

Санкт-Петербургский Государственный Университет

Математико-Механический факультет

Кафедра системного программирования

Система мониторинга MSSQL сервера: отслеживание и  
анализ характеристик производительности

Курсовая работа студента 344 группы

Столпнера Льва Артемовича

Научный руководитель: ст. преп. Давыденко А. А.

Санкт-Петербург

2015 г.

## Оглавление:

1 Введение.....	3
2 Постановка задачи.....	4
3 Существующие решения.....	5
3.1 RedGate SQL Monitor.....	5
3.2 SolarWinds Server&Application Monitor.....	6
3.3 Spotlight Dell Software.....	6
4 Используемые технологии и инструменты.....	7
5 Счетчики производительности.....	8
6 Мониторинг и хранение данных.....	9
7 Статистический анализ.....	11
7.1 Сглаживание.....	11
7.2 Стационарность.....	12
7.3 Определение трендов.....	13
7.4 Определение периодичности.....	14
7.5 Многомерный анализ.....	16
8 Результаты.....	17
9 Литература.....	18

## Введение в предметную область

В современном мире ни одна компания не обходится без специального серверного оборудования. И с этой точки зрения программное обеспечение, предоставляющее возможности хранения данных, является одной из самых важных составляющих успешной и налаженной работы системы. Однако не всегда функционирование сервера протекает гладко – иногда могут возникать перегрузки, потери информации, задержки в работе системы, как следствие, отказ приложения. Этому может предшествовать целый ряд причин, в которых невооруженным глазом человеку будет разобраться достаточно сложно.

В связи с этим очень востребована область мониторинга серверов – постоянно необходимо отслеживать правильность работы приложений, вовремя узнавать о возможных перегрузках, неполадках в работе. Основным принципом организации мониторинга является сбор различных показателей сервера, и его последующий анализ. Это позволяет облегчить выявление причин сбоев, выяснение причинно-следственных связей между отслеживаемыми событиями в сервере, нахождение узких мест, исправление или оптимизацию работы с ним.

Одним из способов анализа является мониторинг встроенных данных о производительности сервера. Например, в рассматриваемом Microsoft SQL сервере существует специальная группа счетчиков, контролирующая разнообразные параметры производительности системы, и предоставляющих состояние системы в каждый момент времени. Благодаря анализу этих данных, выявлению зависимостей между различными характеристиками, можно будет построить полную картину работы сервера за заданный промежуток времени, а также понять причины возникших сбоев, некорректном или неоптимальном использовании его возможностей, проблемных местах.

Стоит отметить, что, благодаря анализу статистики производительности сервера на определенном промежутке времени, можно делать попытки предсказать ближайшее

будущее в работе сервера. Составление прогноза может стать сильным подспорьем в деле своевременной оптимизации работы сервера.

Часто в системах мониторинга основная направленность идет на анализ показателей в реальном времени и создании системы оповещения для быстрого реагирования на эти сигналы. В данной же работе упор будет сделан на анализ статистики фиксированного временного интервала, что позволит получать более полную картину о происходивших событиях и связей между ними, выявлении тенденций в изменении производительности, определении периодичности в скачках показателей или какой-то взаимосвязи между параметрами сервера.

## Постановка задачи

- Мониторинг, сбор и хранение характеристик производительности сервера
- Статистический анализ данных:
  - Сглаживание
  - Стационарность
  - Определение трендов
  - Определение периодичности
  - Многомерный анализ
- Выявление зависимостей между параметрами производительности
- Апробация результатов анализа на модельной схеме

## Существующие решения

Так как мониторинг данных это весьма обширная и развитая область, стоит проанализировать некоторые существующие решения или аналоги, предоставляющие похожий функционал по анализу показателей сервера. Рассмотрим некоторые продукты, представляющие возможности для мониторинга MSSQL-сервера, например:

- SQL Monitor от RedGate
- Server&Application Monitor от SolarWinds
- Spotlight от Dell Software.

