

ОРГАНИЗАЦИЯ ПРИНУДИТЕЛЬНОГО ОПРОСА В КОМПЛЕКСНОЙ СИСТЕМЕ УЧЁТА ЭНЕРГОРЕСУРСОВ

Ковцова И.О.

Международный университет природы, общества и человека «Дубна»
Филиал «Протвино»

В данной статье описываются принципы организации принудительного опроса в комплексной системе учёта ресурсов. Рассмотрены основные моменты реализации данного решения на примере сервера учёта.

THE ORGANIZATION OF COMPULSORY POLL IN THE COMPLEX SYSTEM OF THE POWER RESOURCES ACCOUNTING

Ковцова И.

This article describes the principles of compulsory poll in complex system of the power resources accounting. The main points of this organization solutions for the calculation server.

Одним из важнейших сервисов, которым должна обладать система учёта ресурсов, является принудительный опрос или опрос по запросу оператора. Как правило, данная функциональность необходима в силу того, что оператор должен иметь возможность в любое время запросить данные непосредственно с прибора учёта или УСПД. А так же данная возможность оказывается полезной при осуществлении диагностики работы распределённой многоуровневой системы. Принудительный опрос позволяет:

- получать архивные данные за указанный промежуток времени с любого узла системы, не меняя логики опроса, которая, как правило, осуществляется по расписанию;
- получать текущие данные: например, при возникновении аварийных режимов предусмотрено получение 3-х минутных интервалов усреднения мощности, по запросу оператора.

В данной статье рассматривается способ организации принудительного опроса в комплексной системе учёта энергоресурсов. Межуровневый обмен данными в системе осуществляется в XML-формате при помощи специализированного разработанного протокола.

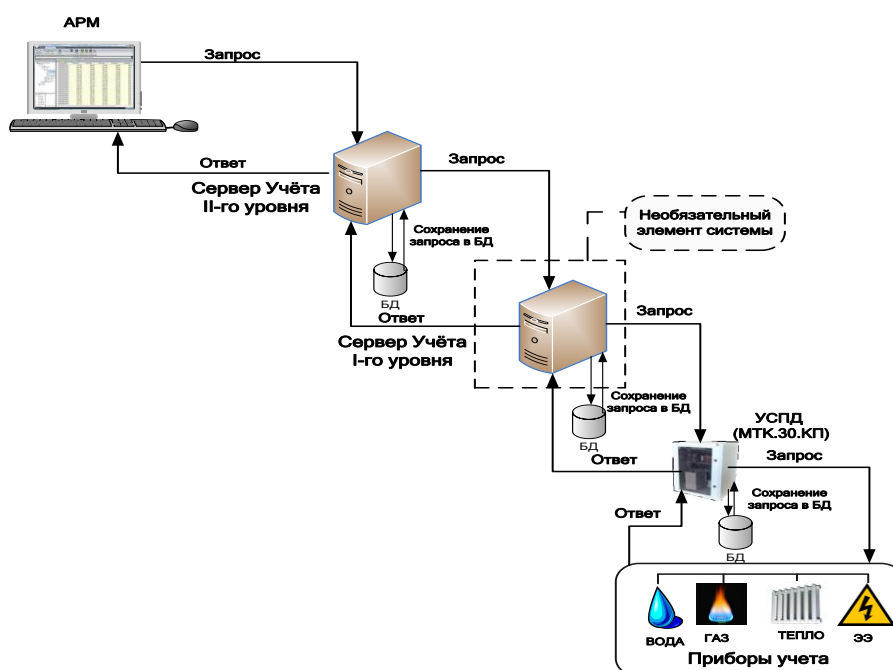


Рис.1. Схема передачи запроса и ответа в системе

При одноуровневой организации системы клиент отправляет запрос серверу, а сервер пересылает его УСПД, которое является конечным устройством, выполняющим принудительный запрос. При многоуровневой организации или, например при двухуровневой (Рис1.), Сервер Учёта II-го уровня должен уметь ретранслировать запрос Серверу Учёта I-го уровня, а тот, в свою очередь, УСПД. Логика работы сервера не зависит от его уровня. При построении таких систем важным фактором является однозначная идентификация сообщений и узлов в системе.

Рассмотрим основные моменты организации принудительного опроса в сервере учёта:

I. **Хранение запросов в сервере и формирование уникального идентификатора:** сервер, получив запрос, создаёт объект `OprRequestElement` и выставляет необходимые атрибуты: `LogTime` (время когда, был получен данный запрос), `XMLId` (идентификатор XML-сообщения) и другие. `XMLId` является очень важным атрибутом, так как позволяет однозначно определять XML-сообщение в системе. Затем сервер формирует запрос УСПД (или серверу) со своим `XMLId`, который определяется способом хранения запроса в сервере.

В сервере запросы на принудительный опрос хранятся следующим образом, представленном на Рис2. Есть массив дней, размер которого определяется переменной, означающей, сколько дней необходимо хранить запросы. В каждом дне хранится список запросов в векторе. Каждый запрос имеет порядковый идентификатор, равный индексу вектора. Таким образом, `XMLId`, который сервер присваивает отправляемому запросу, имеет вид: “индекс дня \ порядковый индекс запроса в данном дне”.

