Для выполнения нагрузочного тестирования использовались средства MS VisualStudio 2010 Ultimate.

**Capacity Testing**

Были записаны 3 автоматизированных запроса к тестируемому сервису по SOAP протоколу и объединены каждый со своей частотой повторения в нагрузочный тест.

При этом нагрузочном тесте число пользователей меняется от 1 до 11, прибавляя каждые 5 секунд по 2 пользователя. При этом каждый пользователь генерирует более 10 запросов в секунду с равномерным распределением. Тестирование длится 1 минуту, включая 5 секунд прогрева.

Для каждого запроса проверяется наличие возвращаемого результата и предельное время отклика (должно быть менее 1 секунды).

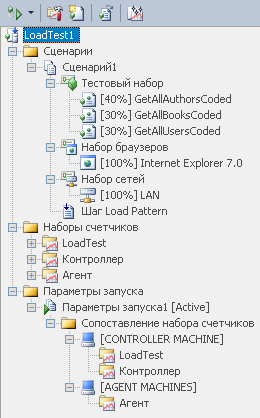
Тестирование показало, что 2 запроса (на получение списка авторов) из 989 превысили предельное время отклика. Это составляет 0,2% при интенсивности поступления этого типа запросов 18 в секунду.

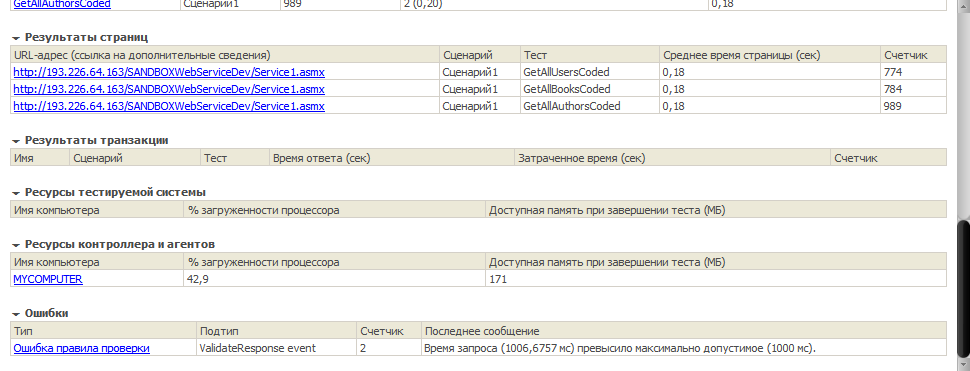
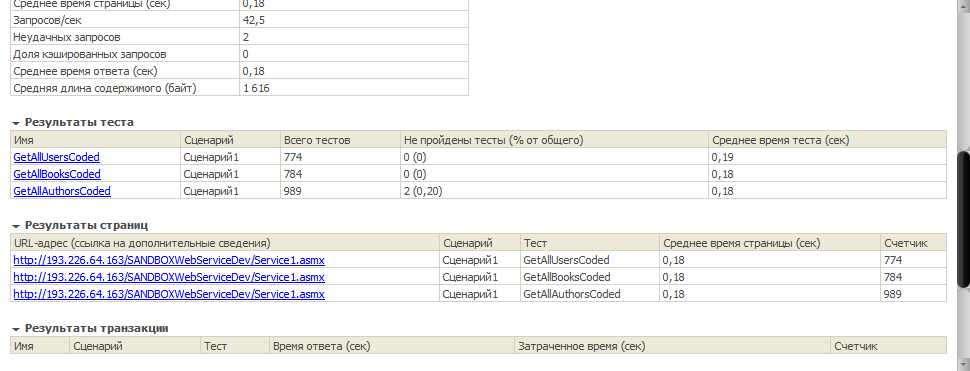
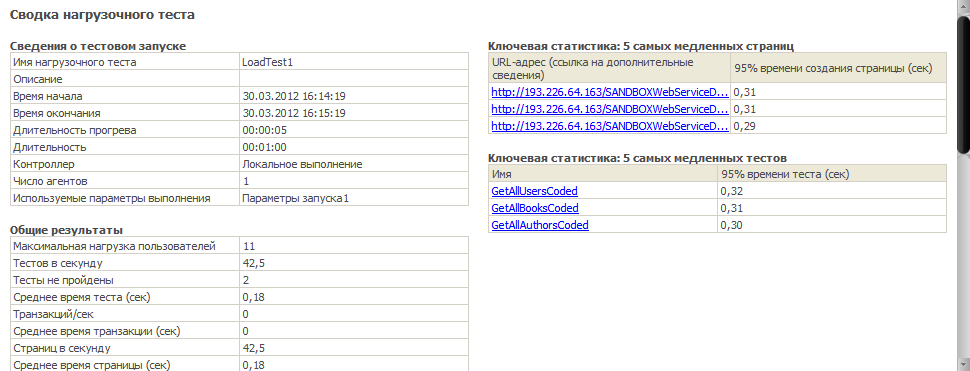
Таким образом в контексте данной задачи этот результат можно считать приемлемым, а предел надежности еще не достигнут.

