

Технология Windows Communication Foundation (WCF) и ее использование для создания распределенных приложений

Князьков К.В., Ковальчук С.В.

Санкт-Петербург — 2012

Терминология

- **Распределённые вычисления** (distributed computing) — способ решения трудоёмких вычислительных задач с использованием двух и более компьютеров.
- **Сёрвис-ориентированная архитектура** (англ. SOA, service-oriented architecture) — модульный подход к разработке программного обеспечения, основанный на использовании сервисов (служб) со стандартизированными интерфейсами.
- **URI** (англ. **Uniform Resource Identifier**) — унифицированный (единообразный) идентификатор ресурса.
- **Windows Communication Foundation (WCF)** — программный фреймворк, используемый для обмена данными между приложениями и входящий в состав .NET Framework.

© Википедия (<http://ru.wikipedia.org/>)

Веб-сервисы

Веб-сервис – программная система, созданная для поддержки интероперабельного межкомпьютерного взаимодействия через сеть.

W3C

Для чего нужны веб-сервисы:

- Стандартизованный способ взаимодействия между приложениями в Интернет
- Связь существующих неоднородных приложений (Java, .NET, Python...)
- Связь приложений на гетерогенных платформах (Windows, Unix...)
- Возможность переиспользования существующих компонентов
- Возможность интеграции ПО, разработанного разными командами

Ориентация на сервисы

- **Четкие границы**
 - Пересечение границ – четко определено
 - Пересечение границ стоит ресурсов
- **Сервисы самостоятельны**
 - Мы должны понимать что используемый нами сервис будет развиваться и у нас нет контроля над этим
 - Сервисы управляются и разрабатываются независимо
 - Сервис, который мы используем может быть недоступен!
- **Сервисы предоставляют схему и контракт, но не код**
 - Сервисы взаимодействуют по контрактам, которые не меняются.
 - Сервисы предоставляют только контракт, реализация может меняться.
- **Совместимость сервисов определяется политикой**
 - Безопасность, гарантированная доставка , и прочее определяется политикой.
 - Требования и возможности сервиса также предоставляются политикой.

Модели проектирования веб-сервисов

RPC

Remote Procedure Call

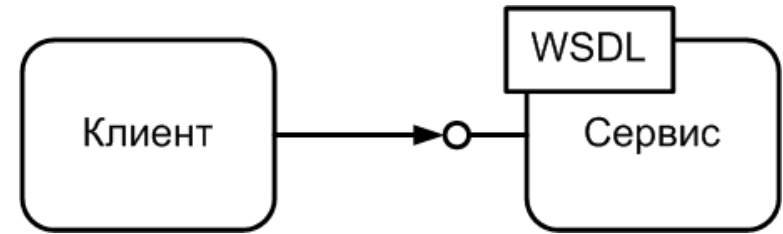
REST

Representational State Transfer

Клиентский контекст (состояние)	Хранится между запросами	Не хранится между запросами
Основной элемент	Метод	Ресурс
Протокол	SOAP	Использует возможности HTTP
Операции	Большое количество методов	Методы HTTP, реализующие CRUD
Адрес	Один или небольшое количество	Для каждого ресурса свой

Характеристики:

- Произошел от XML-RPC
- Формат сообщений: XML
- Расширяемость
- Поддержка любого транспортного протокола: HTTP, SMTP, TCP
- Независимость от модели программирования
- Самоописание за счет WSDL (Web Services Description Language)



*Запрос на исполнение операции **GetStockPrice***

POST /InStock HTTP/1.1

Host: www.example.org

Content-Type: application/soap+xml; charset=utf-8

Content-Length: 299

SOAPAction: "http://www.w3.org/2003/05/soap-envelope"

```
<?xml version="1.0"?>
```

```
<soap:Envelope
```

```
xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
```

```
<soap:Header>
```

```
</soap:Header>
```

```
<soap:Body>
```

```
<m:GetStockPrice
```

```
xmlns:m="http://www.example.org/stock">
```

```
<m:StockName>IBM</m:StockName>
```

```
</m:GetStockPrice>
```

```
</soap:Body>
```

```
</soap:Envelope>
```

