Управление физическими активами в условиях эпизодической (off-line) связи

А.Ю. Молчанов (НПП «СпецТек»)

Информационная система управления физическими активами не может быть построена без использования технологий распределенных вычислений. Специфика объекта управления такова, что зачастую необходимо использовать распределенные вычисления в отсутствие прямой связи в реальном времени между компонентами системы. Представлен обзор соответствующих технологий распределенных вычислений.

Ключевые слова: управление физическими активами, EAM, распределенные вычисления, off-line технологии.

Задачи управления физическими активами наиболее емко сформулированы в международных стандартах ISO серии 55000 и государственных стандартах серии ГОСТ Р 55.0.00 [1]. Под физическими активами в стандартах понимается основное производственное и вспомогательное оборудование, объекты инфраструктуры (транспорт, здания, сооружения, передаточные устройства, системы связи), материальные запасы (запчасти, материалы для ТОиР), контрольно-измерительные приборы и иные объекты организации.

Конечной целью задачи управления физическими активами является нахождение оптимального баланса производительности, затрат и рисков, связанных с активами [2]. Следует отметить, что участники этих процессов часто находятся в разных подразделениях, на разных уровнях иерархии управления, выполняют разные функции (финансовые, производственные, ремонтные). При этом их деятельность по управлению физическими активами должна быть направлена на достижение единых целей и взаимно согласована.

Поэтому информационная система управления физическими активами (далее по тексту — ИС) должна охватывать всю структуру организации, включая удаленные филиалы, дочерние, зависимые и иные организации, которые совместно участвуют в управлении одними и теми же активами. Построение такой ИС невозможно без использования распределенных вычислений.

Причины и условия использования off-line технологий

Существуют различные технологии организации распределенных вычислений, многие из которых широко применяются в системах управления физическими активами [3, 4].

Основное внимание уделяется on-line технологиям, ориентированным на наличие непрерывной связи между компонентами ИС, обеспечивающей их взаимодействие в реальном масштабе времени. Все подобные современные ИС построены с использованием как минимум одной такой технологии, а зачастую и комбинации нескольких [3]. Справедливо и то, что рост производительности каналов связи делает on-line технологии все более доступными и существенно снижает их стоимость.

И все же для ИС управления физическими активами остаются актуальными off-line технологии, предполагающие отсутствие непрерывной связи между компонентами ИС. Причины этого следующие:

- производственный процесс может предусматривать длительное автономное функционирование активов, когда постоянная связь с ними технически невозможна или требует значительных затрат (например, суда, находящиеся в плавании, или горнопроходческие комплексы, функционирующие в шахтных выработках);
- процессы управления могут быть таковы, что их оперативное выполнение при прямом соединении с ИС физически невозможно или осложнено требованиями производственно-технологических процессов (чаще всего это относится к бригадам оперативного персонала, оснащенным мобильными устройствами);
- в составе компании имеются филиалы или подразделения, деятельность которых не должна прерываться при потере прямой связи с центральным офисом (а как известно, любые, даже самые надежные каналы связи могут иногда выходить из строя);
- затраты на организацию единого централизованного сервера ИС могут оказаться существенно выше, чем затраты на создание нескольких распределенных вычислителей, взаимодействующих между собой, при этом его производительность может оказаться недостаточной для взаимодействия со всеми компонентами ИС в реальном времени.

Можно ожидать роста спроса на of-line технологии по мере развития и распространения мобильных устройств. Поэтому уделим внимание именно технологиям организации распределенных вычислений в режиме off-line. И надо заметить, что если без on-line технологий не обходится ни одна современная ИС управления физическими активами, то off-line технологии используются далеко не во всех таких ИС. Причина в том, что их использование требует включения

в ИС дополнительных компонентов и решения ряда проблем, которые не возникают при организации распределенных вычислений в режиме on-line.

ИС, использующие off-line технологии, предполагают только эпизодический обмен данными между компонентами системы. Принципиальная разница заключается в том, что при использовании on-line технологий любые данные, порожденные, полученные или измененные в любом компоненте ИС, в то же самое время (хотя и не мгновенно) становятся доступными всем остальным ее компонентам, которые имеют доступ к этим данным. При использовании off-line технологий эти данные становятся доступны другими компонентам не сразу, а только по истечении времени обмена данными.

