МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ ДАЛЬНЕВОСТОЧНЫЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ

Кафедра "ИТиС"

ОТЧЕТ ПО ПРЕДДИПЛОМНОЙ ПРАКТИКЕ

Выполнила:

Симанкова Лидия

Проверил:

Чехонин К. А.

2014 год

г. Хабаровск

В период с 10.02.14 по 9.03.14 мною была пройдена преддипломная практика в Хабаровском информационно-вычислительном центре Дальневосточной железной дороги (ХИВЦ ДВЖД). Во время прохождения практики была согласована тема дипломного проектирования, а также собран необходимый для дальнейшей работы материал.

**Тема:** Разработка информационной системы мониторинга информационных ресурсов ХИВЦ (Хабаровского информационно-вычислительного центра).

Содержание

Введение 4

1 Обзор существующих комплексов мониторинга 6

1.1 IBM Tivoli Monitoring 6

1.1.1 Архитектура 7

1.1.2 Основные возможности ITM 9

1.1.3 Заключение. 12

2 Исследование существующей системы мониторинга 13

2.1 Перечень систем, подлежащих мониторингу 13

2.2 Обзор существующей системы мониторинга 14

2.3 Анализ ошибок в логах 21

2.3.1 Что такое лог 21

2.3.2 Места возникновения ошибок 21

3 Модернизация существующей системы мониторинга 26

Заключение 26

Список использованной литературы 26

Приложения 26

# Введение

Сегодня любая большая организация уделяет много внимания оснащению своих подразделений программными разработками, полностью отвечающими решаемым в организации задачам. Зачастую в пределах одной организации используется программное обеспечение разных разработчиков, которое необходимо интегрировать между собой для эффективной работы всего предприятия. Иногда во время интеграции и синхронизации систем возникают ошибки, для обнаружения которых не существует встроенных средств обнаружения. В связи с этим возникает необходимость в разработке информационной системы мониторинга информационных ресурсов.

Администраторы, руководители и начальники смены информационно-вычислительных центров должны получать достоверную и своевременную информацию о состоянии информационных ресурсов и баз данных, развернутых в инфраструктуре предприятия, в удобной форме по первому запросу. Правильная организация мониторинга информационных ресурсов позволяет обеспечить сотрудников оперативной информацией о возникновении аварийной ситуации и локализовать места сбоев, а также сократить время на их диагностику.

Актуальность дипломной работы обуславливается тем, что для эффективной работы каждому предприятию необходима система мониторинга информационных ресурсов, если таковой не имеется.

Объектом исследования выступает ХИВЦ ДВЖД.

Предметом исследования является как изучение уже существующих систем мониторинга, так и разработка методов совершенствования имеющейся в ХИВЦ системы мониторинга информационных ресурсов.

Целью дипломного проектирования ставится проектирование системы мониторинга информационных ресурсов, которая:

* контролирует доступность и производительность информационного ресурса;
* оповещает сотрудников ИВЦ о критических событиях;
* выявляет проблемы в функционировании информационного ресурса и определяет характер и место сбоя;
* снижает время на восстановление системы при обнаружении отказа.

Задачи дипломного проектирования:

* рассмотреть системы мониторинга, используемые на предприятиях, и изучить их функциональные возможности;
* проанализировать информационные ресурсы дорожного уровня (сервер БД расположен в пределах ДВЖД);
* исследовать существующую в ХИВЦ систему мониторинга и выявить имеющиеся проблемы;
* провести анализ ошибок, возникающих во время работы модулей интеграции с другими дорожными системами, чтобы затем составить алгоритм их обработки;
* разработать веб-приложение, визуально предоставляющее информацию об информационных ресурсах в удобной пользователю форме;
* связать программные разработки в единый программный комплекс.

Результатом выполнения дипломной работы станет информационная система, полностью отвечающая поставленным задачам и целям.

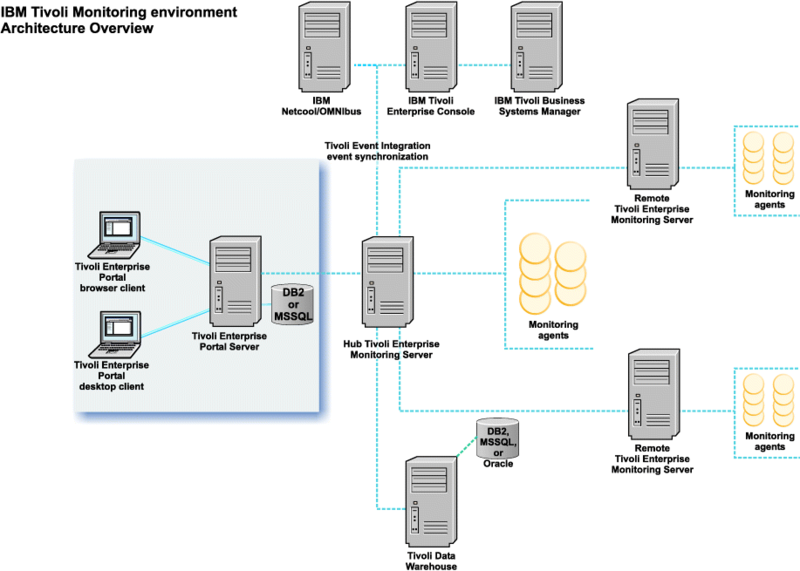
1. Обзор существующих комплексов мониторинга
   1. IBM Tivoli Monitoring

IBM Tivoli Monitoring – система для оптимизации производительности ИТ инфраструктуры и повышения её доступности. Это программное обеспечение позволяет обнаружить нехватку ресурсов в операционных системах физических и виртуальных серверов на ранней стадии, а также выявить проблемы. Tivoli Monitoring позволяют улучшить эффективность ИТ-подразделения, применяя новейшие методы выявления и устранения проблем инфраструктуры.

