

УДК 629.78.018

И.Б. ТУРКИН, Е.В. СОКОЛОВА, П.А. ЛУЧШЕВ

Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина

АНАЛИЗ РЕЗУЛЬТАТОВ ЭКСПЕРИМЕНТАЛЬНОГО ИССЛЕДОВАНИЯ МЕХАНИЗМОВ ИНФОРМАЦИОННОГО ВЗАИМОДЕЙСТВИЯ НА ОСНОВЕ СПЕЦИФИКАЦИИ OLE FOR PROCESS CONTROL

Рассмотрены результаты экспериментальных исследований механизмов доступа к данным OPC-сервера. Приведена оценка затрат процессорного времени на обмен информацией между клиентом и сервером OPC, определена пропускная способность информационного обмена, даны выводы по использованию OPC в условиях мягкого реального времени с учетом детерминизма и реактивности системы.

OLE for Process Control, SCADA, спецификация OPC, OPC-сервер, OPC-клиент, АСУ ТП, унификация доступа к данным, пропускная способность

Введение

Основу программного обеспечения средств автоматизации технологических процессов современных производственных предприятий составляют SCADA-системы [1], которые осуществляют сбор и обработку данных, а также управление автоматизированной системой в целом. Данные, поступающие на вход SCADA-системы с аппаратуры нижнего уровня различных производителей, должны быть унифицированы. Для этого используется технология OPC (OLE for Process Control) [2], разработанная для работы под управлением ОС Windows.

Значительная часть программного обеспечения SCADA-систем характеризуется повышенными требованиями по функциональной безопасности и надежности, жесткими ограничениями на время реагирования на внешние воздействия [3, 4]. Как известно, к числу ключевых характеристик, определяющих возможность применения определенных программных средств при таких ограничениях, относятся:

- детерминизм – возможность системы отвечать на внешние события в специфицированное время;
- реактивность – свойство системы, определяемое ее способностью выдерживать минимальные интервалы времени между внешним событием и получением результата.

Актуальность исследований заключается в необходимости выбора наиболее эффективных по ключевым характеристикам моделей доступа OPC в зависимости от конкретных условий.

1. Общая постановка проблемы

Сложность современных программных средств достигла такого уровня, что даже разработчики классических систем реального времени рекомендуют использовать эмпирические методики для определения основных показателей [6].

Цель исследований – экспериментально оценить возможность применения технологий OPC в программном обеспечении АСУ ТП с повышенными требованиями по функциональной безопасности и надежности, жесткими ограничениями на время реагирования на внешние воздействия.

Для достижения этой цели необходимо:

- рассмотреть модели обмена информацией;
- определить критерии оценки пропускной способности;
- создать ПО для проведения эксперимента и измерения необходимых параметров;
- выполнить измерения;
- провести анализ результатов методами математической статистики.

2. Модели обмена информацией в спецификации OPC

Характеризуя механизмы информационного взаимодействия, рассмотрим модель функционирования потоков в ОС *Windows* [5], согласно которой поток может быть:

- приостановлен;
- заблокирован средствами синхронизации;
- готов к выполнению;
- в состоянии выполнения.

И клиент и сервер работают в различных процессах и соответственно потоках, а обмен данных выполняется синхронно, асинхронно или по подписке.

При *синхронном чтении* клиент посылает серверу запрос со списком интересующих его переменных и ждёт, когда сервер его выполнит. На рис. 1. показана упрощенная схема функционирования синхронного доступа к данным.

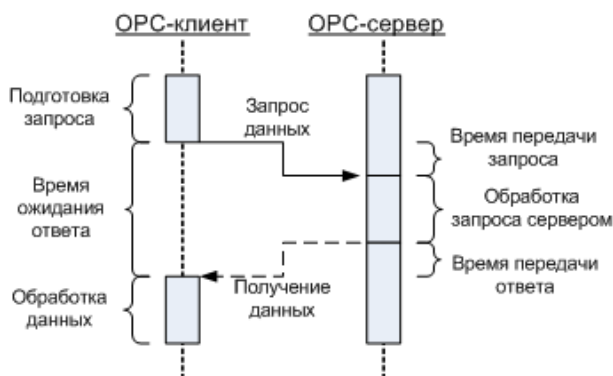


Рис. 1. Упрощенная схема функционирования синхронного доступа к данным

При *асинхронном чтении* клиент посылает серверу запрос, а сам продолжает работать. Когда сервер выполнил запрос, клиент получает данные. На рис. 2. показана упрощенная схема функционирования асинхронного доступа к данным.

