

Система аналитических отчетов (ARS)

03.2022, Скребцов В.И., v.skrebtsov@mail.ru

1. Введение

Система аналитических отчетов (Analytic Report System - ARS) предназначена для получения в виде секционированных (вложенных) таблиц аналитических, статистических, сводных, списочных и прочих видов вторичных данных на основе различных целевых источников первичных данных. Секционированные таблицы представляют собой наиболее общую форму представления аналитических отчетов. В качестве целевых источников первичных данных могут выступать как данные из СУБД (например MySQL для Moodle), так и текстовые файлы логов (например для Jitsi Meet).

2. Основные требования

2.1. ARS является полноценным WWW приложением, функционирующем за сервером Nginx и обеспечивающим многопользовательский доступ к отчетам (параллельное получение множества отчетов).

2.2. ARS не является составной частью какой-либо целевой системы (например Moodle).\ поскольку сама природа аналитических отчетов предполагает наличие ряда свойств (ограничений):

1. Сборка отчета может потребовать значительного времени и вычислительных ресурсов. Особенно к этому чувствительны СУБД, поскольку, как правило, сложные параллельные SQL запросы могут привести к серьезной деградации их производительности и вызвать отказы целевых (особенно интерактивных) систем. Например Moodle является типичной интерактивной системой, используя короткие SQL запросы к СУБД при интерактивной работе с пользователем (речь не идет о таких сервисных функциях как загрузка/выгрузка данных).
2. При этом система индексов СУБД заточена именно под интерактивную работу с пользователем. Как правило для получения приемлемого времени сборки отчета требуется создания специализированных индексов (индексов, которые используются только этим отчетом). Но поддержание индекса в актуальном состоянии также требует вычислительных ресурсов. То же самой можно сказать про специализированные таблицы и/или viewers.

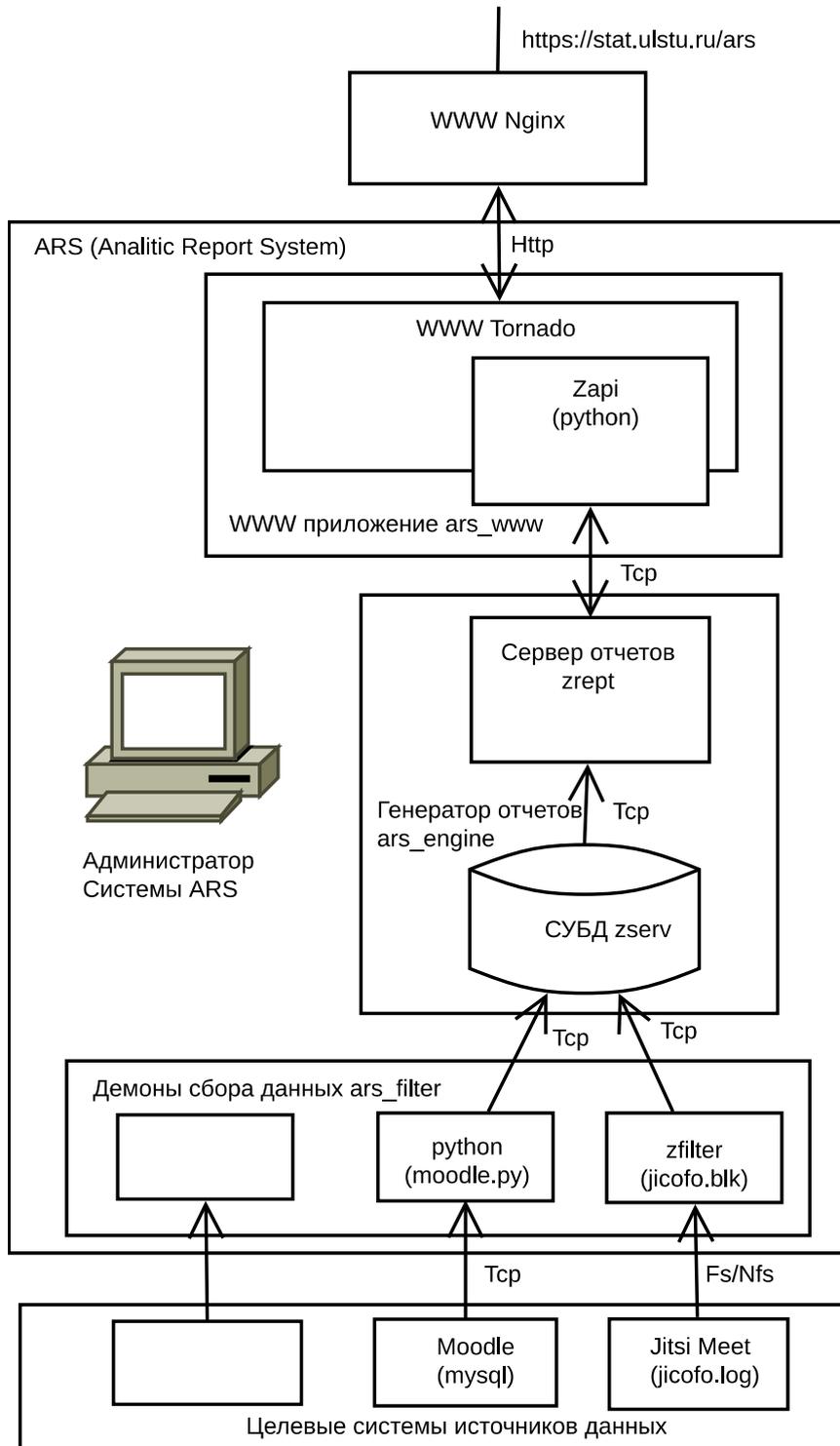
2.3. Учитывая вышеизложенное, ARS должен иметь возможность функционировать на внешнем сервере, а не только на сервере целевых систем (которые, кстати, также могут быть различными).