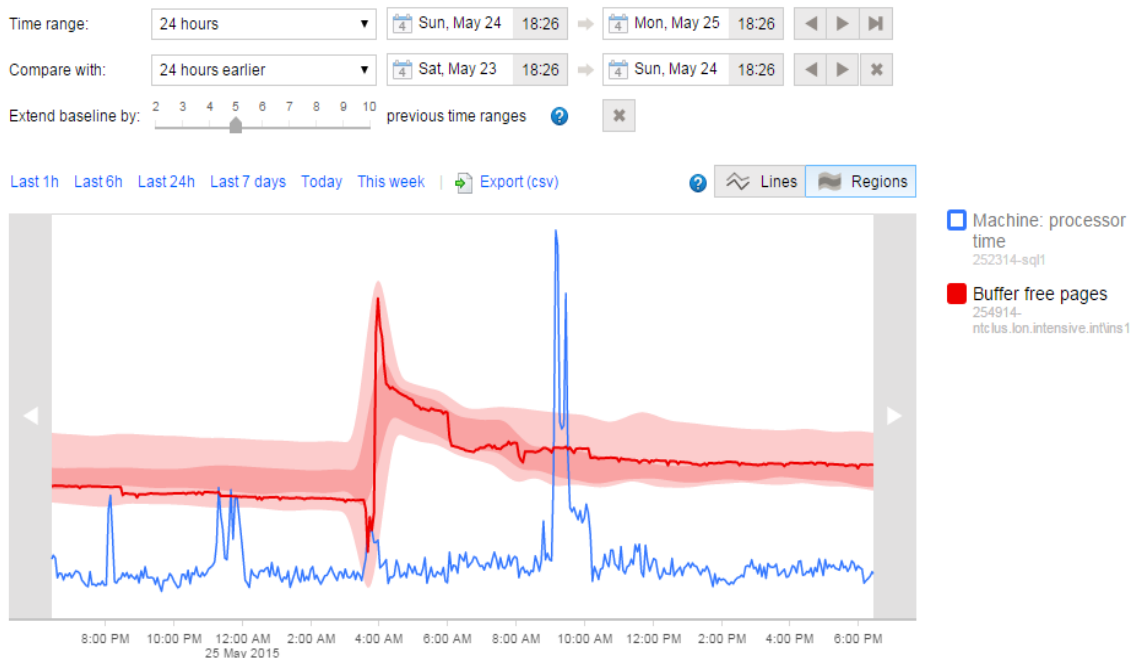


Рис. 1 SQL Monitor, скриншот сделан с сайта компании RedGate<sup>[1]</sup>

SQL Monitor предоставляет возможности для отслеживания различных характеристик машины, небольшой набор счетчиков из MSSQL для построения графиков и упрощения анализа показаний, систему оповещений, оценки продолжительности

выполнения SQL-запросов, получение считанных параметров в виде csv-файла.