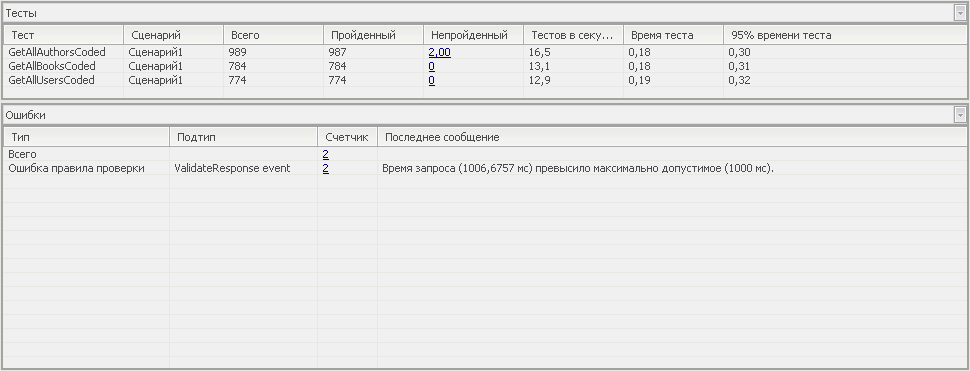
**SOAK/LOAD Testing и Stress Testing**

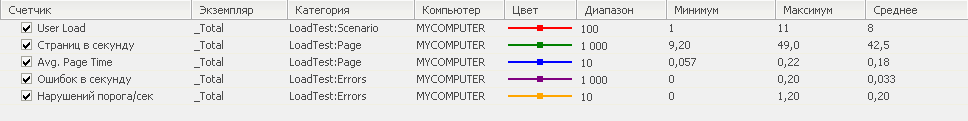
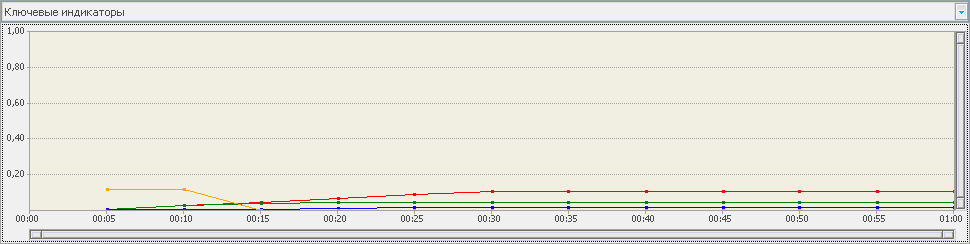
Не представляется необходимым для данной не слишком нагруженной системы, так как ее производительность уже по Capacity Testing намногопревышает требования к пиковой нагрузке.

Тем не менее можно было бы запустить данный нагрузочный тест в течение длительного времени для выявления возможных утечек памяти на сервере.









public class GetAllAuthorsCoded : WebTest

{

public GetAllAuthorsCoded()

{

this.PreAuthenticate = true;

}

public override IEnumerator<WebTestRequest> GetRequestEnumerator()

{

// Инициализация правил проверки, применяемых ко всем запросам в веб-тесте

if ((this.Context.ValidationLevel >= Microsoft.VisualStudio.TestTools.WebTesting.ValidationLevel.High))

{

ValidationRuleRequiredTag validationRule1 = new ValidationRuleRequiredTag();

validationRule1.RequiredTagName = "Author";

validationRule1.MinOccurrences = 1;

this.ValidateResponse += new EventHandler<ValidationEventArgs>(validationRule1.Validate);

}

if ((this.Context.ValidationLevel >= Microsoft.VisualStudio.TestTools.WebTesting.ValidationLevel.High))

{

ValidationRuleRequestTime validationRule2 = new ValidationRuleRequestTime();

validationRule2.MaxRequestTime = 1000;

this.ValidateResponse += new EventHandler<ValidationEventArgs>(validationRule2.Validate);

}

WebTestRequest request1 = new WebTestRequest("http://193.226.64.163/SANDBOXWebServiceDev/Service1.asmx");

request1.Timeout = 3;

request1.Method = "POST";

request1.Headers.Add(new WebTestRequestHeader("SOAPAction", "http://tempuri.org/GetAllAuthors"));

StringHttpBody request1Body = new StringHttpBody();

request1Body.ContentType = "text/xml";

request1Body.InsertByteOrderMark = false;

request1Body.BodyString = @"<?xml version=""1.0"" encoding=""utf-8""?>

<soap:Envelope xmlns:xsi=""http://www.w3.org/2001/XMLSchema-instance"" xmlns:xsd=""http://www.w3.org/2001/XMLSchema"" xmlns:soap=""http://schemas.xmlsoap.org/soap/envelope/"">