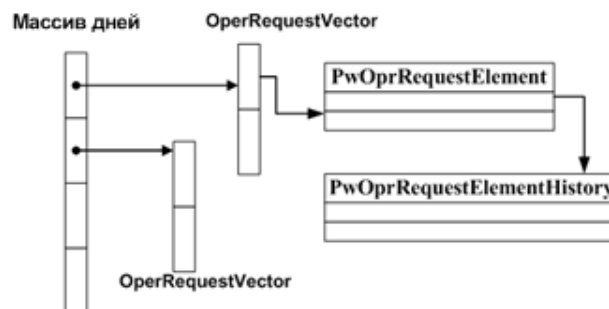


Рис.2. Способ хранения запросов на принудительный опрос в сервере

Пользователи работают, в основном с точками учёта. На каждую из которых монтируется прибор учёта. Это обусловлено тем, что прибор учёта может сломаться, его могут заменить, а для оператора важны данные по потреблению ресурсов по данной точке. Таким образом, когда сервер получает запрос на принудительный опрос, то в нём указывается точка учёта. А УСПД отправляется запрос, где указывается прибор учёта, который смонтирован на заданном промежутке времени, так как УСПД работает только с измерительными устройствами. Если сервер должен передать запрос другому серверу, то тогда указывается точка учёта и дополнительно уникальный идентификатор сервера в системе. Для этого на сервере хранится топология объекта.

II. **Контроль выполнения запроса, формирование статуса:** после того как запрос отправлен, объекту `OprRequestElement` выставляется текущий статус `AcceptCommand`, что запрос принят к исполнению и взводится таймер. Статус запроса может изменяться в процессе выполнения принудительного опроса и принимать одно из следующих значений:

- a) `AcceptCommand` – команда принята к исполнению (п)
- b) `RejectCommand` – команда отклонена (о)
- c) `WaitAnswerCommand` - ожидание ответа на команду (п)
- d) `DoneCommand` - команда выполнена (о)
- e) `ErrorCommand` - ошибка при выполнении команды (о)
- f) `TimeOutCommand` - превышен интервал ожидания на выполнение команды (п)
- g) `BlockCommand` - команда заблокирована (о)
- h) `RejectTryLimitCommand` – превышен лимит повторно выполняемых команд (о)

По истечении времени таймера осуществляется проверка статуса запроса. Статус может быть промежуточным (п) или окончательным (о). Если промежуточный статус, например время

ожидания ответа истекло (TimeOutCommand), тогда посылается запрос повторно и заново взводится таймер. Количество повторно выполненных запросов устанавливается, как правило, равным трём, но по желанию может настраиваться, как и время ожидания запроса. При отсутствии ответа и по истечении всех попыток устанавливается статус, что количество попыток истекло (RejectTryLimitCommand). После того, как статус запроса станет окончательным, таймер будет остановлен.

В целях повышения надёжности и прозрачности работы системы хранится история изменения статуса запроса. Поэтому каждый OprRequestElement имеет список объектов OprRequestElementHistory, где фиксируется статус запроса, время изменения статуса, а в случае ошибки фиксируется её код и текстовое описание.

При восстановлении системы после сбоя запросы, которые имеют промежуточный статус (прерванный запрос), будут выполнены системой. Все запросы должны иметь окончательный статус.

III. **Обработка ответа:** сервер, получив ответ, обновляет статус запроса. Статус запроса берётся из ответа, что позволяет при неуспешном выполнении команды получать детальный анализ возникшей проблемной ситуации. При успешном выполнении команды сервер получает архивные данные, которые он должен обработать.

Когда данные в сервере за запрашиваемый период отсутствуют, то сервер записывает их в базу данных. Однако, когда эти данные у сервера уже есть, то необходимо решить, как с ними поступить. Решение принимается в зависимости от статуса данных.

Данные могут иметь один из следующих статусных признаков: достоверные, недостоверные, замещенные данные и данные, введённые оператором, - ручной ввод.

Логика работы с пришедшими архивными данными:

1. Пришедшие данные достоверны и совпадают с данными сервера – данные отбрасываются.
2. Пришедшие данные достоверны, а у сервера не достоверные – данные перезаписываются.
3. Данные недостоверны, а у сервера достоверны - данные отбрасываются.
4. Данные достоверные, а у сервера ручной ввод:
 - а) С разрешением перезаписи – данные перезаписываются.
 - б) Без разрешения перезаписи – данные отбрасываются.
5. Недостоверные данные и у сервера недостоверные, но данные различны – данные отбрасываются, пишется предупреждение в LogFile.
6. Данные достоверны и у сервера данные достоверны, но значения различны – данные отбрасываются, пишется предупреждение в LogFile.

IV. **Формирование ответа клиенту:** в ходе выполнения сервером принудительного опроса сервер должен информировать клиента о ходе выполнения запроса и его завершении. Клиенту отправляется XML-ответ, когда запрос принят к исполнению или в случае ошибки. При успешном выполнении принудительного опроса сервер информирует клиента об успешном выполнении и о том, что идёт поступление данных.

Аналогичная логика работы и у УСПД. Получив запрос, он его обрабатывает, формирует ответ серверу о статусе запроса и начинает его выполнение. Если запрос на принудительный опрос УСПД, то УСПД делает выборку данных из базы данных. Если это запрос на принудительный опрос прибора учёта, то тогда УСПД начинает опрашивать измерительное устройство. Получив данные и сформировав ответ, УСПД отсылает его серверу, либо сообщает об ошибке.

Таким образом, описанный способ организации принудительного опроса или опроса по запросу оператора позволяет запрашивать текущие и архивные данные в любое время, что делает систему более удобной и функциональной. Предложенный механизм обеспечивает гарантированное выполнение и получение результата за конечное число шагов и время, а так же обеспечивает корректное завершение выполнения запросов после сбоя или остановки системы. Данное решение может быть применимо для выполнения любых операций в асинхронном режиме работы.