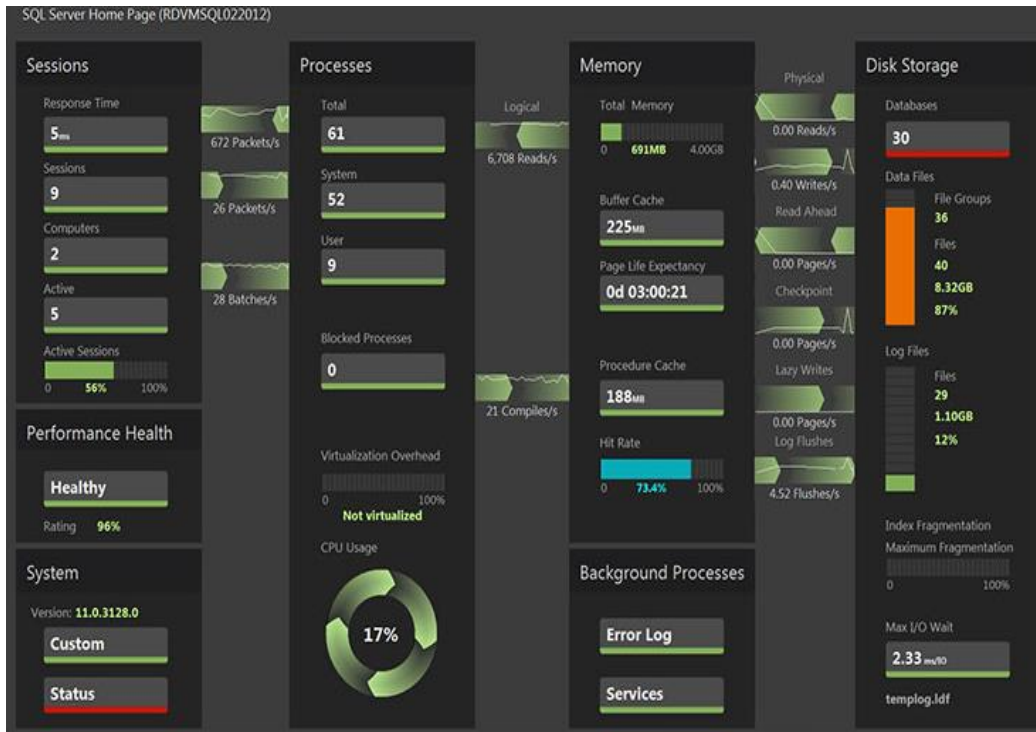


Рис. 2 Spotlight, скриншот сайта компании Dell Software<sup>[2]</sup>

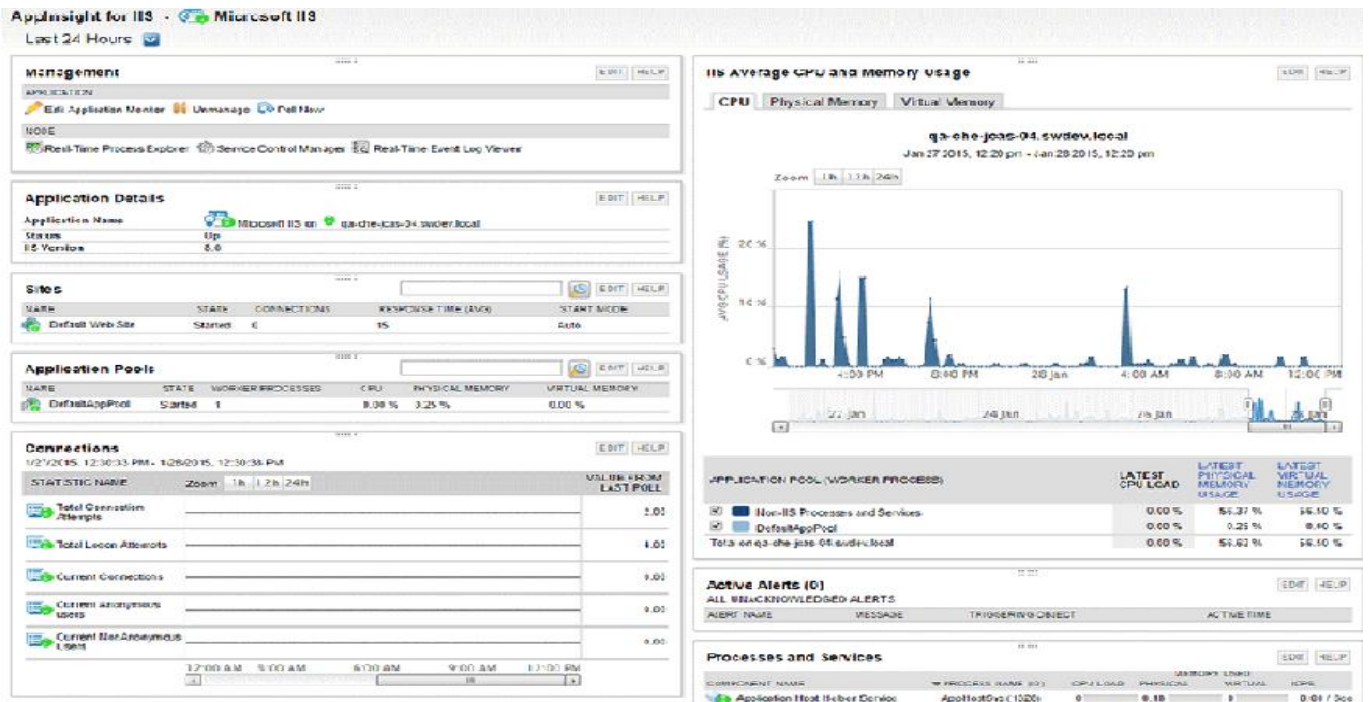
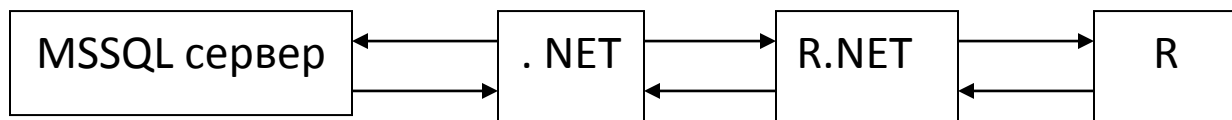


