

Корпорация InterSystems  
Информационный документ

**СУБД INTERSYSTEMS CACHE  
КАК АЛЬТЕРНАТИВА  
БАЗАМ ДАННЫХ  
В ОПЕРАТИВНОЙ ПАМЯТИ**

## Оглавление

Вступление	1
Преимущества постоянного хранения данных	2
Прозрачный доступ через SQL и доступ к объектным данным	2
Протокол ECP	3
Принцип «одна машина – одна кэш-память»	3
Заполнение кэш-памяти	5
Заключение	5
Информация о корпорации InterSystems	6

## Вступление

Чтобы преодолеть ограничения в производительности традиционных реляционных баз данных и широкого спектра приложений – от установленных лишь на одном компьютере до функционирующих в крупной распределенной информационной среде, – нередко для ускорения доступа к данным применяются базы данных в оперативной памяти (in-memory databases). Хотя БД в оперативной памяти повышают производительность работы с данными, им свойственны некоторые недостатки, в частности: отсутствие поддержки больших по объему наборов данных, слишком высокие требования к аппаратному обеспечению, ограничения по масштабируемости.

InterSystems Caché® – высокопроизводительная объектная система управления базами данных с уникальной архитектурой, обеспечивающей пригодность этой СУБД для обслуживания приложений, в которых обычно используются базы данных в оперативной памяти. Уровень производительности Caché сопоставим с производительностью БД в оперативной памяти, при этом Caché, кроме прочего, обеспечивает:

- долговременное хранение – данные не теряются в случае отключения питания или поломки компьютера;
- быстрый доступ к очень большим наборам данных;
- масштабирование для развертывания на сотнях компьютеров и доступности для десятков тысяч пользователей;
- одновременный доступ к данным через SQL и объекты (объектные интерфейсы к Java, C++, .Net)

Далее объясним, почему именно Caché является привлекательной альтернативой in-memory БД для компаний, которым требуется высокоскоростной доступ к крупным массивам данных.

### ***Уникальный "движок данных" позволяет поддерживать долговременное хранение и согласованность данных и обеспечивает высокую производительность***

СУБД Caché – технологический продукт, принадлежащий к типу «persistent database», это означает, что данные из оперативной памяти записываются на диск в фоновом режиме. Может возникнуть вопрос: как тогда Caché может обеспечить уровень производительности, сопоставимый с БД в оперативной памяти, где данные лишь время от времени записываются в постоянное хранилище.

Частично ответ заключается в уникальных характеристиках архитектуры Caché. Вместо строк и колонок традиционной базы данных в Caché применяются многомерные массивы, чья структура основана на определениях объектов. Данные хранятся таким способом, каким его спроектирует специалист по архитектуре, и те же структуры, что используются в кэш-памяти in-memory БД, используются и в InterSystems Caché, но только на диске.

В СУБД Caché данные, относящиеся к единой структуре, и хранятся так же в единой структуре. Благодаря этому Caché обеспечивает исключительно быстрый доступ к данным на диске.

Кроме того, производительность многих продуктов с распределенной

кэш-памятью снижается из-за необходимости выполнения в in-memory БД синхронизаций при обновлении данных. В СУБД Caché процессы обновления данных и их распределения по кэш-блокам логически отделены друг от друга. Результат – высокая производительность благодаря сведению процессов к обработке гораздо более простого потока задач. Caché также обеспечивает внутривещное связывание с языками программирования C++ и Java, что позволяет приложениям, написанным на этих языках, напрямую работать со структурами данных и объектами Caché.

### Преимущества постоянного хранения данных

При производительности, сопоставимой с базами данных in-memory БД, Caché обладает рядом существенных преимуществ перед ними благодаря возможности предоставить доступ к данным, хранимым на диске. Наиболее очевидное преимущество – отсутствие необходимости в отдельном постоянном хранилище данных. Caché сама является постоянным хранилищем данных, которое всегда находится в актуальном состоянии. Данные не теряются в случае отключения компьютера или выхода его из строя.

Еще одно преимущество состоит в том, что при использовании Caché размер наборов данных не ограничен объемом доступной оперативной памяти. Если данные не находятся в локальном кэше, они непрерывно подкачиваются туда с диска. Благодаря отсутствию зависимости от объема оперативной памяти, основанная на Caché система может обрабатывать квадриллионы байт данных, чего не в состоянии обеспечить БД в оперативной памяти.