Причем время обмена данными сопоставимо со временем выполнения основных функций ИС, и складывается из двух частей: времени на передачу данных и времени ожидания тех моментов, когда такая передача возможна. Обмен данными может происходить в заданные моменты времени, по команде пользователя, при наступлении какого-либо события или соблюдении каких-то условий. Важно отметить, что при использовании off-line технологий, как правило, ИС и ее отдельные компоненты не приостанавливают функционирование на время ожидания завершения обмена данными.

Технологии off-line могут использоваться в ИС на постоянной основе или эпизодически, когда в этом возникает необходимость. Интересные результаты достигаются, когда принципы организации распределенных вычислений по технологиям off-line применяются в ИС при фактическом наличии on-line связи: обмен данными между компонентами ИС идет фактически в реальном масштабе времени, но асинхронно с выполнением основных функций системы.

Основные архитектурные решения для построения off-line систем

Далее компоненты ИС, обмен данными между которыми происходит по технологиям off-line, будем называть узлами, чтобы отличать их от компонентов ИС, обменивающихся данными в режиме on-line. В качестве узлов могут выступать как отдельные компьютеры или мобильные устройства, так и целые подразделения, филиалы, удаленные отделы компании, содержащие множество взаимосвязанных устройств. Таким образом, каждый узел может состоять из множества взаимосвязанных компонентов, важно только, чтобы между ними существовала on-line связь.

На каждом из таких узлов должно выполняться программное обеспечение (ПО). Основу ИС управления физическими активами, как правило, составляет ПО класса EAM или ERP [5]. Для ИС, использующей off-line технологии организации распределенных вычислений, важно, чтобы ПО поддерживало соответствующие технологии обмена данными, а также было доступно для максимально широкого круга устройств, так как узлами системы могут быть не только компьютерные сети и отдельные компьютеры, но и различные мобильные устройства.

Каждый узел такой ИС содержит свою базу данных (БД), которая может быть локальной БД, если узел представляет собой отдельный компьютер или мобильное устройство, либо выделенной БД на сервере данных, если узел объединяет несколько устройств. Тогда единая БД всей ИС будет включать в себя все БД, располагающиеся на всех узлах в составе всей информационной системы. В этом случае можно говорить о распределенной БД. Пример ее реализации мы видим в EAM-системе TRIM [6].

Для ИС, использующей off-line технологии, необходимо определить архитектурное решение, на основе которого она будет строиться. Выбор архитектурного решения определяет состав входящих в ИС узлов, структуру и взаимосвязи между ними. Поэтому важно определить критерии выбора. В качестве основных используются следующие критерии:

- затраты на организацию ИС (всех узлов и каналов связи между ними) с учетом как финансовых (стоимость), так и технических аспектов;
- надежность функционирования ИС, в том числе в зависимости от надежности каналов связи и отдельных узлов;

Не существует единого для всех ИС архитектурного решения, поэтому далее рассмотрим типичные варианты.