Рис.3 Server&Application Monitor, скриншот с сайта компании SolarWinds<sup>[3]</sup>

В основном, работа этих продуктов заключается в создании мониторинга для конкретного сервера, возможности выбора различных машинных и серверных метрик для анализа, отображения этих данных в виде графиков, обеспечении системы оповещений и общем анализе качества работы системы, в случае SolarWinds сделан больший уклон на работу приложений. Однако они не предоставляют возможностей для автоматического анализа статистики, выявления каких-либо трендов в показаниях, определении периодичности данных для получения важной информации о работе сервера и упрощения принятия решений тех, кто эту информацию будет анализировать. Все это остается на совести и опыте людей, вручную отслеживающих производительность сервера, также отсутствуют механизмы для выявления зависимостей между показаниями или прогнозирования поведения сервера на ближайший отрезок времени. Поэтому было решено реализовать собственную систему мониторинга, учитывающую указанные недостатки в аналогичных разработках.

## Используемые технологии и инструменты:

- .NET – платформа, на которой ведется основная часть разработки системы
- Microsoft SQL Server 2012 – используемый для мониторинга сервер
- R – пакет для проведения статистического анализа
- R.NET-библиотека, служащая связующим звеном между проектом в .NET и R, и позволяющая из проекта получать удобный доступ к возможностям и функциям R



Общая схема проекта

## Счетчики производительности

Основным объектом изучения производительности являются встроенные в MSSQL сервер счетчики производительности. Более тысячи параметров являются открытыми для наблюдения, и, соответственно, для дальнейшего качества мониторинга необходимо создать правильную и показательную выборку из этого множества. После прочтения нескольких статей<sup>[4]</sup> с набором наиболее важных, на взгляд авторов, счетчиков была сделана предварительная выборка, составлявшая около 20 счетчиков для постоянного мониторинга. Затем было решено, что необходимо разделить показатели на две подгруппы с более определенными целями: первая для отслеживания более глубоких внутренних характеристик производительности,

Buffer cache hit ratio	Процент страниц, найденных в буфере
Page life expectancy	Время, которое страница остается в буфере
Locks: Average Wait Time (ms)	Среднее время ожидания для всех блокировок
SQL Compilations/sec	Количество компиляций в секунду
Full Scans/sec	Количество полных сканирований в секунду

Табл. 1 Пример счетчиков производительности первой группы

вторая для набора «пользовательских» параметров и счетчиков, прямым образом зависящих от приложения, работающего с сервером, и от работы пользователей.

User connections	Количество подключенных пользователей
Active Transactions	Количество активных транзакции в базе данных
Batch Requests/sec	Количество запросов к серверу в секунду
Longest Transaction Running Time	Максимальное время работы транзакции в секундах
Processes blocked	Количество заблокированных процессов в этот момент

Табл. 2 Пример счетчиков производительности второй группы

Система также предоставляет возможность для самостоятельного создания и настройки списка счетчиков в зависимости от целей мониторинга и использования его вместо предусмотренных изначально групп счетчиков.



