**3.37. Оценка производительности**

Подсистема «Оценка производительности» предназначена для оценки интегральной производительности системы по методике APDEX.

**Настройка**

Для настройки подсистемы «Оценка производительности» на сбор данных замеров производительности необходимо принять решение по поводу состава ключевых операций, время выполнения которых требуется замерять и отслеживать.

Ключевые операции – это такие операции, в процессе выполнения которых пользователь непрерывно ожидает отклика от информационной системы. Например: проведение некоторого документа, запись некоторого справочника, загрузка данных и т. п. Ключевые операции не могут включать в себя интерактивные действия пользователя.

Длительные ключевые операции – это такие ключевые операции, время выполнения которых зависит от количества обрабатываемых данных. Например: обработка нескольких документов – допустимо тратить на обработку одного документа некоторое время, однако на обработку 1000 документов времени уйдет больше, чем на обработку 10. В таком случае необходимо при выполнении ключевой операции ориентироваться не на время обработки всех документов, а на время обработки одного документа. В некоторых случаях в ходе длительной ключевой операции может выполняться несколько шагов, некоторые из которых зависят от объема обрабатываемых данных. В таком случае для каждого шага необходимо фиксировать время выполнения, чтобы при выполнении анализа легко определить, где именно возникла проблема.

Ключевые операции задаются по названию. Настройку приоритета и целевого времени ключевых операций должен будет выполнить администратор информационной базы, после того как в ИБ будут собраны замеры. Настройку приоритета и целевого времени ключевых операций выполнять не обязательно. Данная настройка необходима для последующего анализа замеров.

Для анализа замеров необходимо использовать отчет ОценкаПроизводительности. Поскольку ключевых операций может быть много, для целей анализа удобно объединять интересующие ключевые операции в группы с помощью справочника ПрофилиКлючевыхОпераций. Там же необходимо назначить целевое время и приоритет, которые будут учитываться при построении отчета.

В коде конфигурации, в местах выполнения ключевых операций, надо вставить вызовы подсистемы «Оценка производительности» для замера времени их выполнения. При этом имя ключевой операции указывается в качестве параметра процедур начала и/или окончания замера времени.

Подсистема «Оценка производительности» поддерживает пять сценариев работы с замерами времени:

1. Начать замер на клиенте и автоматически завершить замер на клиенте. Кроме того, до момента завершения замера можно программно доопределить имя ключевой операции и комментарий замера времени (если они еще не были известны на момент начала замера).

2. Начать замер на сервере и явно завершить его на сервере (по усмотрению разработчика).

3. Начать замер на клиенте и явно завершить его на клиенте (по усмотрению разработчика). Кроме того, до момента завершения замера можно программно доопределить имя ключевой операции и комментарий замера времени (если они еще не были известны на момент начала замера).

4. Начать замер длительной ключевой операции на клиенте и явно завершить его на клиенте. При этом необходимо фиксировать выполнение каждого шага с указанием количества данных.

5. Начать замер длительной ключевой операции на сервере и явно завершить его на сервере. При этом необходимо фиксировать выполнение каждого шага с указанием количества данных.

Для выполнения замеров ключевой операции по сценариям № 1 и № 3 необходимо вызвать функцию НачатьЗамерВремени(Истина, Неопределено) общего модуля ОценкаПроизводительностиКлиент, которая вернет уникальный идентификатор замера. Если параметр функции АвтоЗавершение = Истина, то замер времени завершится автоматически. Если же параметр функции АвтоЗавершение = Ложь, то для завершения замера времени необходимо вызвать процедуру ЗавершитьЗамерВремени(УИДЗамера) общего модуля ОценкаПроизводительностиКлиент, где УИДЗамера – уникальный идентификатор, который вернула функция НачатьЗамерВремени. В случае если КлючеваяОперация = Неопределено, необходимо до автозавершения замера установить имя ключевой операции вызовом процедуры УстановитьКлючевуюОперациюЗамера(УИДЗамера, КлючеваяОперация) общего модуля ОценкаПроизводительностиКлиент. В случае если до завершения автозамера разработчик не установил имя ключевой операции, то данный замер при автозавершении будет удален из клиентского буфера и не будет записан в регистр сведений.

Для выполнения замеров ключевой операции по сценарию № 2 необходимо вызвать функцию НачатьЗамерВремени общего модуля ОценкаПроизводительности. Затем для завершения замера времени вызывать функцию ЗакончитьЗамерВремени общего модуля ОценкаПроизводительности.

