинхронного процессов.

данных во внутреннем объекте, установление соединения с SQL и сохранения его там.

В свою очередь SQL запускается на другой машине в сети и работает под другим процессом, что может занимать значительное время.

Иными словами, если запрос обрабатывается одним процессом, экран пользователя будет находиться в зависшем состоянии до тех пор, пока процесс не завершится. Именно поэтому большинство приложений и фреймворков сегодня основываются на асинхронной модели, позволяющей наиболее оптимальным образом использовать имеющиеся ресурсы электронных систем.

В то время как выполняется вышеупомянутый запрос пользователя на сохранение данных, около 70–80% ресурсов системы попадают в ожидание зависимых задач, и именно данное обстоятельство позволяет в случае применения асинхронного программирования максимально повышать вычислительную производительность.

Таким образом, как только задача передается другому потоку (например, SQL), текущий поток сохраняет состояние и доступен для выполнения другого процесса, а когда задача SQL завершается, любой поток, который является свободным, может заняться данной задачей [3].

Иными словами, в синхронном коде каждая операция не может начаться, пока не завершена предыдущая. Поэтому вся программа может зависнуть, если для одной из команд требуется слишком много времени на выполнение. Асинхронный код призван убирать, обходить такие блокирующие операции в основном потоке про-

граммы, так что она продолжает выполняться, но где-то в другом месте, а потоковые операции могут исполняться дальше, т.е. главный «процесс» ставит задачу и передает ее другому независимому «процессу».

Сопоставление последовательности синхронного асинхронного процессов представлены на рисунке 1.

Приведенная на рисунке схема описывает поведение потоков для клиент-серверного приложения.

Клиент отправляет запрос данных от сервера, и, вместо того чтобы «зависнуть» ожидая ответ (как в синхронном подходе) продолжает работать, предоставляя пользователю другой функционал приложения.

В настоящее время можно выделить три шаблона асинхронного программирования:

- Asynchronous Programming Model (APM);
- Event-based Asynchronous Pattern (EAP);

Шаблон Asynchronous Programming Model позволяет формировать асинхронные версии синхронных методов посредством двух методов — `Begin <MethodName>` и `End <MethodName>`.

Шаблон Event-based Asynchronous Pattern (EAP) основан на событиях и асинхронных методах. Класс, реализующий этот шаблон содержит методы `MethodNameAsync` и `MethodNameAsyncCancel`

Шаблон базируется на использовании задач. Основа TAP — два типа `System.Threading.Tasks.Task` и `System.Threading.Tasks.Task <TResult>`.

ТАР позволяет определять асинхронные функции в рамках одного метода, чем исключается необходимость создавать функции начала и конца асинхронной операции, а потом еще и обратного вызова.

Проектирование и системная реализация модели асинхронной обработки данных доказала свою востребованность при реализации большинства крупных проектов цифровизации в современной России. В частности, данная модель обработки данных нашла свое применение в реализации отечественных систем спутникового позиционирования и связанных с ней смежных сфер, например таких, как определение границ и кадастрирование земельных участков.

При этом пожалуй самым масштабным из реализуемых сегодня в России проектов, где за основу уже взят принцип асинхронной обработки данных является повсеместное внедрение системы межведомственного электронного взаимодействия (СМЭВ), построение которой должно помогать органам власти исполнять Федеральный закон № 210-ФЗ от 27.07.2010 «Об организации предоставления государственных и муниципальных услуг», согласно которому государственные и муниципальные органы не имеют права требовать с обратившихся к ним за госуслугами граждан какие-либо документы, или сведения, если такие документы или сведения име-

ются в распоряжении других подразделений государственной либо муниципальной власти. Такие сведения органы власти должны запрашивать самостоятельно друг у друга через СМЭВ.

Система межведомственного электронного взаимодействия позволяет федеральным, региональным и местным органам власти, контроля и надзора в электронном виде передавать и обмениваться данными, необходимыми для оказания госуслуг. Система позволяет реализовать принцип «одного окна» при оказании госуслуг населению. Гражданин обращается за услугой в профильное ведомство, а специалисты ведомства добывают необходимые данные в других ведомствах, используя СМЭВ. Применяемые в рамках такого взаимодействия цифровые технологии, связанные с обработкой огромных массивов информации, требуют дальнейшей оптимизации процессов поиска и обработки данных согласно заданным параметрам.

Построение и системная реализация модели асинхронной обработки данных позволяет не просто повысить производительность систем, но и решать широкий спектр прикладных задач, реализуя комплексную интеграцию приложений в информационных системах уровня предприятия (учреждения, или ведомства) а также на межведомственном и межкорпоративном уровне.

#### ЛИТЕРАТУРА

1. А. С. Казарицкий, Ю. Е. Мороховец «Применение модели асинхронных вычислений в программных системах распределенной обработки данных», [Электронный ресурс]: «Вычислительные сети: теория и практика». URL: <http://network-journal.mpei.ac.ru/cgi-bin/main.pl?l=ru&n=5&pa=4&ar=1> (дата обращения: 22.02.2019).
2. Ю. А. Тимошин, Т. Г. Шемсединов, В. П. Ярченко, А. И. Мороз «Технология распределенной обработки данных и приложений с использованием динамически интерпретируемых метамоделей» [Электронный ресурс]: Docplayer.ru, URL: <http://docplayer.ru/46645432-Udk-681-3-621-3-62-52-yu-a-timoshin-t-g-shemsedinov-v-p-yarchenko-a-i-moroz.html> (дата обращения: 22.02.2019).
3. Параллелизм против многопоточности против асинхронного программирования: разъяснение, [Электронный ресурс]: HABR.COM, URL: <https://habr.com/ru/post/337528/> (дата обращения: 22.02.2019).

© Юдин Дмитрий Сергеевич ( [order@echogroup.ru](mailto:order@echogroup.ru) ).

Журнал «Современная наука: актуальные проблемы теории и практики»