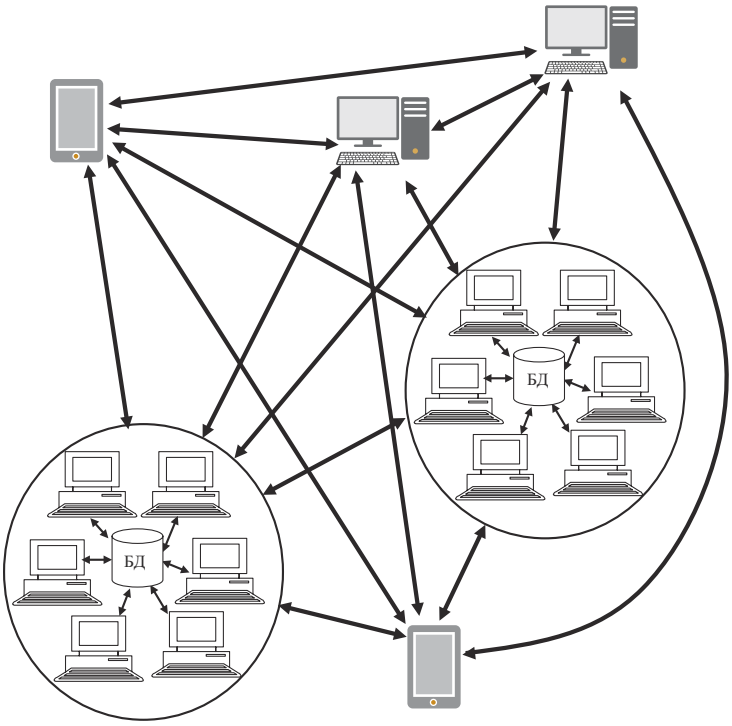

Рис. 1. Пример архитектуры полносвязной сети, состоящей из 5 разнородных узлов ИС

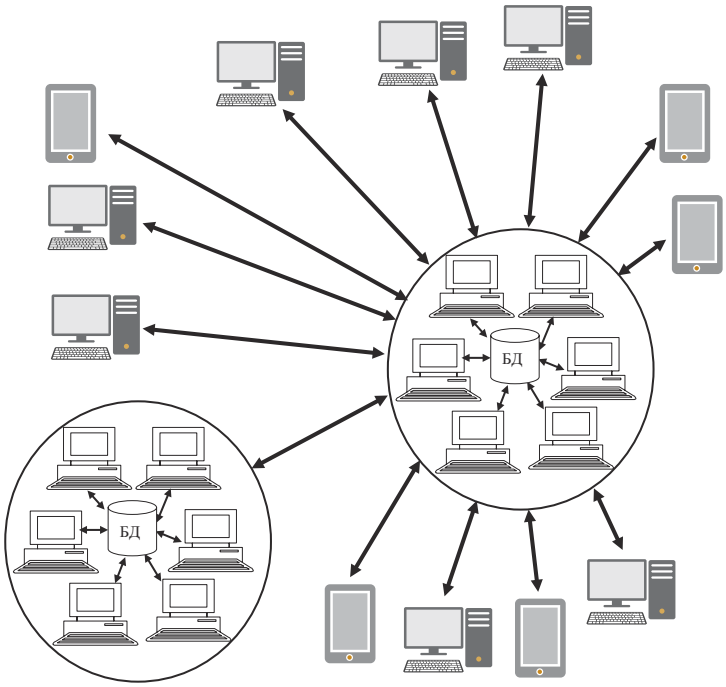

Рис. 2. Пример архитектуры типа «звезда», состоящей из 14 разнородных узлов ИС

Наиболее универсальным решением для построения распределенной ИС является полносвязная сеть. В этом случае ИС состоит из множества одноранговых компонентов, взаимосвязанных между собой по принципу «каждый с каждым». Применительно к использованию off-line технологий это означает, что ИС состоит из множества узлов, каждый из которых имеет каналы связи со всеми другими узлами. Пример структуры такой ИС приведен на рис. 1.

Архитектура полносвязной сети позволяет предельно упростить обмен данными между узлами: как только на узле появляются данные, которые необходимо передать на другой узел, он задействует соответствующий канал связи. Однако при этом возникают проблемы с управлением потоками данных, так как каждый узел сам должен определить, куда и какие данные передавать. Главной же сложностью является организация каналов связи всех со всеми. Поэтому ИС, построенные на основе такой архитектуры, встречаются крайне редко.

Наиболее универсальным архитектурным решением является архитектура типа «звезда»: в составе ИС выделяется один центральный узел, а все остальные узлы системы должны иметь каналы связи только с ним. При необходимости передать данные любой узел в такой системе (кроме центрального) передает их по каналу связи с центральным узлом. Центральный узел обладает всей полнотой информации и, исходя из этого, самостоятельно определяет, каким узлам передавать те или иные данные. Пример структуры такой ИС приведен на рис. 2.