Для выполнения замеров ключевой операции по сценарию № 4 и 5 необходимо вызвать функцию НачатьЗамерДлительнойОперации(ИмяКлючвойОперации) общего модуля ОценкаПроизводительностиКлиент или ОценкаПроизводительности соответственно, которая вернет описание замера – соответствие, хранящее данные замера. Для фиксации шага необходимо вызвать метод ЗафиксироватьЗамерДлительнойОперации(ОписаниеЗамера, КоличествоДанных, ИмяШага). Если вызвать ЗафиксироватьЗамерДлительнойОперации повторно с тем же именем шага, то данные просуммируются с результатом предыдущей фиксации шага. Для завершения замера необходимо вызвать метод ЗакончитьЗамерДлительнойОперации(ОписаниеЗамера, КоличествоДанных, ИмяШага) – метод завершает замер и осуществляет запись коллекции замеров: полный, удельный и для каждого шага. При этом для удельного замера и замеров шагов операции время записывается как отношение потраченного времени на выполнение шага к количеству обработанных данных. Имена шагов формируются по принципу ИмяКлючвойОперации.ИмяШага.

Пример замера времени проведения документа на клиенте с разделением на замеры корректного проведения и возникновением исключения при проведении (сценарий № 1):

[Копировать в буфер обмена](javascript:)

// Модуль формы документа \_ДемоЗаказПокупателя

&НаКлиенте

Перем ИдентификаторЗамераПроведение, ИдентификаторЗамераПроведениеНеНужнаРегистрацияОшибки;

&НаКлиенте

Процедура ПередЗаписью(Отказ, ПараметрыЗаписи)

Если ПараметрыЗаписи.РежимЗаписи = РежимЗаписиДокумента.Проведение Тогда

ИдентификаторЗамераПроведение = ОценкаПроизводительностиКлиент.НачатьЗамерВремени(Истина, "\_ДемоПроведениеДокументаОшибка");

ОценкаПроизводительностиКлиент.УстановитьПризнакОшибкиЗамера(ИдентификаторЗамераПроведение, Истина);

ИдентификаторЗамераПроведениеНеНужнаРегистрацияОшибки = ОценкаПроизводительностиКлиент.НачатьЗамерВремени(Истина);

КонецЕсли;

// Далее следует текст обработчика формы -->>

КонецПроцедуры

&НаКлиенте

Процедура ПослеЗаписи(ПараметрыЗаписи)

ОценкаПроизводительностиКлиент.УстановитьКлючевуюОперациюЗамера(ИдентификаторЗамераПроведение, "\_ДемоПроведениеДокумента");

ОценкаПроизводительностиКлиент.УстановитьПризнакОшибкиЗамера(ИдентификаторЗамераПроведение, Ложь);

ОценкаПроизводительностиКлиент.УстановитьКлючевуюОперациюЗамера(ИдентификаторЗамераПроведениеНеНужнаРегистрацияОшибки, "\_ДемоПроведениеДокументаНеНужнаРегистрацияОшибки");

// Далее следует текст обработчика формы -->>

КонецПроцедуры

Пример замера времени выполнения регламентного задания (сценарий № 2):

[Копировать в буфер обмена](javascript:)

// Процедура обработки регламентного задания по выгрузке данных

Процедура ЭкспортОценкиПроизводительности(КаталогиЭкспорта) Экспорт

ДатаНачала = ОценкаПроизводительности.НачатьЗамерВремени();

// Далее следует текст выполнения регламентного задания -->>

ОценкаПроизводительности.ЗакончитьЗамерВремени("ЭкспортОценкиПроизводительности", ДатаНачала);

КонецПроцедуры

Пример замера времени длительной операции на клиенте – формирование отчета в фоновом задании (сценарий № 3):

[Копировать в буфер обмена](javascript:)

&НаКлиенте

Перем УникальныйИдентификаторЗамера, УникальныйИдентификаторФоновогоЗадания;

&НаКлиенте

Процедура СформироватьОтчет()

УникальныйИдентификаторЗамера =

ОценкаПроизводительностиКлиент.НачатьЗамерВремени(Ложь, "ОтчетПоПродажам");

// Далее следует текст запуска формирования отчета на сервере в фоновом задании -->>

ПодключитьОбработчикОжидания("ПроверкаФормированияОтчета", 1);

КонецПроцедуры

&НаКлиенте

Процедура ПроверкаФормированияОтчета()

РезультатПроверки = ПроверкаВыполненияФоновогоЗаданияНаСервере(УникальныйИдентификаторФоновогоЗадания);

Если РезультатПроверки.Состояние <> "Активно" Тогда

ОтключитьОбработчикОжидания("ПроверкаФормированияОтчета");

Если РезультатПроверки.Состояние = "Завершено" Тогда

// Далее следует текст вывода отчета на сервере в фоновом задании -->>

ОценкаПроизводительностиКлиент.ЗавершитьЗамерВремени(УникальныйИдентификаторЗамера);