## Мониторинг и хранение данных

Чтобы получить необходимые счетчики, необходимо было реализовать взаимодействие через .NET с SQL-сервером, для чего была использована библиотека SqlClient. Система в отдельном потоке создает собственное подключение к указанному серверу с определенным способом аутентификации. Затем генерирует для каждого счетчика из списка соответствующие SQL-запросы, содержащие сведения об объекте сервера, содержащем нужный элемент, и самом названии счетчика. Отправляя этот запрос в систему, производится считывание всех полей, хранящихся в счетчике, преобразование их к удобному виду для дальнейшего использования и сохранение их в специальном списке счетчиков, хранящем «вектора» показаний. И эта операция выполняется с указанным периодом до момента окончания работы программы или поступления команды на окончание мониторинга данных, от него будет зависеть точность и полнота данных.

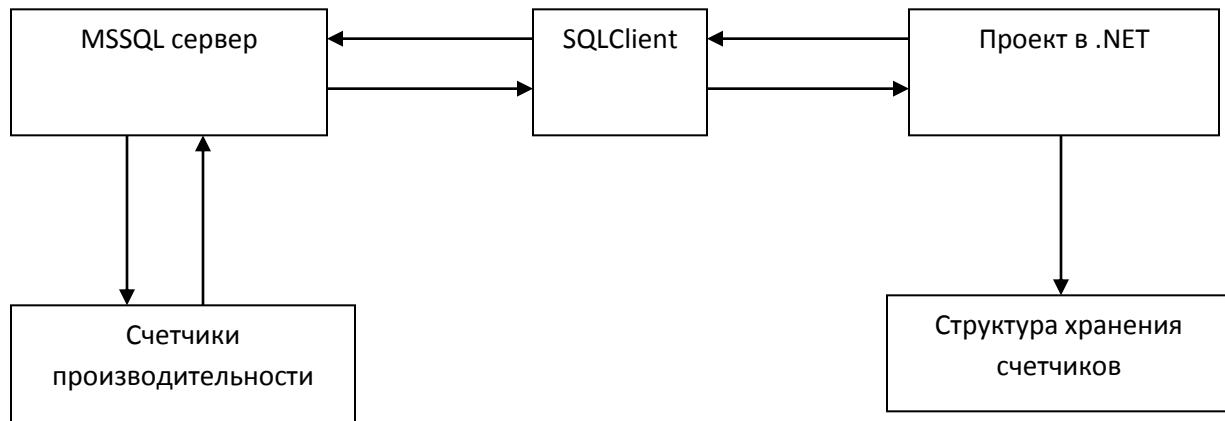


Рис. 4 Схема получения данных из MSSQL-сервера

Счетчики производительности имеют собственную структуру, состоящую из 5 различных полей, где и хранится необходимая для работы с ними информация:

- object\_name – поле, содержащее сведения о принадлежности счетчика конкретному объекту MSSQL, содержащему набор специализированных счетчиков
- counter\_name – название счетчика
- instance\_name – обычно, имя базы данных
- cntr\_value – само значение счетчика
- cntr\_type – тип счетчика

#### Общий вид счетчика производительности

object_name	counter_name	instance_name	cntr_value	cntr_type
-------------	--------------	---------------	------------	-----------

#### Пример одного из счетчиков

MSSQL\$MSSQL:Databases	Transactions/sec	Test	149415	272696576
------------------------	------------------	------	--------	-----------

# Статистический анализ

## Сглаживание

Одним из самых простых элементов статистического анализа можно назвать сглаживание. Изначально был исследован этот способ обработки данных с целью реализации выявления тренда в показаниях счетчиков производительности. Для этого было выбрано несколько алгоритмов, а именно – метод простого скользящего среднего и простого экспоненциального сглаживания:

- Алгоритм простого скользящего среднего при достаточно большом параметре  $N$  слишком сильно искажает изначальные данные, а при малых  $n$  сохраняются случайные колебания величин:

$$M_t = \frac{\sum_{i=0}^{N-1} X_{t-i}}{N},$$

где  $N$  – количество усредняемых данных,  $X_t$  – значения в определенные моменты времени,  $M_t$  – новые значения