Архитектура организации распределенных off-line вычислений типа «звезда» наиболее проста в реализации, так как предусматривает создание каналов связи только с центральным узлом и удобна с точки зрения управляемости, поскольку все правила и права доступа к данным также задаются на центральном узле. Но отсюда же следует и ее основной недостаток — при потере связи с центральным узлом или при выходе его из строя все узлы лишаются возможности передавать и получать данные. Кроме того, с ростом числа узлов центральный узел становится узким местом ИС и его производительность является ключевым фактором для функционирования всей системы в целом.

Тем не менее, при общем числе узлов до нескольких десятков и при должной организации поддержки функционирования центрального узла такая архитектура имеет реальное практическое применение.

Развитием архитектуры типа «звезда» является древовидная архитектура. В этом случае сеть взаимосвязей между узлами представляет собой древовидный граф, в корне которого находится центральный узел ИС, имеющий каналы связи со всеми узлами первого уровня. Каждый узел первого уровня, в свою очередь, имеет каналы связи с относящимися к нему узлами второго уровня (если они есть) и т.д. В результате каждый узел в такой системе имеет связь с одним вышесто-

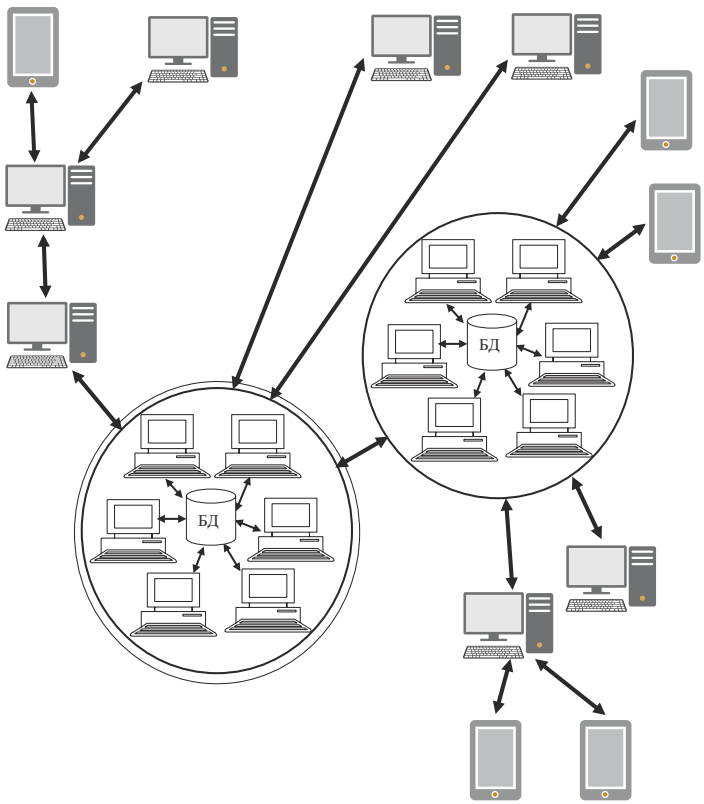

Рис. 3. Пример древовидной архитектуры из 14 узлов ИС, имеющей 3 уровня иерархии

ящим узлом (кроме центрального корневого узла) и несколькими нижестоящими узлами (если они есть). Пример структуры такой ИС приведен на рис. 3. При этом функции промежуточных узлов не возлагаются на мобильные устройства.

Для передачи данных в такой архитектуре необходимо знать, где находится узел-получатель данных. Если он расположен среди нижестоящих узлов, то передача данных должна происходить по каналу связи с тем узлом, в иерархию которого входит получатель данных. Иначе передача данных должна осуществляться на вышестоящий узел, и далее уже он, получив данные, будет определять их адресата по такому же алгоритму. Если данные предназначены всем узлам, отправитель должен направить их и нижестоящим узлам и вышестоящему узлу.

Преимуществом такой организации распределенных off-line вычислений по сравнению с архитектурой типа «звезда» является более высокая надежность системы (в случае выхода из строя канала связи с вышестоящим узлом или самого этого узла возможность обмена данными с нижестоящими узлами сохраняется), а также снижение нагрузки на центральный корневой узел (для обмена данными в пределах веток древовидной структуры нет необходимости задействовать вышестоящие узлы). При этом администрирование и управление потоками данных по-прежнему можно возложить на центральный узел. Платой является снижение скорости обмена данными между узлами, находящимися в разных ветвях дерева.