При *методе подписки* клиент передает серверу список интересующих его переменных, а сервер затем регулярно присылает клиенту информацию об изменившихся переменных из этого списка. Эти списки в терминологии OPC называются группами. Каждый клиент может поддерживать одновременно

много групп с разной периодичностью обновления (рис. 3).

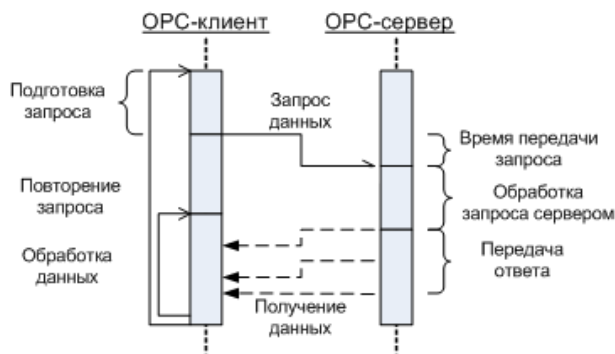


Рис. 2. Упрощенная схема функционирования асинхронного доступа к данным

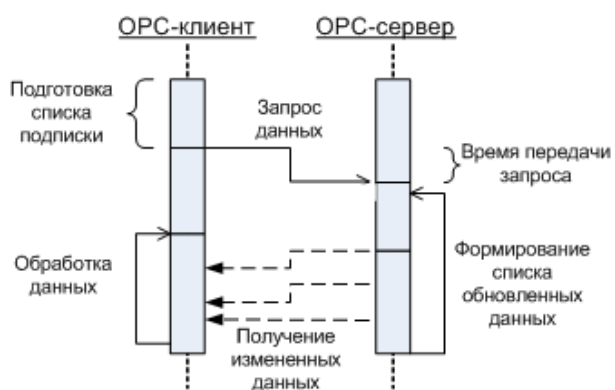


Рис. 3. Упрощенная схема функционирования доступа к данным – подписка

Помимо указания желаемой периодичности клиент может задать максимальное относительное изменение измеряемой величины, при превышении которого сервер должен оповестить клиента.

3. Характеристики пропускной способности клиент-серверного взаимодействия

Производительность OPC-сервера определяется особенностями его информационного взаимодействия с клиентами и контроллером, а также затратами времени на обработку информации в самом OPC-сервере.

Сервер, взаимодействуя с клиентом, выполняет несколько функций, поэтому можно говорить о пропускной способности каждой из них:

- создание/удаление связей с данными (то есть создание/удаление элементов *OPCItem*) в секунду;

- изменение состояния групп (количество изменений в секунду);
- изменение состояния связей с данными (количество изменений в секунду);
- синхронное чтение/запись (количество переменных, считываемых/записываемых в секунду в синхронном режиме);
- асинхронное чтение/запись (количество переменных, считываемых/записываемых в секунду в асинхронном режиме);
- обновление (количество обновлений через функцию *Refresh* в секунду).

Производительность в связке с контроллером определяется следующими параметрами:

- обновление данных (переменных). Это количество переменных, передаваемое от контроллера к серверу за секунду;
- реакция на запись данных. Время, которое проходит от момента подачи команды клиентом на запись до момента считывания клиентом полученного значения;
- скорость записи. Это количество переменных, которое можно записать в контроллер за секунду.

4. Планирование эксперимента

В результате анализа различных факторов, определяющих пропускную способность клиент-серверного взаимодействия, выделены наиболее актуальные и важные для исследования:

- количество передаваемых данных определяется геометрической прогрессией в диапазоне $1 \div 10^4$ тегов в группе;
- способ передачи данных: синхронное чтение/запись тегов в группе; синхронное чтение/запись всей группы; асинхронное чтение/запись группы.

В целом и *PC*-совместимая платформа, и ОС *Windows* слабо приспособлены для решения задач реального времени [7]. Зачастую приходится самостоятельно поддерживать механизмы, которые на

других платформах поддерживаются по умолчанию. Тем не менее, определенные средства измерения и контроля времени в ней имеются.

В ПО, созданном для проведения эксперимента, для измерения времени и затрат процессора используются функции *API QueryPerformanceCounter* и *QueryPerformanceFrequency*, которые обеспечивают разрешающую способность $\approx 10^{-7}$ с и, следовательно, являются удобным инструментом для измерения интервалов времени в диапазоне: $10^{-5} \div 10^{-2}$ с.