Задачи, решаемые IBM Tivoli Monitoring:

* помогает заранее выявить простаивающие ресурсы, а также определить узкие места, которые угрожают работе критически важных приложений компании;
* постоянно оценивает ресурсы системы, выявляет потенциальные проблемы на ранних этапах и автоматически на них реагирует. При помощи заблаговременного выявления неполадок средствами Tivoli Monitoring, компании имеют возможность быстро исправить ситуацию, не дав пользователям почувствовать ухудшение качества обслуживания и снижения производительности;
* обладают функционалом динамических порогов срабатывания, который основан на статистическом анализе исторических данных производительности и предназначен для упреждения аварий;
* повышает доступность и среднее время наработки на отказ с быстрой визуализацией аварии и исторических данных, предшествовавших ей, для скорейшего определения причины. Позволяет принять восстановительные меры и сократить перебои в оказании услуг за несколько минут, а не часов;
* контролирует, оповещает и создает отчеты о потенциальных узких местах, связанных с дефицитом ресурсов;
* облегчает мониторинг систем благодаря использованию гибкого, интуитивно понятного интерфейса и настраиваемых графических рабочих консолей. Помимо прочего, имеет в составе хранилище данных и расширенные возможности создания отчетов.
  + 1. Архитектура

Существует основной сервер мониторинга TEMS (Tivoli Enterprise Monitoring Server), портальный сервер TEPS (Tivoli Enterprise Portal Server), агенты мониторинга и база данных (MSSQL, DB2, Oracle или встроенная Derby) (см. Рис. 1).

  
Рис. 1 Архитектура IBM Tivoli Monitoring

Все системы связаны между собой и слаженно работают. Агенты собирают информацию и передают ее на сервер мониторинга, затем на сервере мониторинга эта информация анализируется, и при превышении пороговых значений срабатывают ситуации.

Доступ пользователя ко всему этому функционалу осуществляется через клиента к портальному серверу, который именуется TEP (Tivoli Enterprise Portal) (см. Рис. 2) и выпускается в двух версиях: в виде приложения на рабочем столе и веб-приложения.

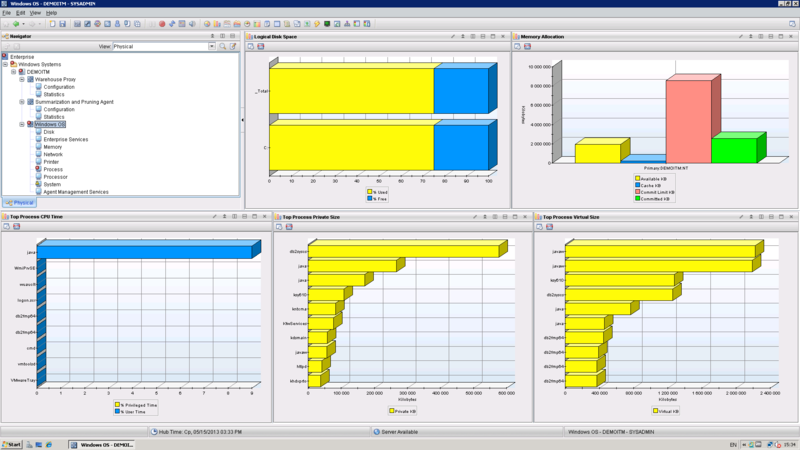


Рис. 2 Интерфейс Tivoli Enterprise Portal

В целом об архитектуре можно сказать следующее:

* У продукта хорошая масштабируемость, он надежен. Им удобно управлять агентами из центра;
* Обновление системы в некоторых случаях весьма затратно по времени.
  + 1. Основные возможности ITM
       1. Мониторинг параметров серверов и приложений.

Совсем не трудно догадаться, что это – основная функция любой системы мониторинга. За сбор информации, как упоминалось, отвечают агенты. В большинстве своем они устанавливаются на сервер, который необходимо мониторить. Однако существуют агенты, собирающие параметры удаленно. Ознакомиться с ним можно на сайте IBM. С помощью Agent Builder всегда можно разработать уникального агента, отвечающего всем потребностям заказчика (см. Рис. 3).

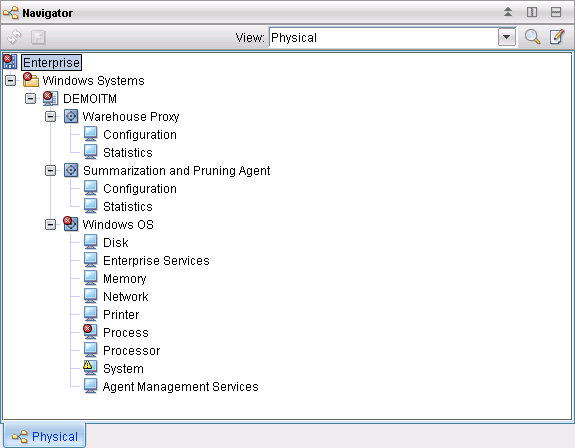


Рис. 3 Управляемые системы и агенты

После того как параметры поступают в TEMS, их значения сравниваются с пороговыми и при превышении порога срабатывают ситуации. Разумеется, все пороги можно менять и создавать новые ситуации (см. Рис. 4).

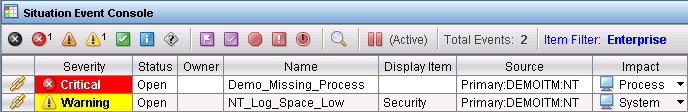


Рис. 4 Активные ситуации

Все вышеописанные функции хороши, при появлении проблем администраторы смогут оперативно решить их, но ведь иногда возникновения проблемы можно и вовсе избежать. Для целей предиктивного анализа в ITM используется TPA (Tivoli Performance Analyser). Для его работы нужен большой объем информации для анализа, т.е. построить точный прогноз, основываясь на данных за неделю, невозможно.