Для ИС управления физическими активами, применяющих off-line технологии распределенных вычислений, желательно иметь возможность выбора архитектуры из нескольких доступных. Это позволит более гибко строить ИС и настраивать ее функциональность в зависимости от потребностей заказчика. Например, в EAM-системе TRIM [6] для организации распределенных вычислений в режиме off-line основной является архитектура типа «звезда», однако для сложных систем с большим числом узлов возможно использование древовидной архитектуры.

Принципы взаимодействия узлов off-line систем

Вторым важным вопросом является организация взаимодействия узлов ИС. Следует определить принципы синхронизации данных между узлами распределенной ИС в режиме off-line. Для этого необходимо разработать протокол или стек протоколов взаимодействия узлов ИС, в которых определить, какие данные будут передаваться, в каком формате и по каким правилам.

Такое взаимодействие должно быть организовано на основе нескольких основополагающих принципов. Во-первых, протокол обмена данными не должен зависеть от типа канала между узлами, так как заранее не известно, какой способ обмена данными между ними будет доступен. Это может быть и канал прямого обмена через глобальную или локальную сеть, и обмен пакетами через электронную почту или файлами по одному из возможных протоколов (например, FTP — File Transfer Protocol, протокол передачи файлов). В предельном случае этот канал может отсутствовать, и тогда данные между узлами должны передаваться путем обмена сменными носителями. Во-вторых, формат обмена данными не должен зависеть от аппаратных требований, поскольку в состав системы могут входить совершенно разные устройства и сети устройств. В-третьих, объем данных, передаваемых на каждый узел сети, должен регулироваться потребностями этого узла и решаемыми на нем задачами. Это исключит дублирование всех данных по всей сети, замусоривание узловых БД излишними данными и перегрузку каналов связи между узлами. Наконец, должна быть предусмотрена процедура актуализации данных с учетом времени и места их происхождения и маршрута передачи по всей сети.

Разнообразие протоколов обмена данными в различных ИС с использованием off-line технологий велико, но все их можно объединить в две основные группы:

- протоколы обмена данными;
- протоколы обмена командами (функциями).

Протоколы обмена данными предполагают, что между узлами передаются непосредственно сами данные. В этом случае передающий узел формирует их для передачи, а принимающий узел получает их и вносит в свою БД. Формально это очень похоже на файловый обмен при использовании распределенных вычислений на основе технологий on-line [4]. Протоколы обмена данными достаточно просто реализуются, но главным их недостатком является большой объем передачи данных при их массовом изменении на одном из узлов ИС.

Протоколы обмена командами предполагают, что для синхронизации информации между узлами передаются команды обработки данных. Тогда передающий узел, выполняя команду обработки данных, формирует пакет обмена данными, содержащий всю информацию об этой команде, и передает его на узел-получатель. Получатель, приняв команду, выполняет ее точно так же, как и узел-отправитель. При выполнении одной и той же команды с одинаковыми исходными данными должен получаться одинаковый результат — таким образом, данные на узлах синхронизируются. Формально этот механизм похож на взаимодействие серверной и клиентской частей системы типа «клиент-сервер» при использовании распределенных вычислений в режиме on-line [4]. В этом случае объем передаваемых данных при массовой обработке данных может существенно сокращаться, но важным условием реализации этого механизма является обеспечение одинакового контекста выполнения команд на разных узлах, что делает его более сложным в реализации, чем обмен данными, рассмотренный выше.

Как правило, при обмене данными или командами в ИС, использующих off-line технологии органи-

зации распределенных вычислений, эти данные или команды разбиваются на пакеты для обмена. Состав и структура пакетов зависит от функциональности системы и состава передаваемых данных или команд. А как уже было сказано, форматы обмена данными не должны зависеть от требований конкретных программно-аппаратных компонентов. Поэтому в основу формирования пакетов чаще всего закладываются текстовые форматы. Наиболее распространенные из них строятся на основе языка гипертекстовой разметки XML (eXtensible Markup Language — расширяемый язык текстовой разметки) или на основе текстовой структуры JSON (Java Script Object Notification — описание объектов в формате языка Java Script).