WSDL

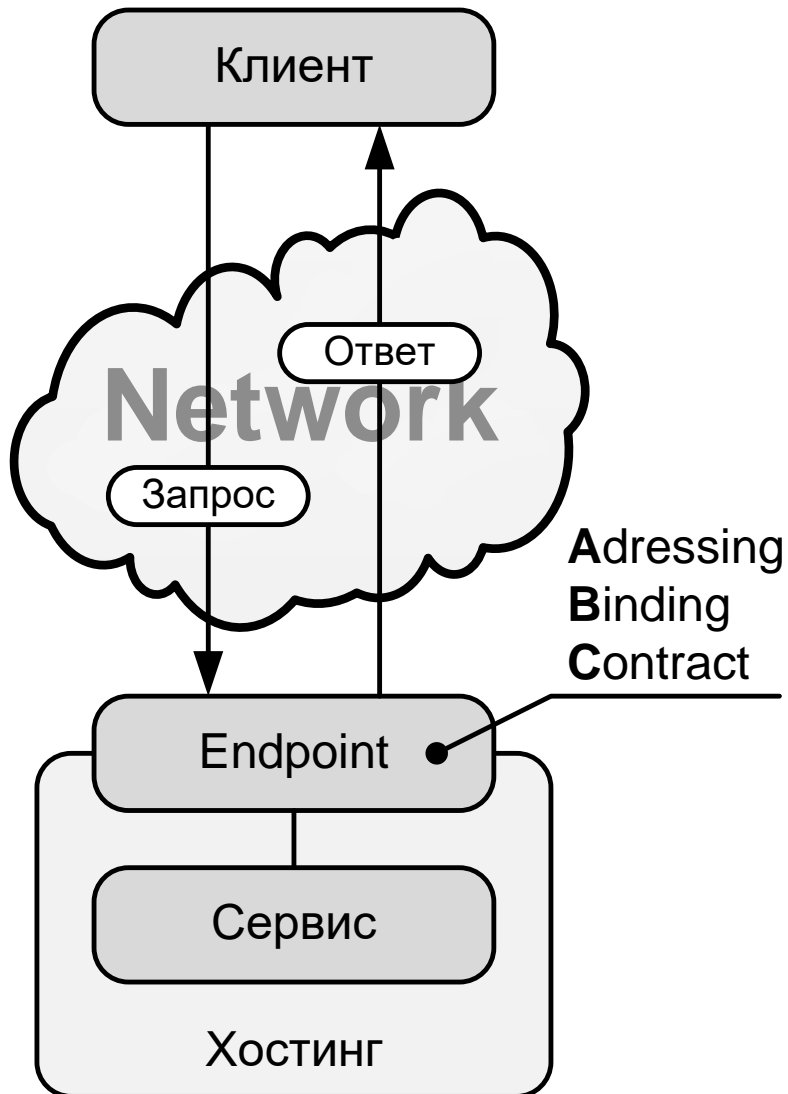
WSDL (англ. Web Services Description Language) – язык описания веб-сервисов и доступа к ним, основанный на языке XML.

В WSDL-описании сервис характеризуется:

1. Порт (Port, Endpoint) – адрес точки соединения (URL)
2. Привязка (Binding) – описывает, каким образом происходит взаимодействие
3. Интерфейс (Интерфейс)
4. Операции (Operations) – предоставляемые действия в виде функций
5. Типы данных (Types)