- Алгоритм простого экспоненциального сглаживания чуть более точен, но также не дает точного представления о тренде в показаниях, и тем более периодичности:

$$S_t = \alpha X_{t-1} + (1 - \alpha)S_{t-1}, 0 < \alpha \leq 1, t \geq 3,$$

где  $S_t$  – новые сглаженные значения,  $X_t$  – изначальные значения,  $\alpha$  – параметр, подбираемый таким образом, чтобы среднее квадратов ошибок было минимальным:

$$MSE = \frac{\sum_{i=1}^n (X_i - S_i)^2}{n} \rightarrow \min$$

Алгоритм является экспоненциальным, так как веса перед элементами убывают с экспоненциальной скоростью:

$$S_t = \alpha \sum_{i=1}^{t-2} (1 - \alpha)^{i-1} X_{t-i} + (1 - \alpha)^{t-2} S_2, t \geq 2$$

Но, так как эти методы не показали высокой точности для получения необходимых результатов, то были отброшены как неподходящие для решения поставленной задачи.

## Стационарность данных

Основным показателем отсутствия тренда и периодичности в данных могло бы служить подтверждение факта о стационарности временного ряда. Стационарность в одном из своих определений означает отсутствие изменений в характеристиках ряда с течением времени, в данном случае рассматривалось сохранение среднего значения и дисперсии (разброса) значений по всему временному промежутку, из чего можно сделать вывод о том, что данные не содержат определенного тренда или периодичности в своих значениях. Для анализа стационарности данных можно было бы использовать какие-нибудь из наиболее известных алгоритмов, например, расширенный тест Дики-Фуллера, KPSS тест (Kwiatkowski–Phillips–Schmidt–Shin), но было решено определять наличие и составляющие тренда и периодичности другими методами, указанными ниже.

## Определение трендов

Так как алгоритмы простого сглаживания данных не зарекомендовали себя в качестве хороших методов для анализа трендов, следует для вычисления тенденций в данных использовать более сложный алгоритм двойного экспоненциального сглаживания. Он заключается в усложнении простого сглаживания с добавлением новой константы, добавляющей трендовую составляющую и делящую временной ряд на два других:

$$S_t = \alpha X_t + (1 - \alpha)(S_{t-1} + b_{t-1}), 0 \leq \alpha \leq 1$$

$$b_t = \gamma(S_t - S_{t-1}) + (1 - \gamma)b_{t-1}, 0 \leq \gamma \leq 1$$

Первое приведенное выше уравнение, в отличие от простого сглаживания, устанавливает  $S_t$  непосредственно для тренда предыдущего периода с помощью добавления к прошлому сглаженному значению  $S_{t-1}$  значения  $b_{t-1}$ . Это помогает сильно уменьшить отставание от реальных значений. Второе же уравнение напоминает алгоритм простого сглаживания, но его целью является обеспечение постоянного обновления тренда.

Выбор коэффициентов  $\alpha$  и  $\gamma$  для уравнений осуществляется с помощью нелинейных оптимизирующих алгоритмов, например, с помощью алгоритма Марквардта для решения задачи о наименьших квадратах, однако, в случае данной работы эти возможности предоставляет система R.

## Определение периодичности

Однако, алгоритм двойного экспоненциального сглаживания все еще недостаточно полезен при выделении периодичности в значениях ряда, несмотря на всю пользу при определении тренда, вследствие чего возникает желание использовать еще более усовершенствованный алгоритм, и получить также составляющую периодичности значений.

Для определения периодичности сначала предполагалось использовать методы, определяющие автокорреляции и частичные автокорреляции во временном ряде. Основной принцип их работы в том, чтобы найти зависимость значений одного ряда между данными с каким-то фиксированным временным промежутком между ними. Также в среде программирования R благодаря функции `decompose()` предоставляется удобная возможность быстрого разделения временного ряда на модель с 3 составляющими: тренда, периода и остаточных значений.

Однако этот алгоритм использует достаточно примитивные алгоритмы скользящих средних, из-за чего более показательным и удобным является метод Холта-Винтерса, по сути, являющийся алгоритмом тройного экспоненциального сглаживания. Алгоритм представляет собой модификацию двух предыдущих добавлением еще одной формулы для периодичности.