Еще одним важным вопросом является выделение данных для обмена из БД узла. Каждый узел ИС, инициирующий обмен данными, должен определять, какие данные или команды и в каком составе необходимо передавать на другие узлы. Логика обмена данными напрямую зависит от администрирования потоков данных и от выбранного архитектурного решения. Но в любом случае узел должен выделять в своей БД актуальные данные, которые подлежат передаче на другие узлы.

Чаще всего выделение таких данных обеспечивается по метке времени изменения записей в БД: на узле должна быть информация, когда был последний обмен данными с каждым взаимосвязанным узлом, и тогда передаче на этот узел могут подлежать только те данные, у которых будет более поздняя метка последнего изменения.

Следующий вопрос, связанный с использованием off-line технологий, — сложности с удалением данных из БД узла: любая запись может быть окончательно удалена из БД узла только после того, как информация о ее удалении будет передана на все взаимосвязанные узлы. Рассмотренные проблемы являются иллюстрацией того, что ИС, использующая off-line технологии распределенных вычислений, изначально должна быть спроектирована с учетом всех нюансов их использования, а БД каждого узла должна быть построена с учетом того, что она будет частью распределенной БД.

Обмен информацией в режиме off-line не гарантирует, что пакеты с данными будут приняты в том же порядке, в каком они были переданы. Произвольный порядок обработки принятых пакетов может привести к рассинхронизации данных на узлах: например, в какое-то поле на передающем узле было внесено значение N, а позже — значение M. Если принимающий узел сначала обработает пакет, содержащий M, а потом содержащий N, то значение этого поля на двух узлах разойдется. Как вариант, возможно снабжение каждой порции данных универсальной меткой времени, содержащей информацию о точном времени их создания. Тогда на принимающем узле можно отклонять и не обрабатывать те пакеты, аналоги которых уже были получены с более поздней меткой времени. Однако это не всегда гарантирует от несогласованности данных на узлах. Например, если на передающем узле сначала была создана запись, а потом она была удалена, то при нарушении порядка пакетов на принимающем узле могут быть проблемы: если на принимающий узел сначала придет пакет с данными об удалении записи, он не сможет быть обработан, так как на этом узле данной записи еще нет. А если потом этот же узел получит предыдущий пакет о создании записи, то может добавить в свои данные новую запись, которая на узле-отправителе была удалена.

Поэтому зачастую для соблюдения порядка обработки пакетов на узле-получателе в состав каждого пакета на узле-отправителе добавляется порядковый номер пакета, отсчитываемый для каждого получателя. Тогда получатель не обрабатывает принятый пакет, пока не будет принят и обработан пакет с предыдущим номером. В этой связи возникает вопрос о гарантированной доставке пакетов: если какой-то пакет с данными будет потерян в процессе передачи, то весь обмен данными застопорится, получатель будет ожидать потерянный пакет и откладывать все остальные пакеты. Как правило, эта проблема решается за счет использования механизмов квитанций и интервалов повторной передачи (аналогичная схема применяется для транспортного протокола TCP в известном стеке протоколов TCP/IP сети Internet).

Еще одна задача, которую необходимо решить — это правила сопоставления на каждом узле записей, полученных от других узлов ИС, с записями, уже имеющимися в БД данного узла. Принимая каждую запись данных, узел-получатель должен решать: добавлять ли ее в БД как новую, либо же обновлять уже существующие в БД данные на основе полученных. Эта проблема решается за счет использования в БД уникальных ключей. Для ИС, построенных на основе on-line технологий, существуют механизмы, гарантирующие уникальность и непротиворечивость таких ключей при одновременном доступе различных компонент к БД. Однако они не могут быть использованы при организации распределенных вычислений на основе off-line технологий, так как БД различных узлов не имеют связи между собой в реальном масштабе времени.