Основные моменты по реализации функции мониторинга:

* Существует большое количество агентов мониторинга и возможность разработки собственных агентов. Вместе с агентами поставляется достаточно объемный набор заранее настроенных ситуаций;
* При установке некоторых агентов приходится перезагружать программную часть TEMS/TEPS.
  + - 1. Сбор исторической информации

Эта функция весьма полезна функция для составления отчетности руководству.

Историческая информация собирается в БД Warehouse, а затем на основе этой информации строятся всевозможные отчеты. Усреднением и очисткой исторических данных занимается Summarization & Pruning агент. В качестве движка отчетности выступает Cognos, работа с отчетами осуществляется через Tivoli Integrated Portal – это единый портал для управления всеми продуктами линейки Tivoli, по крайней мере, так он позиционируется. Также историческая информация используется для предиктивного анализа, упомянутом в предыдущем разделе (см. Рис. 5).

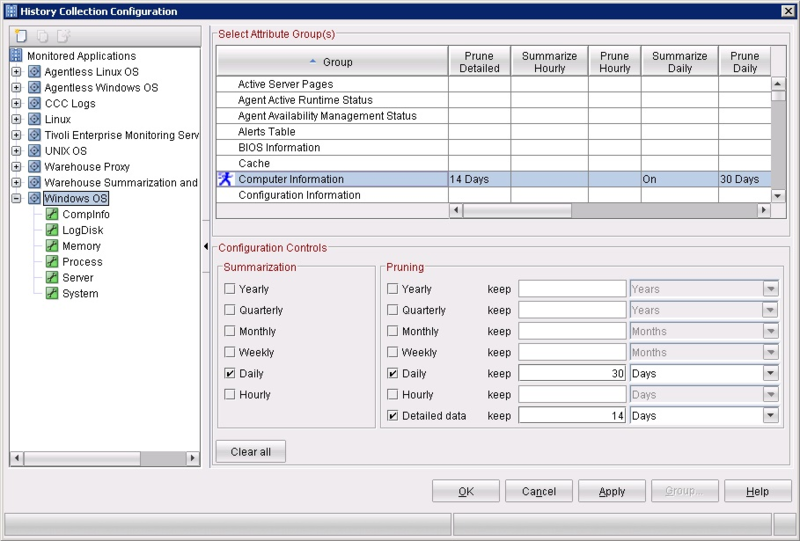
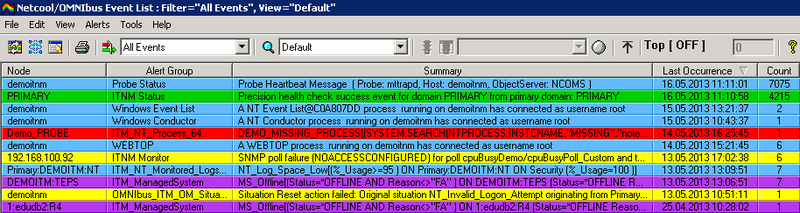


Рис. 5 Конфигурация сбора исторической информации

* + - 1. Интеграция

ITM можно интегрировать с обширным количеством продуктов. То могут быть системы мониторинга, системы обработки событийной информации, любые системы которые, так или иначе, служат для работы с ситуациями, событиями или параметрами мониторинга. Логическим продолжением для работы с ситуациями является их передача в систему обработки событийной информации. В линейке Tivoli – это Netcool/OMNIbus (см. Рис. 6).

Рис. 6 Система обработки событийной информации

Дальше уже возможна интеграция с системами Service Desk и системами контроля качества предоставления сервисов, например, TBSM (Tivoli Business Service Manager).

Насчет интеграции можно кратко сказать следующее:

— При желании ITM интегрируется практически с любым продуктом;

— Интеграция не всегда проходит гладко, но в целом нельзя выделить существенные недостатки.

* + 1. Заключение.

Система описана как можно более сжато и объективно. Обзор написан в общих чертах и не отягощен обилием технических подробностей. Ниже представлена краткая рецензия на то, что из себя на данный момент представляет ITM:

* Система мониторинга огромного количества разнородных компонентов информационной инфраструктуры;
* Подходит для мониторинга как небольшой, так и огромной инфраструктуры;
* Содержит все необходимое для создания отчетов любой сложности;
* Без проблем интегрируется с большинством систем;
* Является не самым дешевым и достаточно требовательным к ресурсам ПО.

1. Исследование существующей системы мониторинга
   1. Перечень систем, подлежащих мониторингу

АС АПВО – Автоматизированная система оперативного анализа планирования и выполнения «окон» на железных дорогах

АС КМО – Автоматизированная система ведения актов комиссионных месячных осмотров и контроля за устранением неисправностей

АСУ ЗМ – Подсистема учета замечаний машинистов

АСУ П – Автоматизированная система управления путевым хозяйством

АСУ Ш2 – Автоматизированная система управления хозяйством сигнализации, централизации и блокировки

АСУ Э – Автоматизированная система управления хозяйством электрификации и электроснабжения

АСУТ НБД – подсистема учета, анализа и расследования нарушений безопасности движения по результатам расшифровки скоростемерных лент

ГИС РЖД – Автоматизированная система ведения геоинформационной базы данных в увязке с параметрами работы и развития ОАО «РЖД»

ПО ГО – Программное обеспечение проведения генеральных осмотров пути

ЕК АСУИ – Единая корпоративная автоматизированная система управления инфраструктурой

ЕСМД П – Единая система мониторинга и диагностирования путевого хозяйства

ЕСМД Ш – Единая система мониторинга и диагностирования объектов эксплуатационной инфраструктуры хозяйства автоматики и телемеханики ОАО «РЖД»

ЕСМД Э – Единая система мониторинга и диагностирования хозяйства электрификации и электроснабжения

КАС АНТ – Комплексная автоматизированная система учета, контроля устранения отказов технических средств и анализа их надежности

КАСАТ – Комплексная автоматизированная система учета, расследования и анализа технологических нарушений

* 1. Обзор существующей системы мониторинга

На данный момент реализован мониторинг ресурсов, имеющих веб-приложение, а также мониторинг состояния баз данных на платформах Oracle и SQL.