Добавление оперативной памяти для повышения общей производительности системы – более дорогостоящее мероприятие, чем добавление дискового пространства для хранения данных. (1 терабайт дискового пространства дешевле 1 терабайта RAM). К тому же, многие системы, основанные на БД в оперативной памяти, должны хранить дополнительные копии данных на отдельных компьютерах в целях резервирования для защиты от последствий в случае выхода компьютера из строя. Следовательно, применение систем с распределенной кэш-памятью и БД типа «persistent database», как система Caché, сопряжено с меньшими расходами на аппаратное обеспечение.

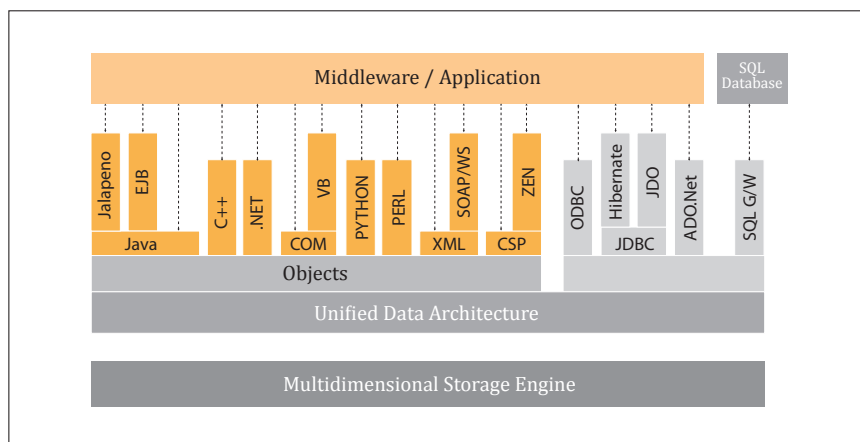
### Прозрачный доступ через SQL и доступ к объектным данным

Одна из проблем, общих для большинства БД в оперативной памяти, состоит в том, что, поскольку структуры их данных оптимизируются для обеспечения высокоскоростной обработки, данные, как правило, не сразу доступны через язык запросов SQL. Для совместимости с большинством аналитических и отчетных инструментов данные должны быть сначала отображены на реляционные таблицы. Обычно это делается тогда, когда данные переносятся из БД в оперативной памяти в постоянное хранилище, и, как правило, с выполнением трехфазного процесса «извлечения-преобразования-загрузки» (ETL). Дополнительные затраты, связанные с обработкой данных, и дополнительное время, затрачиваемое на их отображение на реляционных таблицах, – основная причина того, что реляционные БД недостаточно быстры для чрезвычайно скоростных распределенных приложений, и поэтому часто вместо них используются БД в оперативной памяти.

Лишь небольшое число хранящихся в оперативной памяти баз данных основаны на реляционной модели и предоставляют доступ к данным через SQL. При этом такие системы страдают от проблемы противоположного характера. Она состоит в том, что данные не сразу доступны для объектно-ориентированных технологий, которые обычно применяются для разработки приложений. Стоит добавить, что большинство баз данных в оперативной памяти не предназначены для поддержки многокомпьютерных конфигураций. Они работают только на одном компьютере и имеют ограничения в части зависимости от объема RAM.

Совсем другая ситуация в случае СУБД Caché, потому что используемые в ней многомерные массивы могут быть одновременно представлены и как

реляционные таблицы, и как объекты. Унифицированная архитектура данных СУБД Caché (Caché Unified Data Architecture™) обеспечивает как объектное, так и реляционное отображение данных в любое время – без необходимости маппирования.



<i>Middleware / Application</i>	Промежуточное ПО / Приложение
<i>SQL Database</i>	База данных SQL
<i>Objects</i>	Объекты
<i>Unified Data Architecture</i>	Унифицированная архитектура данных
<i>Multidimensional Storage Engine</i>	Средство многомерного хранения

Рис. 1. Унифицированная архитектура данных СУБД Caché предоставляет различные способы доступа к данным

SQL-доступ в Caché совместим с ODBC, JDBC и ADO.Net. Помимо этого, СУБД Caché поддерживает связывание данных с объектно-ориентированными языками, включая Java, .NET и C++. В СУБД Caché обеспечивается полное отображение объектов, а также поддержка трех основополагающих для объектно-ориентированного подхода концепций: наследования, полиморфизма и инкапсуляции.

### Протокол ECP

В многокомпьютерных прикладных решениях СУБД Caché автоматически поддерживает распределенные кэш-блоки путем использования протокола Enterprise Cache Protocol™ (ECP).