В этом случае в системе должен быть предусмотрен механизм, обеспечивающий уникальную идентификацию каждой записи в пределах всей распределенной БД. Одним из вариантов такой идентификации является использование GUID (Globally Unique Identifier, глобальный уникальный идентификатор — статистически уникальный 128-битный идентификатор). Однако применение GUID осложнено необходимостью обработки и хранения 128-битных данных. Другим вариантом уникальной идентификации записей является использование пары идентификаторов — идентификатор узла + идентификатор записи, уникальность которой будет гарантирована, если каждый узел в пределах ИС будет иметь присвоенный ему уникальный

идентификатор, а для порождения идентификаторов записей в пределах узла будут применяться те же механизмы, которые используются в ИС, построенных на основе on-line технологий.

При использовании off-line технологий для организации распределенных вычислений в ИС неизбежно возникает и проблема конфликтов между данными. Поскольку прямая связь между узлами ИС отсутствует, а время обмена данными между ними сопоставимо с временем выполнения функций системы, не исключены ситуации, когда одни и те же данные будут изменены на разных узлах системы в пересекающиеся интервалы времени. Тогда в момент обмена данными между узлами возникнет конфликт, связанный с тем, какие изменения данных следует принять, а какие — отвергнуть?

Поэтому должны быть предусмотрены правила разрешения подобных конфликтов. Эти правила могут быть различны и достаточно сложны, а их применение должно зависеть от логики системы, архитектурного решения и состава обрабатываемых данных. Но чаще всего в основе этих правил лежат два основных принципа: принцип метки времени и принцип приоритетности (а также их возможные комбинации). В первом случае данные, имеющие более позднюю метку времени, принимаются, а данные, имеющие более раннюю метку — отклоняются. Во втором случае принимаются данные, измененные на узле, который считается более приоритетным (как правило, это вышестоящий по иерархии узел), а данные менее приоритетного узла отклоняются.

Существующие ИС управления физическими активами, построенные на основе off-line технологий организации распределенных вычислений, используют различные способы решения обозначенных выше проблем. Зачастую одна и та же ИС может предложить пользователям разные варианты решений в зависимости от конкретной ситуации. Например, в EAM-системе TRIM [6] применяются методы синхронизации информации, основанные и на обмене данными, и на обмене командами, а для решения конфликтов данных используются сложные правила.

Заключение

Современная ИС управления физическими активами должна предполагать возможность организации распределенных вычислений без непрерывной связи в режиме реального времени (в режиме off-line). Такой способ предполагает, что клиентское рабочее место все равно запрашивает услуги серверной части ИС, но делает это не в режиме реального времени. То есть любое клиентское рабочее место в таком режиме не выпадает из целостной системы и остается ее частью, но его взаимодействие с системой происходит на основе иных принципов, нежели в режиме непрерывного соединения.

Для грамотного выбора такой ИС заказчику необходимо учитывать многие указанные выше аспекты в тех случаях, когда предполагается использование распределенных вычислений в режиме off-line. При этом конфигурацию используемой ИС он должен определять с учетом упомянутых особенностей организации распределенных вычислений, а также исходя из своих возможностей и потребностей, принимая во внимание все имеющиеся у него производственные и организационно-технические условия.

Список литературы

1. *Хромова Н.А., Крюков И.Э., Пугачев В.М.* Стандарты ISO серии 55000 кодифицируют знания в области управления активами // Мир стандартов. 2014. № 2(83). С. 35-38.
2. *Иорш В.И.* Преимущества хорошего управления активами // Генеральный директор. Управление промышленным предприятием. 2015. №4. С. 18-25.
3. *Молчанов А.Ю.* Организация распределенных вычислений для управления физическими активами // Автоматизация в промышленности. 2017. №8. С. 23-28.
4. *Молчанов А.Ю.* Системное программное обеспечение. Учебник для вузов. 3-е изд. СПб.: Питер. 2010. 400 с.
5. *Данилов О., Скворцов Д., Свистула О.* Автоматизация ТОиР. Хроника внедрений // Главный энергетик. 2007. №7. С.11-16.
6. *Антоненко И.Н.* EAM-система TRIM: от автоматизации ТОиР к управлению активами // Автоматизация в промышленности. 2015. №1. С. 40-43.

***Молчанов Алексей Юрьевич** – канд. техн. наук, директор по разработкам*
ООО «НПП «СпецТек».
Контактный телефон +7 (812) 329-45-60.
E-mail: mill@spectec.ru