Опишем совокупность уравнений для метода Холта-Винтерса:

$$S_t = \alpha \frac{X_t}{I_{t-L}} + (1 - \alpha)(S_{t-1} + b_{t-1})$$

$$b_t = \gamma(S_t - S_{t-1}) + (1 - \gamma)b_{t-1}$$

$$I_t = \beta \frac{X_t}{S_t} + (1 - \beta)I_{t-L}, \text{ где}$$

X – показания, S-сглаженные показания, b коэффициент для тренда, I – индекс периодичности, t – индекс для обозначения временных периодов, L-количество периодов

Коэффициенты  $\alpha$ ,  $\beta$ ,  $\gamma$  вычисляются таким же способом, как и при двойном экспоненциальном сглаживании, обеспечивая наименьшее среднее квадратов ошибок.

Для использования этого метода требуется хотя бы один полный период данных для вычисления значений  $I_{t-L}$ .

Так как этот алгоритм позволяет одновременно определять и тренд, и периодичность в значениях временного ряда, он и был выбран как главный при решении выбранной задачи.

## Многомерный анализ

Одним из самых важных и практически полезных в анализе статистических данных производительности сервера является так называемый многомерный анализ, так как он позволяет сопоставить различные характеристики друг с другом, степень их коррелированности, выявить причинно-следственные связи в изменениях трендов, зависимости между различными счетчиками. Это бы позволило точнее оценивать и, как следствие, оптимизировать работу сервера.

Для анализа зависимости различных данных будем определять корреляция временных рядов для различных счетчиков. В зависимости от коэффициента корреляции, можно будет определить степень их влияния друг на друга, и, проанализировав этот факт, могут предприниматься различные действия по изменению или улучшению работы системы.

Вероятно, впоследствии блок многомерного анализа можно будет развить, используя более углубленные методы анализа зависимости параметров друг от друга. Например, анализ главных компонент (для создания наиболее показательных компонент из наименьшего числа элементов) или линейный дискриминантный анализ (для описания различных классов счетчиков).



## Результаты

В рамках проведенной работы были исследованы различные системы мониторинга MSSQL сервера, разработана собственная система, имеющая по умолчанию две группы счетчиков производительности для отслеживания, позволяющая также пользователю самому создать свой список счетчиков для обзора. С помощью обращений к базе данных запросами в форме SQL было создано считывание и хранение данных в системе. Затем была проведена оценка алгоритмов статистического анализа, выбраны наиболее удовлетворяющие задачам работы, их применение, последним этапом работы было тестирование системы на различных показателях сервера и их сравнение.

## Дальнейшие перспективы

В данной работе были созданы предпосылки для усовершенствования системы, а именно, добавления прогнозирования с учетом проанализированных данных, системы оповещений/предупреждений о возможных проблемах производительности системы, создания удобного интерфейса для использования возможностей этой работы.

## Литература

1. SQL Monitor: [сайт]  
URL: <http://www.red-gate.com/products/dba/sql-monitor/>
2. Spotlight on SQL Server Enterprise: [сайт]  
URL: <http://software.dell.com/products/spotlight-on-sql-server-enterprise/>
3. Server&Application Monitor: [сайт]  
URL: <http://www.solarwinds.com/server-application-monitor.aspx>
4. Understanding SQL Performance Counters: [сайт].  
URL: <http://extremeexperts.com/sql/articles/sqlcounters.aspx>
5. Шипунов А.Б., Балдин Е.М. Наглядная статистика. Используем R! 2014
6. Zucchini W., Nenadic O. Time Series Analysis with R.
7. Croarkin C., Tobias P. – Engineering statistics handbook 2012  
URL: <http://www.itl.nist.gov/div898/handbook>
8. Venables W., Smith D. Introduction to R  
URL: <http://cran.r-project.org/doc/manuals/r-release/R-intro.html>
9. Coghlan A. Little Book of R for Time Series 2010  
URL: <http://a-little-book-of-r-for-time-series.readthedocs.org>
10. Coghlan A. Little Book of R for Multivariate Analysis 2010  
URL: <http://little-book-of-r-for-multivariate-analysis.readthedocs.org/>