С использованием протокола ECP экземпляры Caché могут быть настроены как серверы данных и/или серверы приложений. Каждый элемент данных принадлежит серверу данных. Серверы приложений понимают, где находятся данные, и поддерживают локальные кэши недавно использовавшихся данных. Если сервер приложений не может удовлетворить запросы из локального кэша, он запрашивает требуемые данные на удаленном сервере данных. С помощью протокола ECP выполняется автоматическое управление согласованием кэш-памяти.

Протокол ECP не требует изменения приложений: они просто работают со всей базой данных, как если бы она была локальной. В этом состоит главное отличие от тех систем распределенной кэш-памяти, в которых каждому приложению-клиенту до выполнения любых запросов нужно указать, какое подмножество данных ему нужно.

### Принцип «одна машина – одна кэш-память»

Еще одно различие между СУБД Caché и другими системами с распределенными кэш-памятями заключается в том, что большинство таких систем поддерживает отдельную кэш-память для каждого процесса, выполняемого на компьютере. Например, если на одном компьютере работает восемь

приложений-клиентов, ему нужно поддерживать восемь отдельных кэш-памятей.

В отличие от этого подхода, СУБД Caché поддерживает кэш-память в совместно используемой памяти и дает необходимую привязку, чтобы обеспечить доступ к нужным данным тем процессам, которые выполняются в их адресном пространстве памяти. Доступ к данным может быть предоставлен одновременно по TCP-based протоколам, таким как JDBC, посредством языковых привязок, а также – для достижения исключительно высокой производительности – путем привязок, которые позволяют приложениям напрямую манипулировать кэш-памятью.

Возможность использования несколькими приложениями-клиентами одной кэш-памяти предоставляет целый ряд преимуществ. Одно из них состоит в том, что система, дающая такую возможность, снижает требования к объему памяти. Чтобы обеспечить работу в тех случаях, когда отдельным клиентам (а это бывает довольно часто) требуется доступ к общим данным, системы с распределенной кэш-памятью вынуждены поддерживать множественные копии данных. А при использовании Caché достаточно поддерживать только одну копию данных для каждого компьютера.

Реализация принципа «одна машина – одна кэш-память» также сокращает объем сетевого ввода/вывода, что существенно, поскольку в случае высокопроизводительных приложений важным вопросом может оказаться поддержка сетевого трафика в контексте поддержания необходимой кэш-памяти. В системе, работающей по принципу «одна машина – одна кэш-память», нужно обновлять только одну кэш-память по мере изменения основных данных, а не выполнять обновления многочисленных кэшей.

Даже в случае применения компьютеров с многоядерными процессорами, система на базе Caché использует только одну общую область кэш-памяти для каждого компьютера, отсюда – уровень масштабируемости, значительно превосходящий способность расширяемости системы с распределенной кэш-памятью. К примеру, для системы на основе Caché, установленной на 250 компьютерах, каждый из которых оснащен 8-ядерным процессором, нужно всего лишь 250 кэш-памятей для «общения» друг с другом с целью поддержания взаимной когерентности. Для сравнения, системам с разделенными кэш-памятями для каждого ядра требуется обеспечить согласованность уже между 2000 кэш-памятей. При том, что процессоры современных компьютеров могут иметь восемь, шестнадцать и более ядер, преимущество Caché становится еще более очевидным.

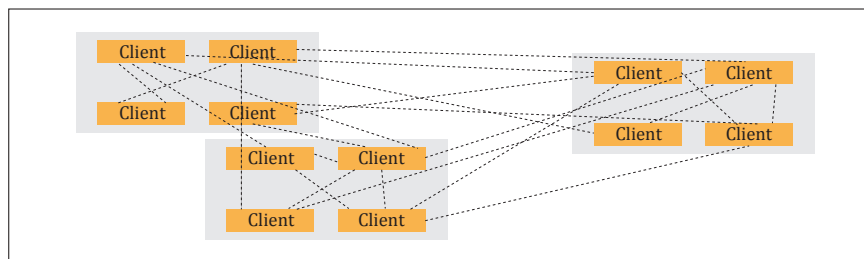


Рис. 2а. Обеспечение когерентности кэш-памятей в системах, где не применяется Enterprise Cache Protocol

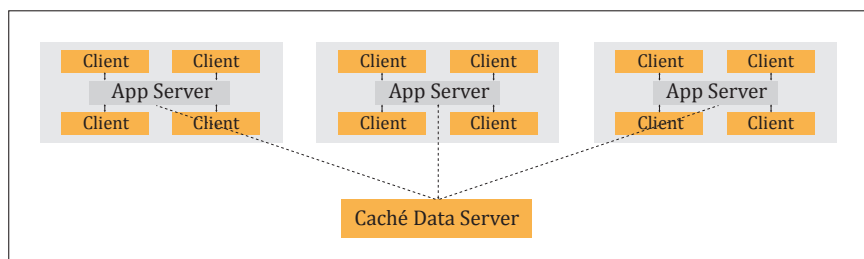


Рис. 2б. Обеспечение когерентности кэш-памятей в системе на базе технологии Caché.