<soap:Body>

<GetAllAuthors xmlns=""http://tempuri.org/"" />

</soap:Body>

</soap:Envelope>";

request1.Body = request1Body;

yield return request1;

request1 = null;

}

}

public class GetAllBooksCoded : WebTest

{

public GetAllBooksCoded()

{

this.PreAuthenticate = true;

}

public override IEnumerator<WebTestRequest> GetRequestEnumerator()

{

// Инициализация правил проверки, применяемых ко всем запросам в веб-тесте

if ((this.Context.ValidationLevel >= Microsoft.VisualStudio.TestTools.WebTesting.ValidationLevel.High))

{

ValidationRuleRequestTime validationRule1 = new ValidationRuleRequestTime();

validationRule1.MaxRequestTime = 1000;

this.ValidateResponse += new EventHandler<ValidationEventArgs>(validationRule1.Validate);

}

if ((this.Context.ValidationLevel >= Microsoft.VisualStudio.TestTools.WebTesting.ValidationLevel.High))

{

ValidationRuleRequiredTag validationRule2 = new ValidationRuleRequiredTag();

validationRule2.RequiredTagName = "Book";

validationRule2.MinOccurrences = 1;

this.ValidateResponse += new EventHandler<ValidationEventArgs>(validationRule2.Validate);

}

WebTestRequest request1 = new WebTestRequest("http://193.226.64.163/SANDBOXWebServiceDev/Service1.asmx");

request1.Timeout = 3;

request1.Method = "POST";

request1.Encoding = System.Text.Encoding.GetEncoding("utf-8");

request1.Headers.Add(new WebTestRequestHeader("SOAPAction", "http://tempuri.org/FindBooks"));

StringHttpBody request1Body = new StringHttpBody();

request1Body.ContentType = "text/xml";

request1Body.InsertByteOrderMark = false;

request1Body.BodyString = @"<?xml version=""1.0"" encoding=""utf-8""?>

<soap:Envelope xmlns:xsi=""http://www.w3.org/2001/XMLSchema-instance"" xmlns:xsd=""http://www.w3.org/2001/XMLSchema"" xmlns:soap=""http://schemas.xmlsoap.org/soap/envelope/"">

<soap:Body>

<FindBooks xmlns=""http://tempuri.org/"" />

</soap:Body>

</soap:Envelope>";

request1.Body = request1Body;

yield return request1;

request1 = null;

}

}

public class GetAllUsersCoded : WebTest

{

public GetAllUsersCoded()

{

this.PreAuthenticate = true;

}

public override IEnumerator<WebTestRequest> GetRequestEnumerator()

{

// Инициализация правил проверки, применяемых ко всем запросам в веб-тесте

if ((this.Context.ValidationLevel >= Microsoft.VisualStudio.TestTools.WebTesting.ValidationLevel.High))

{

ValidationRuleRequiredTag validationRule1 = new ValidationRuleRequiredTag();

validationRule1.RequiredTagName = "User";

validationRule1.MinOccurrences = 1;

this.ValidateResponse += new EventHandler<ValidationEventArgs>(validationRule1.Validate);

}

if ((this.Context.ValidationLevel >= Microsoft.VisualStudio.TestTools.WebTesting.ValidationLevel.High))

{

ValidationRuleRequestTime validationRule2 = new ValidationRuleRequestTime();

validationRule2.MaxRequestTime = 1000;

this.ValidateResponse += new EventHandler<ValidationEventArgs>(validationRule2.Validate);

}

WebTestRequest request1 = new WebTestRequest("http://193.226.64.163/SANDBOXWebServiceDev/Service1.asmx");

request1.Timeout = 3;

request1.Method = "POST";

request1.Encoding = System.Text.Encoding.GetEncoding("utf-8");

request1.Headers.Add(new WebTestRequestHeader("SOAPAction", "http://tempuri.org/GetAllUsers"));

StringHttpBody request1Body = new StringHttpBody();

request1Body.ContentType = "text/xml";

request1Body.InsertByteOrderMark = false;

request1Body.BodyString = @"<?xml version=""1.0"" encoding=""utf-8""?>

<soap:Envelope xmlns:xsi=""http://www.w3.org/2001/XMLSchema-instance"" xmlns:xsd=""http://www.w3.org/2001/XMLSchema"" xmlns:soap=""http://schemas.xmlsoap.org/soap/envelope/"">

<soap:Body>

<GetAllUsers xmlns=""http://tempuri.org/"" />

</soap:Body>

</soap:Envelope>";

request1.Body = request1Body;

yield return request1;

request1 = null;

}

}