Опрос состояния ресурсов должен осуществляться в интервале от 10 до 20 минут хранимой процедурой. На сегодняшний день такой опрос не реализован.

Интерфейс представляет из себя таблицу с такими столбцами, как: Система, Web, БД, Интеграция, Количество пользователей, Администраторы и Разработчики.

Для некоторых систем вообще отсутствует какая бы то ни было информация, поля Администраторы и Разработчики не заполняются.

Поле Интеграция не соответствует желаемому представлению. В нем должно быть отражено состояние интеграции двух систем, например:

АСУ ЗМ – ЕК СУИ – ОК

Или

АСУ ЗМ – ЕК СУИ – АВАРИЯ.

Разработка программного кода производится в среде БД Oracle с помощью программы SQL-Navigator for Oracle компании Quest Sowtware Inc.

Как видно из Рис. 7, систему мониторинга необходимо доработать, а визуальное представление и вовсе переработать.



Рис. 7 Визуальное представление имеющейся системы мониторинга

Примеры хранимых процедуры:

* Опрос веб-ресурса

**procedure** WebSystem (vurl **in** **char**, httpsearch **out** **varchar2**, bgcolor **out** **varchar**)

**is**

ignore **boolean**;

*--httpsearch varchar2(200);*

err **varchar2**(2000);

**begin**

**begin**

httpsearch:=utl\_http.request(vurl);

**exception** **when** **others** **then**

err:=**SQLERRM**;

**end** ;

**if** err **is** **not** **null** **or** **instr**(**upper**(httpsearch),'FOUND')>0 **or** **instr**(**upper**(httpsearch),'FAIL')>0

**or** **instr**(**upper**(httpsearch),'Forbidden')>0 **or** **instr**(**upper**(httpsearch),'denied')>0

**or** **instr**(**upper**(httpsearch),'403')>0

**or** httpsearch **is** **null** **then**

*--httpsearch:=vurl||' - АВАРИЯ! '||httpsearch;--||' - '||vurl;*

**if** **instr**(**upper**(httpsearch),'403')>0 **then**

httpsearch:='<A HREF="'||vurl||'" TARGET="\_blank"> АВАРИЯ! <br> <font size=-2>Forbidden</font> </A>';*--'||httpsearch||'-'||err||'*

**else**

httpsearch:='<A HREF="'||vurl||'" TARGET="\_blank"> АВАРИЯ! <br> <font size=-2>'||httpsearch||'</font> </A>';*--'||httpsearch||'-'||err||'*

**end** **if**;

bgcolor:='#ff0000'; *-- красный*

**else**

*-- httpsearch:=vurl||' - ОК';*

httpsearch:='<A HREF="'||vurl||'" TARGET="\_blank"> ОК </A>';

bgcolor:='#009900'; *--зеленый*

**end** **if**;

**end**;

* Опрос БД ОРАКЛ.

**procedure** BDOraSystem (vlink **in** **char**, vbdurl **in** **char**, procent **out** **varchar**, err **out** **varchar2**, bgcolor **out** **varchar2** )

**is**

*--httpsearch varchar2(200);*

*--err varchar2(2000);*

v\_CursorD2 **integer**;

v\_SelectStr2 **varchar2**(2000);

*--procent varchar2(500);*

vused **integer**;

vfree **integer**;

vtotal **integer**;

v\_exe2 **integer**;

vname **varchar**(20);

temp **varchar2**(200);

bgcolor\_temp **varchar**(10);

**begin**

bgcolor\_temp:='';

**begin**

V\_CursorD2:=DBMS\_SQL.OPEN\_CURSOR;

**if** **instr** (vlink,'maximo')>0 **then**

v\_selectStr2:='SELECT 1 from dual@'||vlink;

**else**

v\_selectStr2:='SELECT sum(PERCENT\_SPACE\_USED) FROM V$FLASH\_RECOVERY\_AREA\_USAGE@'||vlink;

**end** **if**;

DBMS\_SQL.PARSE(V\_CursorD2, v\_selectStr2, DBMS\_SQL.**native**);

*--Привязка выходных данных*

DBMS\_SQL.DEFINE\_COLUMN(V\_CursorD2,1,procent,10);

v\_exe2:=DBMS\_SQL.**execute**(V\_CursorD2);

**loop**

**if** DBMS\_SQL.FETCH\_ROWS(V\_CursorD2)=0 **then**

**exit**;

**end** **if**;

DBMS\_SQL.column\_value(V\_CursorD2,1,procent);

**end** **loop**;

DBMS\_SQL.CLOSE\_CURSOR(v\_cursord2);

*-- dbms\_session.close\_database\_link(vlink);*

**exception** **when** **others** **then**

err:=**SQLERRM**;

DBMS\_SQL.CLOSE\_CURSOR(v\_cursord2);

*--dbms\_session.close\_database\_link(vlink);*

**end**;

**if** **instr**(**upper**(err),'FAIL')>0 **or** err **is** **not** **null** **or**

**instr**(**upper**(err),'ORA')>0 **then** procent:='АВАРИЯ! '||**substr**(err,1,256); bgcolor:='#ff0000'; *-- красный*

*--elsif instr(upper(err),'EXIST')>0 then procent:='АВАРИЯ! '*

**end** **if**;

**if** err **is** **null** **then**

**if** procent **is** **null** **then** bgcolor:=''; *--черный*

**elsif** procent **is** **not** **null** **then**

**if** **to\_number**(procent)<=50 **then** bgcolor:='#009900'; *--зеленый*

**elsif** **to\_number**(procent) **between** 50 **and** 80 **then** bgcolor:='#FF9e00'; *--желтый*

**elsif** **to\_number**(procent)>80 **then** bgcolor:='#ff0000'; *-- красный*

**end** **if**;

*-- procent:='Занято - '||procent||'% - ОК';*

procent:='<A HREF="'||vbdurl||'" TARGET="\_blank">'||'FRA - Занято - '||procent||'% - ОК'||'</A>';