Иначе

ВызватьИсключение(ПодробноеПредставлениеОшибки(ИнформацияОбОшибке()));

КонецЕсли;

КонецЕсли;

КонецПроцедуры

Пример замера времени длительной операции на клиенте – создание элементов справочников в фоновом задании (сценарий № 4 и 5):

[Копировать в буфер обмена](javascript:)

Процедура ВыполнитьДействие(Параметры, АдресРезультата) Экспорт

ОписаниеЗамера = ОценкаПроизводительности.НачатьЗамерДлительнойОперации("ДемоЗамерДлительнойОперации");

КоличествоКонтрагентов = Параметры.КоличествоКонтрагентов;

КоличествоБанковскихСчетовКонтрагента = Параметры.КоличествоБанковскихСчетовКонтрагента;

УдалитьСозданные = Параметры.УдалитьСозданные;

НаименованиеКонтрагента = НСтр("ru = 'Контрагент %1'");

НаименованиеСчета = НСтр("ru = 'Банковский счет контрагента %1'");

МассивОбъектов = Новый Массив;

НачатьТранзакцию();

Попытка

Для СчетчикКонтрагента = 1 По КоличествоКонтрагентов Цикл

КонтрагентОбъект = Справочники.\_ДемоКонтрагенты.СоздатьЭлемент();

КонтрагентОбъект.Наименование = СтроковыеФункцииКлиентСервер.ПодставитьПараметрыВСтроку(НаименованиеКонтрагента, Формат(СчетчикКонтрагента, "ЧГ="));

КонтрагентОбъект.Записать();

ОценкаПроизводительности.ЗафиксироватьЗамерДлительнойОперации(ОписаниеЗамера, 1, "ЗаписьКонтрагента");

МассивОбъектов.Добавить(КонтрагентОбъект.Ссылка);

Для СчетчикСчета = 1 По КоличествоБанковскихСчетовКонтрагента Цикл

СчетОбъект = Справочники.\_ДемоБанковскиеСчета.СоздатьЭлемент();

СчетОбъект.Владелец = КонтрагентОбъект.Ссылка;

СчетОбъект.Наименование = СтроковыеФункцииКлиентСервер.ПодставитьПараметрыВСтроку(НаименованиеСчета, Формат(СчетчикКонтрагента, "ЧГ="));

СчетОбъект.Записать();

ОценкаПроизводительности.ЗафиксироватьЗамерДлительнойОперации(ОписаниеЗамера, 1, "ЗаписьБанковскогоСчета");

КонецЦикла;

КонецЦикла;

ЗафиксироватьТранзакцию();

Исключение

ОтменитьТранзакцию();

КонецПопытки;

Если УдалитьСозданные Тогда

Для Каждого Элемент Из МассивОбъектов Цикл

ЭлементОбъект = Элемент.ПолучитьОбъект();

ЭлементОбъект.Удалить();

КонецЦикла;

КонецЕсли;

ОценкаПроизводительности.ЗакончитьЗамерДлительнойОперации(ОписаниеЗамера, МассивОбъектов.Количество(), "УдалениеКонтрагентов");

КонецПроцедуры

Пример реализации ключевых операций см. в демонстрационной конфигурации.

При завершении работы пользователя часть выполненных клиентских замеров может быть потеряна. Связано это с тем, что клиентские замеры хранятся на клиенте, а на сервер передаются периодически, с целью минимизации клиент-серверных вызовов. Периодичность задается константой ОценкаПроизводительностиПериодЗаписи (по умолчанию, раз в минуту).

Необходимо разместить в командном интерфейсе ответственного за оценку производительности константу ВыполнятьЗамерыПроизводительности и обработку ОценкаПроизводительности. См. пример размещения в демонстрационной конфигурации в группе Оценка производительности формы ПоддержкаИОбслуживание обработки ПанельАдминистрированияБСП.

Для того чтобы система автоматически начала собирать замеры производительности, необходимо константе ВыполнятьЗамерыПроизводительности установить значение Истина в процедуре – обработчике обновления и первоначального заполнения ИБ.

**Настройка прав доступа пользователей**

Роли, используемые для работы с подсистемой:

Таблица 3.71.

| № | Роли и их назначение |
| --- | --- |
|  | НастройкаИОценкаПроизводительности  Анализ результатов замера производительности ключевых операций с помощью обработки ОценкаПроизводительности |

**Использование при разработке конфигурации**

**Настройка обмена данными**

Все объекты подсистемы не должны включаться в планы обмена распределенной ИБ (РИБ), т. к. требования и условия работы в каждой базе могут быть свои. Однако в некоторых задачах, в которых распределенная ИБ (РИБ) рассматривается как единая и однородная система в плане производительности, имеет смысл в планы обмена включить все объекты подсистемы, содержащие данные.