При больших интервалах времени следует учитывать рост погрешности измерений, вызванный затратами времени на работу ядра ОС. Для этого можно использовать функции *GetProcessTimes* или *GetThreadTimes*, которые возвращают временную информацию для указанного в аргументах процесса или потока соответственно. Разрешающая способность функций 10^{-2} с, максимальное значение для практических целей практически неограниченно, что свидетельствует о целесообразности использования данных функций для измерения затрат процессорного времени на работу достаточно продолжительных алгоритмов ($\geq 0,1$ с).

5. Анализ полученных результатов

Эксперименты были проведены с использованием *AMD Athlon XP 1700+*, *DDR SDRAM PC3200 512 MB* и *Fastwel UniOPC 2.41*. В результате получены серии по 50-300 измерений, к которым применены статистические методы оценки результатов измерений: расчет математического ожидания, дисперсии, среднеквадратичного отклонения, определение минимального и максимального значений в серии для оценки разброса измеренных параметров относительно среднего значения, построение гистограммы, определение доверительного интервала по Стьюденту.

Первичный анализ позволяет сделать следующие выводы:

1. Закон распределения случайной величины «время выполнения операции обмена» в общем случае является сложным, двухмодальным.

2. Основную составляющую вносит экспоненциальное распределение, имеющее четко выраженный максимум в районе минимума времени затрат времени на одну операцию.

3. В большинстве случаев наблюдается четко выраженный второй максимум с формой, напоминающей нормальный закон распределения. Скорее всего, этот максимум связан с работой диспетчера ОС. На практике от этого нежелательного максимума, ухудшающего детерминизм системы, можно избавиться, если использовать функции для управления работой диспетчера.

Затраты процессорного времени на обмен информацией в пересчете на один тег можно характеризуются следующими выводами.

1. При количестве тегов в группе менее 10 результаты недостоверны, ввиду недостаточной разрешающей способности функций для измерения затрат процессорного времени.

2. При решении практических задач нецелесообразно применение групп, включающих более 1000 тегов из-за резкого роста затрат процессорного времени на организацию обмена. Очевидно, этот рост объясняется неэффективностью работы механизмов ОС при обработке динамических массивов большой размерности, а ведь именно динамические массивы в виде сложной структуры *Variant* пересылаются по интерфейсу *OPC* между клиентом и сервером.

3. Во время операции обмена данными клиент и сервер проводят в режиме пользователя более 90% общего времени, что свидетельствует об общей управляемости системы и ее готовности отреагировать на внешние воздействия с должной оперативностью.

Зависимость времени передачи в расчете на один тег от объема передачи и способа его организации характеризуется следующими выводами.

1. Случайная составляющая погрешности не ве-

лика, что подтверждается и шириной «коридора», в который попали измерения, и расчетным СКО.

2. Операции записи выполняются быстрее.

3. При решении практических задач нецелесообразно применение групп, включающих более 1000 тегов из-за заметного увеличения времени, требуемого для передачи.

Характеризуя пропускную способность системы можно утверждать, что:

1. Пропускная способность на операциях записи в *OPC*-сервер заметно выше, чем на операциях чтения. С точки зрения практических потребностей более желательна обратная зависимость, так как выдача управляющих воздействий в систему осуществляется эпизодически, а сбор информации о ее состоянии – постоянно.

2. Операция асинхронного чтения становится быстрее синхронной на больших объемах информации (более 1 000 тегов). Чтение по одному тегу более эффективно при малых группах (менее 10 тегов). В диапазоне 10÷1000 тегов в группе наиболее быстрыми являются синхронные методы обмена, скорость при этом достигает 80 000 тегов в секунду.

На рис. 4. приведена зависимость пропускной способности (скорости передачи данных) от способа обмена и количества тегов в группе.