**end** **if**;

*--else*

**end** **if**;

temp:=procent; procent:='';

**if** **instr** (vlink,'maximo')<=0 **then**

**begin**

V\_CursorD2:=DBMS\_SQL.OPEN\_CURSOR;

v\_selectStr2:='select df.tablespace\_name "Tablespace",

totalusedspace "Used MB",

(df.totalspace - tu.totalusedspace) "Free MB",

df.totalspace "Total MB",

100 - round(100 \* ( (df.totalspace - tu.totalusedspace)/ df.totalspace))

"Pct. Full"

from

(select tablespace\_name,

round(sum(bytes) / 1048576) TotalSpace

from dba\_data\_files@'||vlink||'

group by tablespace\_name) df,

(select round(sum(bytes)/(1024\*1024)) totalusedspace, tablespace\_name

from dba\_segments@'||vlink||'

group by tablespace\_name) tu

where df.tablespace\_name = tu.tablespace\_name order by "Pct. Full" desc';

DBMS\_SQL.PARSE(V\_CursorD2, v\_selectStr2, DBMS\_SQL.**native**);

*--Привязка выходных данных*

DBMS\_SQL.DEFINE\_COLUMN(V\_CursorD2,1,vname,20);

DBMS\_SQL.DEFINE\_COLUMN(V\_CursorD2,2,vused);

DBMS\_SQL.DEFINE\_COLUMN(V\_CursorD2,3,vfree);

DBMS\_SQL.DEFINE\_COLUMN(V\_CursorD2,4,vtotal);

DBMS\_SQL.DEFINE\_COLUMN(V\_CursorD2,5,procent,10);

v\_exe2:=DBMS\_SQL.**execute**(V\_CursorD2);

**loop**

**if** DBMS\_SQL.FETCH\_ROWS(V\_CursorD2)=0 **then**

**exit**;

**end** **if**;

DBMS\_SQL.column\_value(V\_CursorD2,1,vname);

DBMS\_SQL.column\_value(V\_CursorD2,2,vused);

DBMS\_SQL.column\_value(V\_CursorD2,3,vfree);

DBMS\_SQL.column\_value(V\_CursorD2,4,vtotal);

DBMS\_SQL.column\_value(V\_CursorD2,5,procent);

**exit**;

**end** **loop**;

bgcolor\_temp:=bgcolor;

**if** err **is** **null** **then**

**if** procent **is** **null** **then** bgcolor:=''; *--черный*

**elsif** procent **is** **not** **null** **then**

**if** **to\_number**(procent)<=80 **then** bgcolor:='#009900'; *--зеленый*

**elsif** **to\_number**(procent) **between** 80 **and** 97 **then** bgcolor:='#FF9e00'; *--желтый*

**elsif** **to\_number**(procent)>98 **then** bgcolor:='#ff0000'; *-- красный*

**end** **if**;

*-- procent:='Занято - '||procent||'% - ОК';*

procent:=temp||'<br><A HREF="'||vbdurl||'" TARGET="\_blank">'||vname||'. Всего -'||vtotal||'Gb. Зан. -'||procent||'% '||'</A>';

**end** **if**;

**if** bgcolor\_temp='#ff0000' **or** bgcolor\_temp='#FF9e00' **then** bgcolor :=bgcolor\_temp; **end** **if**;

*--bgcolor :=bgcolor\_temp;*

**end** **if**;

DBMS\_SQL.CLOSE\_CURSOR(v\_cursord2);

**exception** **when** **others** **then**

err:=**SQLERRM**;

DBMS\_SQL.CLOSE\_CURSOR(v\_cursord2);

*--dbms\_session.close\_database\_link(vlink);*

**end**;

**end** **if**

**end**;

* Вывод информации о состояние ресурса на веб-страницу

**procedure** WatchSystem

**is**

ignore **boolean**;

**cursor** list\_new **is**

**select** a.id\_system, a.name\_system, a.full\_name\_system, s.cod\_it||' - '||s.name\_it

**FROM** espp.espp\_system\_info@orcl2.dvgd.rzd a, espp.sd\_event\_group@orcl2.dvgd.rzd b,

espp.sd\_event\_office@orcl2.dvgd.rzd c, espp.espp\_category\_it\_sub@orcl2.dvgd.rzd s

**where** a.id\_group=b.id\_group **and** a.id\_office=c.id\_office

**and** a.id\_sub=s.id\_service(+) **and** c.mcod\_office='АСУИ'

**and** a.id\_system **not** **in** (**select** b.id\_system **from** system\_info b) **order** **by** a.name\_system;

vidsystem system\_info.id\_system%**type**;

vname system\_info.name\_system%**type**;

vfullname system\_info.full\_name\_system%**type**;

vservice system\_info.id\_service%**type**;

err **varchar2**(1000);

schet **number**(10);

**cursor** razmer **is** **SELECT** \* **FROM** V$FLASH\_RECOVERY\_AREA\_USAGE@orcl.dvgd.rzd;

v\_CursorD2 **integer**;

v\_SelectStr2 **varchar2**(2000);

procent **varchar2**(500);

v\_exe2 **integer**;

bgcolor **varchar**(15);

httpsearch **varchar2**(20000);

vurl system\_info.url%**type**;

**begin**

htp.htmlopen;

htp.headopen;

htp.meta ('Content-Type',**NULL**,'text/html; charset=windows-1251');

htp.meta ('refresh', **NULL**, 600);

htp.p('<LINK rel=stylesheet type="text/css" href="/'||url\_path||'/css/style.css">');

htp.title('АС Мониторинг | Автоматизированная система мониторинга информационных систем');

htp.headclose;

schet:=0;

htp.p('<font color="#000080"><font size=+1>Последнее обновление информации о состоянии систем '||**to\_char**(**sysdate**,'dd.mm.yyyy hh24:mi:ss')||'</font>');