2.4. Пользовательский интерфейс ARS должен предусматривать следующие элементы:

1. Список отчетов (возможно иерархический).
2. Возможность параметризации сборки отчета. Например форма для ввода интервала дат и времени для конкретного отчета.

3. Возможность мониторинга хода сборки отчета (строка прогресса и процент выполнения)
4. Возможность прерывания сборки отчета и получения частично-собранного отчета.

2.5. Получение аналитического отчета, это не только SQL запрос к СУБД. Как правило требуется пост-обработка полученных данных и возможно дополнительные SQL запросы. Таким образом, встает вопрос о специализированном декларативном языке отчетов, который используется для быстрого создания отчетов на базе шаблонов.



Структурная схема системы аналитических отчетов ARS (Analytic Report System)

3. Технические решения ARS

С архитектурной точки зрения ARS состоит из трех компонентов: WWW сервер `ars_www` (не путать с Nginx, который является для него frontend), демон данных `ars_data` и движок отчетов `ars_engine`.

3.1. **Ars_www** сервер реализован на базе открытого ПО Tornado (<https://www.tornadoweb.org>). Tornado предоставляет асинхронный неблокируемый сетевой ввод/вывод и является идеальным решением для приложений, которые используют долгоиграющие соединения с каждым пользователем. Как раз то, что и требуется для системы отчетов. Кроме того, Tornado реализует WWW шаблоны с поддержкой CSS, позволяя создавать пользовательский интерфейс на их основе. И наконец, Tornado изначально интегрирован с Nginx.

3.2. **Ars_data** демон представляет собой процесс, который периодически (например каждую минуту) анализирует данные целевых систем (например, некоторые таблицы MySQL Moodle и/или логфайлы Jitsi Meet) на предмет появления новых данных и на каждой итерации загружает порцию новых данных целевых систем в специализированное СУБД Zcode. СУБД Zcode представляет собой полноценное реляционное СУБД с индексами и триггерами, включая поддержку резидентных данных, находящихся исключительно в оперативной памяти. Важным свойством Zcode является его высокое быстродействие.

`Ars_data` по сути является посредником между целевыми системами и системой сборки отчетов, защищая целевые системы от несвойственных им операций. `Ars_data` выгружает из целевых систем только новые данные, не нагружая их каким-либо анализом. Все данные для отчетов, включая необходимые индексы, находятся в СУБД Zcode.

3.3. **Ars_engine** движок отчетов является центральной частью ARS, выполняющей собственно сборку отчетов. Логика сборки каждого отчета описывается шаблоном на декларативном языке Zcode. Этот язык реализует ряд ключевых понятий табличного представления данных и ориентирован на сборку отчетов.

Со стороны `ars_www` поступает параметризованный запрос на сборку конкретного отчета. `Ars_engine` выполняет соответствующий шаблон отчета, получая необходимые данные из СУБД Zcode. В результате сборки отчета `ars_engine` формирует `www` шаблон Tornado, который через `ars_www` передается конечному пользователю.

4. Заключение

ARS позволяет организовать параллельное получение множества отчетов через WWW, не нагружая при этом целевые системы (например Moodle). Все компоненты ARS созданы на свободном ПО и следовательно имеют доступные исходные коды.