## Заполнение кэш-памяти

Во многих приложениях с распределенной кэш-памятью ее предзагрузка может оказаться весьма небыстрым процессом. Причиной тому является уже сам объем имеющихся данных и/или время, которое требуется для отображения данных, содержащихся в хранилищах реляционных структур, на объектно-ориентированные структуры, используемые приложением. Если мы возьмем «нагруженные» данными приложения, то в этом случае заполнение оперативной памяти занимает больше времени, чем выполнение соответствующих вычислений.

При применении технологии Caché все обстоит по-другому. Благодаря обеспечению исключительно простого SQL-доступа к данным, технология Caché дает возможность «вытягивать» данные из основных исходных реляционных структур. И конечно же, Caché может, как БД типа «persistent database», быть базовым источником данных. При этом нет необходимости в предварительной загрузке кэш-памятей. Локальные кэш-памяти автоматически загружают те данные, которые им требуются в процессе исполнения запросов.

Также немаловажен вопрос, сколько компьютеров участвует в выполнении задачи заполнения кэш-памятей. В случае применения СУБД Caché необходимые данные поддерживаются небольшим числом компьютеров, включенных в распределенную среду. Все, что требуется для заполнения такой среды данными, – это доступ по протоколу ECP к серверам данных. Они могут быть загружены в фоновом режиме, не затрагивая другие компьютеры, используемые в это время для выполнения других задач. Когда серверы включаются в рабочий процесс, их кэш-памяти автоматически перезаполняются данными из запрошенных.

В большинстве систем, основанных на БД в оперативной памяти, все происходит совсем не так: при загрузке данных в такие системы они разделяются для распределения по отдельным кэш-памятям, чтобы все или почти все данные оказались в памяти, по крайней мере, одного компьютера. Результатом является то, что зачастую задача осуществления загрузок данных может стать невыполнимой для охвата всех входящих во взаимосвязанную среду компьютеров при ограниченном подмножестве активных компьютеров.

## Заключение

Главная причина, по которой часто выбирают базы данных в оперативной памяти, – это скорость. Но, хотя такие базы данных и быстры, они имеют и ряд заметных недостатков, среди которых: невысокая масштабируемость, отсутствие поддержки SQL-доступа к данным, слишком высокие требования к аппаратному обеспечению, риск потери данных в результате незапланированного сбоя в работе вычислительного оборудования.

InterSystems Caché – единственная СУБД, обеспечивающая долговременное хранение данных и демонстрирующая производительность равную производительности баз данных в оперативной памяти. InterSystems Caché также позволяет поддерживать очень крупные массивы данных, предоставляет SQL-доступ к ним, а также обеспечивает доступ к объектам с использованием объектно-ориентированных языков, может обслужить распределенные системы из сотен взаимосвязанных компьютеров и отличается исключительной надежностью.

Все это по праву делает InterSystems Caché предпочтительной для выбора с целью использования вместе с приложениями, применяемыми для высокоскоростной обработки очень крупных объемов данных.

## Информация о корпорации InterSystems

Корпорация InterSystems – один из ведущих в мире поставщиков ИТ-платформ. Корпорация имеет головной офис в Кембридже, штат Массачусетс, США, и зарубежные представительства в 23 странах. InterSystems производит инновационные информационно-технологические продукты для быстрой разработки, развертывания и интеграции корпоративных приложений.

**InterSystems Caché®** – высокопроизводительная объектная база данных, обеспечивающая ускоренную работу приложений и возможность их широкого масштабирования.

**InterSystems Ensemble®** – технологическая платформа для быстрой интеграции и разработки приложений, позволяющая обогатить их новыми функциональными возможностями и повысить уровень их соединяемости во взаимно-связанных информационных средах.

**InterSystems HealthShare™** – ИТ-платформа, которая позволяет быстро создавать на ее основе системы электронной медицинской карты для обмена медицинской информацией на региональном и общенациональном уровне.

**InterSystems DeepSee™** – программное обеспечение, позволяющее встроить бизнес-анализ в транзакционные приложения для принятия эффективных операционных решений.

The logo for InterSystems, featuring the word "INTERSYSTEMS" in a stylized, blue, serif font. The letters are closely spaced and have a slightly irregular, hand-drawn appearance.