**begin**

*--dbms\_session.close\_database\_link('maximo.dvgd.rzd');*

*--dbms\_session.close\_database\_link('esmd.dvgd.rzd');*

*--dbms\_session.close\_database\_link('orcl.dvgd.rzd');*

*--dbms\_session.close\_database\_link('orcl2.dvgd.rzd');*

*--dbms\_session.close\_database\_link('orcl3.dvgd.rzd');*

**open** list\_new; **fetch** list\_new **into** vidsystem, vname, vfullname, vservice; **close** list\_new;

**exception** **when** **others** **then**

err:=**SQLERRM**;

**insert** **into** protokol a (a.date\_err, a.text\_err) **values** (**sysdate**, err); **commit**;

schet:=1;

**end**;

**if** vidsystem **is** **not** **null** **and** schet=0 **then** *-- обновление списка систем*

**insert** **into** system\_info a ( a.id\_system, a.name\_system, a.full\_name\_system, a.id\_service)

**select** a.id\_system, a.name\_system, a.full\_name\_system, s.cod\_it||' - '||s.name\_it

**FROM** espp.espp\_system\_info@orcl2.dvgd.rzd a, espp.sd\_event\_group@orcl2.dvgd.rzd b,

espp.sd\_event\_office@orcl2.dvgd.rzd c, espp.espp\_category\_it\_sub@orcl2.dvgd.rzd s

**where** a.id\_group=b.id\_group **and** a.id\_office=c.id\_office

**and** a.id\_sub=s.id\_service(+) **and** c.mcod\_office='АСУИ'

**and** a.id\_system **not** **in** (**select** b.id\_system **from** system\_info b) **order** **by** a.name\_system;

**commit**;

**end** **if**;

htp.tableopen;

htp.tableRowOpen ();

htp.tableHeader ('Система ', 'center',**null**,**null**,**null**,**null**,'width="8%" ');

htp.tableHeader ('WEB', 'center',**null**,**null**,**null**,**null**,'width="2%"');

htp.tableheader ('БД', 'center', **null**,**null**,**null**,**null**,'width="15%" title=""');

htp.tableheader ('Интеграция', 'center', **null**,**null**,**null**,**null**,'width="5%" title=""');

htp.tableheader ('Кол-во пользователей', 'center', **null**,**null**,**null**,**null**,'width="5%" title=""');

htp.tableheader ('Администраторы', 'center', **null**,**null**,**null**,**null**,'width="15%" title=""');

htp.tableheader ('Разработчики', 'center', **null**,**null**,**null**,**null**,'width="15%" title=""');

**for** i **in** (**SELECT** a.id\_system, a.name\_system, a.db\_link, a.is\_oracle, a.url, a.accompany, a.master,

a.url\_count, a.is\_web, a.db\_url, a.count\_users

**FROM** system\_info a **where** a.is\_deleted **is** **null** **order** **by** a.name\_system) *--order by a.id\_system*

**loop**

htp.p('<tr><td>');

err:='';

htp.anchor2(i.url,i.name\_system,ctarget=>'\_blank');

htp.p('</td>');

*-- vurl:=i.url;*

**if** i.is\_web **is** **not** **null** **then**

websystem(i.url, httpsearch, bgcolor );

**end** **if**;

htp.tableData (httpsearch||' '||err, 'center', **null**,**null**,**null**,**null**,'width="5%" bgcolor="'||bgcolor||'" '); *-- веб*

err:=''; httpsearch:=''; bgcolor:='';

**if** i.is\_oracle **is** **not** **null** **then** *--and i.id\_system!=89*

*--err:='';*

BDOraSystem(i.db\_link, i.db\_url, procent, err, bgcolor );

**end** **if**;

**if** err **is** **not** **null** **then** procent:=err; **end** **if**;

htp.tableData ( procent, 'center', **null**,**null**,**null**,**null**,'width="5%" bgcolor="'||bgcolor||'" ');

htp.p('<td>');

htp.anchor2(i.url\_count||i.id\_system,i.name\_system,ctarget=>'\_top'); *--интеграция*

htp.p('<td>');

htp.anchor2('http://dvgd-armapp-01.dvgd.oao.rzd/pls/sapporo/nci.otchet?vidsystem='||i.id\_system,i.count\_users,ctarget=>'\_blank'); *--число пользователей*

htp.tableData ( i.accompany, 'center', **null**,**null**,**null**,**null**,'width="5%" bgcolor="" ');

htp.tableData ( i.master, 'center', **null**,**null**,**null**,**null**,'width="5%" bgcolor="" ');

procent:=''; bgcolor:=''; *-- красный*

**end** **loop**;

htp.tableclose;

htp.anchor('javascript:history.back();','<font color="#0000FF">Назад</font>');

htp.htmlclose;

**EXCEPTION**

**WHEN** **others** **THEN**

htp.p('<html><body>'||**SQLERRM**||'</body></html>');

htp.anchor('javascript:history.back();','<font color="#0000FF">Назад</font>');

htp.bodyclose;

htp.htmlclose;

**end**;

Имеются и другие разработанные процедуры.

В переработанной таблице веб-приложения должна быть представлена следующая информация:

* доступность ресурса (аварийное состояние или стабильная работа);
* данные о состоянии базы данных (свободное место, количество подключений);
* состояние интеграции с другими ресурсами (авария или интеграция без ошибок);
* количество пользователей, работающих с ресурсом;
* администратор ресурса;
* контактная информация о разработчике ресурса;
* тип ресурса (веб-приложение или исполняемый файл).

В нынешней системе мониторинга основную сложность представляет анализ журналов интеграции систем – практически все они представлены в текстовом виде, и для локализации сбоя сотрудник вынужден вычленять ошибки из каждого лога вручную. Такой кропотливый перебор текстового документа занимает много времени, которое можно было бы употребить на более важные производственные задачи. Автоматизированный и регулярный поиск ошибок, а также их запись в базу данных сэкономят время сотрудникам и упростят анализ аварийных ситуаций.