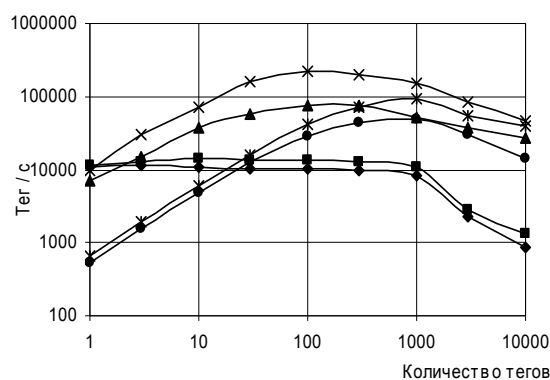


Рис. 4. Зависимость пропускной способности от способа обмена и количества тегов в группе:

—●— ReadItem —■— WriteItem —▲— SyncRead
—×— SyncWrite —*— AsyncRead —●— AsyncWrite

Характеризуя затраты процессорного времени на 1 тег, можно отметить, что наиболее экономичным является асинхронный способ обмена, если пренеб-

речь затратами времени в диспетчере ОС на переключение контекста потока и буферизацию сообщений. Данные затраты можно оценить только по косвенным признакам. Обмен информацией по 1 тегу является наиболее неэффективным способом.

На рис. 5. приведена зависимость суммарных затрат процессорного времени на один тег от способа обмена и количества тегов в группе.

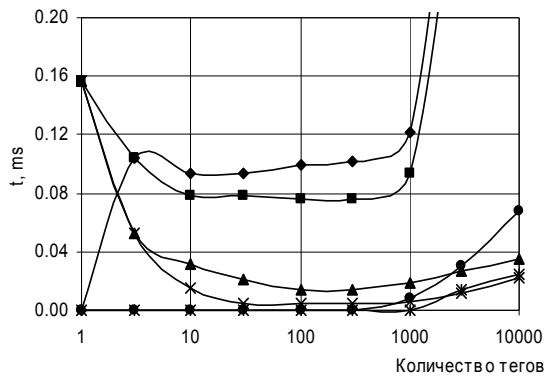


Рис. 5. Зависимость суммарных затрат процессорного времени на один тег от способа обмена и количества тегов в группе:

—◆— ReadItem —■— WriteItem —▲— SyncRead
—×— SyncWrite —*— AsyncRead —●— AsyncWrite

Выводы

Разработанное ПО, обеспечило управление режимами, измерение первичных показателей и сохранение их в файл для последующей обработки. Обработка результатов измерений, выполненная с помощью функций статистического анализа, позволила сделать ряд важных для практики заключений.

1. Существуют зависимости между пропускной способностью систем на основе OPC-спецификации и затратами процессорного времени, с одной стороны, и способом обмена информацией, количеством передаваемых тегов в одном обмене, с другой. Эти зависимости таковы, что производительность систем ухудшается в десятки и более раз, если обмен организован нерационально.

2. Если актуальной является задача повышения пропускной способности и вычислительной мощности таких систем, то целесообразна разработка адаптивных алгоритмов, которые будут подбирать наиболее эффективный способ обмена. В частных слу-

чаях, когда нагрузка системы динамически не меняется, возможен выбор рациональных режимов взаимодействия клиента с сервером на этапе разработки.

3. Создание таких адаптивных алгоритмов потребует оценки соответствующих затрат времени на выполнение операций.

4. Клиент-серверная система на основе OPC обладает достаточными запасами детерминизма и реактивности, чтобы гарантировать обработку информации с интенсивностью обмена не менее 10 000 тегов в секунду, удовлетворяя при этом требованиями мягкого реального времени. Если предъявляются требования жесткого реального времени, то указанную величину следует уменьшить в несколько раз.

Литература

1. Кузнецов А. SCADA-системы: Программистом можешь ты не быть... // Современные технологии автоматизации. – 1996. – № 1. – С. 32-35.
2. OLE for Process Control [Электрон. ресурс]. – Режим доступа: <http://opcfoundation.org>.
3. Финогенов К.Т. Программирование измерительных систем реального времени. – М.: Энергоатомиздат, 1990. – 256 с.
4. Проверка и утверждение программ реального времени / Под. ред. Уи. Дж. Квирка. – К.: Наук. думка, 1990. – 216 с.
5. Рихтер Дж. Windows для профессионалов: создание эффективных Win32-приложений с учетом специфики 64-разрядной версии Windows. – С.-Пб.: Питер, 2001. – 752 с.
6. QNX Neutrino Real-time OS: Kernel Benchmark Methodology [Электрон. ресурс]. – Режим доступа: <http://www.qnx.com>.
7. PC-club [Электрон. ресурс]. – Режим доступа: <http://pcclub.com.ua>.

Поступила в редакцию 4.06.2007

Рецензент: д-р техн. наук, проф. И.В. Чумаченко, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.