Исходя из вышеописанного, в веб-интерфейсе необходимо предусмотреть поле, нажав на которое можно будет ознакомиться с перечнем ошибок, приведших к аварийному состоянию интеграции двух систем. Для этого ознакомимся с логами баз данных и рассмотрим возникающие в них ошибки.

* 1. Анализ ошибок в логах
     1. Что такое лог

Файл регистрации, протокол, журнал или лог (англ. log) — файл с записями о событиях в хронологическом порядке. Различают регистрацию внешних событий и протоколирование работы самой программы – источника записей (хотя часто всё записывается в единый файл).

При активном использовании сервера пользователями или очень подробном журналировании или просто продолжительном времени работы ПО, в журнале становится очень много записей, что приводит к следующим затруднениям:

* файл лога потребляет много дискового пространства;
* большой файл лога становится тяжелым для чтения и анализа.

Таким образом бывает необходима ротации логов – усечение текущего лога, архивирование или удаление предыдущих накопленных данных:

* по временному промежутку (каждый час, каждый день, за последние сутки/неделю) в новый файл.
* по достижению определенного объема или количества записей.
* циклический журнал – методика хранения какого-то определенного объема или количества записей, при этом новые записи удаляют самые старые записи, на базе такой структуры как кольцевой буфер.
* Каждый новый запуск экземпляра ПО создает новый файл лога или переименовывает/откладывает предыдущий файл лога.

Если автоматическая ротация журнала не предусмотрена при разработке ПО, то эта задача ложится на системного администратора, которые автоматизируют её с помощью планировщиков, или специализированых утилит, как например logrotate.

Логи ХИВЦ, рассматриваемые на наличие ошибок, являются логами приложений. Они хранится в таблицах или в файлах. Ротация логов происходит по временному промежутку, а именно за сутки.

* + 1. Места возникновения ошибок

Найдем места возникновения сбоев в файлах с разрешением .log для различных систем. Разработчиком принят такой порядок наименования файлов, что название файла пишется через знак нижнего подчеркивания, а в конце пишется дата в формате «год месяц день» без пробелов.

**АСК\_ПС\_годмесяцдень**

1. [2014-02-08T00:35:03] R

[2014-02-08T00:35:03] R

[2014-02-08T00:35:03] R

[2014-02-08T00:35:03] R

[2014-02-08T00:35:03] R

[2014-02-08T00:35:03] R

[2014-02-08T00:35:03] Не найдена точка КТС

1. [2014-02-08T00:56:01] Вызов: ОК, Последняя позиция: -1
2. [2014-02-09T01:21:31] Timeout expired. The timeout period elapsed prior to completion of the operation or the server is not responding.

[2014-02-09T01:21:31] Синхронизация завершена с ошибками

1. [2014-02-09T05:01:22] Unable to connect to the remote server

[2014-02-09T05:01:22] Синхронизация завершена с ошибками

**ЕСМД-Ш КЗМ\_годмесяцдень**

**ЕСМД-Ш ОКНА\_**

[2014-02-08T23:05:03] Отобрано строк: 0 из БД Oracle

[2014-02-08T23:05:03] Разбор окон->Загрузка в #ASUI\_OKNA\_RAZBOR [0/0]

[2014-02-08T23:05:03]

Ошибка НСИ ГИД УРАЛ: Основной объект отсутствует в ЦНСИ - 0:0; ID\_строки = 134143; Date\_Create = 1970-01-06T08:27:16; ID\_GID = 954; Stan1\_Id = 0; ЕСР код1 = 98250;Stan2\_Id = 0; ЕСР код2 = 0

Ошибка НСИ ГИД УРАЛ: Основной объект отсутствует в ЦНСИ - 0:КЕНАДА; ID\_строки = 164229; Date\_Create = 1970-01-06T10:13:17; ID\_GID = 954; Stan1\_Id = 0; ЕСР код1 = 96654;Stan2\_Id = 21119; ЕСР код2 = 96660

Ошибка НСИ ГИД УРАЛ: Основной объект отсутствует в ЦНСИ - ДАТТА:0; ID\_строки = 164230; Date\_Create = 1970-01-06T10:15:03; ID\_GID = 954; Stan1\_Id = 21118; ЕСР код1 = 96652;Stan2\_Id = 0; ЕСР код2 = 96654

Ошибка НСИ ГИД УРАЛ: Основной объект отсутствует в ЦНСИ - УГЛОВАЯ:АРТЕМ-ПРИМ I; ID\_строки = 196106; Date\_Create = 1970-01-06T09:46:04; ID\_GID = 955; Stan1\_Id = 21283; ЕСР код1 = 98220;Stan2\_Id = 21285; ЕСР код2 = 98240

Ошибка НСИ ГИД УРАЛ: Основной объект отсутствует в ЦНСИ - 0:АРТЕМ-ПРИМ III; ID\_строки = 196210; Date\_Create = 1970-01-06T10:16:36; ID\_GID = 954; Stan1\_Id = 0; ЕСР код1 = 98250;Stan2\_Id = 21286; ЕСР код2 = 98260

Ошибка НСИ ГИД УРАЛ: Основной объект отсутствует в ЦНСИ - АРТЕМ-ПРИМ I:0; ID\_строки = 436776; Date\_Create = 1970-01-06T10:27:25; ID\_GID = 954; Stan1\_Id = 21285; ЕСР код1 = 98240;Stan2\_Id = 0; ЕСР код2 = 98250

Ошибка НСИ ГИД УРАЛ: Основной объект отсутствует в ЦНСИ - РАССЫПНАЯ ПАДЬ:ГРОДЕКОВО КИТ; ID\_строки = 591235; Date\_Create = 1970-01-06T11:00:59; ID\_GID = 954; Stan1\_Id = 21378; ЕСР код1 = 98922;Stan2\_Id = 21379; ЕСР код2 = 98930

Ошибка НСИ ГИД УРАЛ: Основной объект отсутствует в ЦНСИ - ВОДОПАДНОЕ:ЛЕЙТ ГОРДЕЕВ; ID\_строки = 1393062; Date\_Create = 1970-01-06T12:25:18; ID\_GID = 954; Stan1\_Id = 21304; ЕСР код1 = 98390;Stan2\_Id = 21305; ЕСР код2 = 98400

Ошибка НСИ ГИД УРАЛ: Основной объект отсутствует в ЦНСИ - УДАРНЫЙ:0; ID\_строки = 5007175; Date\_Create = 2013-10-27T23:32:45; ID\_GID = 0010100096; Stan1\_Id = 21035; ЕСР код1 = 96141;Stan2\_Id = 0; ЕСР код2 = 96139

Ошибка НСИ ГИД УРАЛ: Основной объект отсутствует в ЦНСИ - КИПАРИСОВО:НАДЕЖДИНСКАЯ; ID\_строки = 5962246; Date\_Create = 1970-01-06T21:26:18; ID\_GID = 954; Stan1\_Id = 21335; ЕСР код1 = 98670;Stan2\_Id = 21332; ЕСР код2 = 98650

Ошибка НСИ ГИД УРАЛ: Основной объект отсутствует в ЦНСИ - ЛЕЙТ ГОРДЕЕВ:СЕРГЕЕВКА; ID\_строки = 6881547; Date\_Create = 1970-01-06T22:42:10; ID\_GID = 954; Stan1\_Id = 21305; ЕСР код1 = 98400;Stan2\_Id = 21306; ЕСР код2 = 98411

Ошибок разбора окон: 11

**ЕСМД-Ш\_**

1. [2014-02-05T00:00:53] system.exception: the remote server returned an error: (500) internal server error.
2. BMXAA4082E - Ошибки при проверке или в списке условия where домена crossover, заданного для CLASSID\_ELEMENT.
3. bmxaa0031e - объект incident - это объект только-для-чтения

**ЕТБ-Ш Полная сверка\_**

**ЕТБ-Ш\_**

1. [2014-02-05T15:08:05] System.Exception: The remote server returned an error: (500) Internal Server Error.
2. [2014-02-05T16:09:59] Синхронизация завершена с ошибками (в конце лога)
3. BMXAA1496E - PERSON не существует. Невозможно обновить или удалить.
4. BMXAA4191E - Табличное значение 9900 недопустимо.
5. BMXAA4214E - Произошла неизвестная ошибка.

Обратитесь за помощью к системному администратору.

1. BMXAA5359E - Прежде чем добавлять классификацию, введите спецификацию объекта.
2. BMXAA4211E - При работе в ASSETSPEC Атрибут=VIDTOK\_RZ1\_ID Актив=A19063867 Раздел= Площадка=96 ID линейной спецификации=0 произошла ошибка базы данных номер 1.

**ТС-2-Загрузка\_**

**ТС-2-Инциденты\_**

1. [2014-02-05T18:00:27] Ошибка вызова сервиса: RZD\_SH\_WO\_EXCH

System.Exception: The remote server returned an error: (500) Internal Server Error.

'>' is an unexpected token. The expected token is '"' or '''. Line 1, position 50.

1. [2014-02-05T18:00:35] Ошибка вызова сервиса: RZD\_SH\_WO\_EXCH

System.Exception: The remote server returned an error: (500) Internal Server Error.

1. [2014-02-05T19:00:20] Синхронизация завершена с ошибками
2. [2014-02-05T19:20:19] Прерывание выполнения схемы из-за критичной ошибки (CheckExceptionBreak: System.Exception: The remote server returned an error: (500) Internal Server Error.
3. BMXAA6089E - Не удалось загрузить хэндл Enterprise JavaBeans (EJB).
4. BMXAA4200E - Запись обновлена другим пользователем. Повторите выборку и попробуйте еще раз.

**ТС-2-НСИ\_**

1. [2014-02-05T00:00:19] System.Exception: The remote server returned an error: (500) Internal Server Error.
2. BMXAA4190E - Описание периодичности выполнения работ 127 нет в списке значений

**ТС-2-Оперативный\_**

1. [2014-02-06T00:02:38] The remote server returned an error: (500) Internal Server Error.
2. [2014-02-06T02:41:00] Синхронизация завершена с ошибками
3. BMXAA2627E - Работник P909206 недопустим.

Если проанализировать все найденные ошибки в логах за несколько дней, то можно прийти к выводу о том, что универсальный алгоритм поиска ошибок в файле должен реагировать на ряд слов или частей слов: «ошибк…», «error», «bmxaa» и быть регистронезависимым.

Необходимо предусмотреть возможность корректировки существующих, а также внесения новых ключевых слов. Помимо этого, длина сообщения об ошибке должна быть настраиваемой.

Следует определиться с тем, каким образом строка с ошибкой будет внесена в базу данных. Логичнее всего предположить, что запись об ошибке будет сделана, основываясь на дате и времени. Необходимо найти строку, содержащую одно из ключевых слов, затем дойти до начала строки с датой и внести описание ошибки в базу данных. Не стоит забывать, что для удобства следует проработать структуру таблицы базы данных, куда будут внесены ошибки.

Поля таблицы LogErrors:

|  |  |  |
| --- | --- | --- |
| Поле | Расшифровка | Тип |
| LogName | Название лога с ошибкой | Varchar |
| ErrorDate | Дата ошибки (системное время обнаружения ошибки, до минут) | Date |
| ErrorText | Текст ошибки (фрагмент) | 1000 |
| ErrorNumber | Количество повторений ошибки | Int |

Программа должна быть универсальной, иметь файл настроек. Запуск программы будет осуществляться по системному заданию Windows. В настройках указывается имя файла лога, ключевые слова ошибок, строка соединения с БД, название таблицы куда записывается фрагмент ошибки. Файл temp содержит номер последней прочитанной строки.

В качестве языка программирования системы будет выступать Python. В нем есть все необходимые для разработки библиотеки.