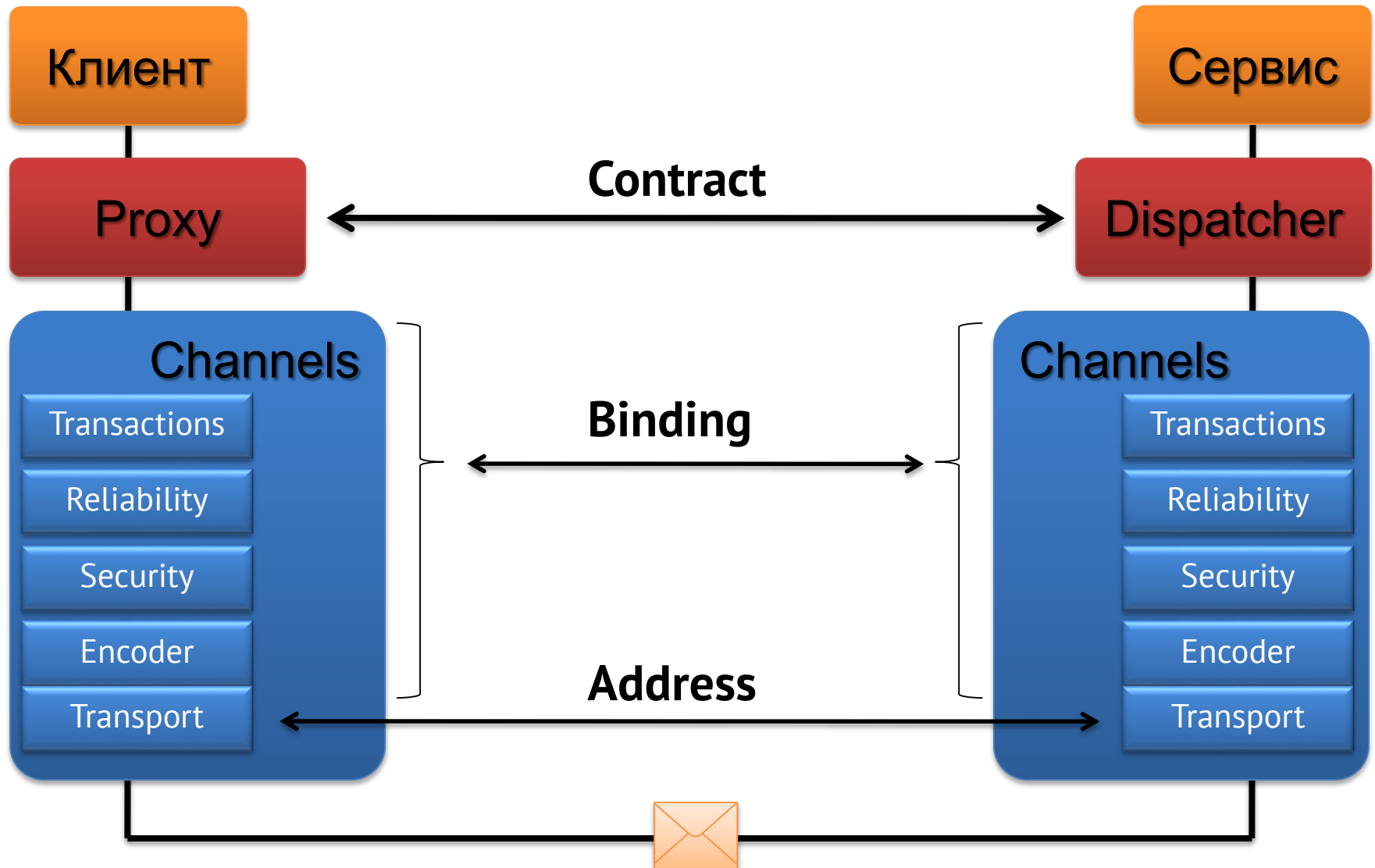
*Пример реального открытого сервиса –
проверка правописания от Яндекс:
<http://speller.yandex.net/services/spellservice>*

WCF



- **Addressing (где?)**
 - Описание расположения сервиса в сети
 - Пример:
`net.tcp://host.ru:88/Service`
- **Binding (как?)**
 - Описание протокола, кодирование данных и пр.
- **Contract (что?)**
 - Интерфейс сервиса
 - Реализуемые сервисом методы
 - Используемые составные структуры данных

WCF ABC



Адресация

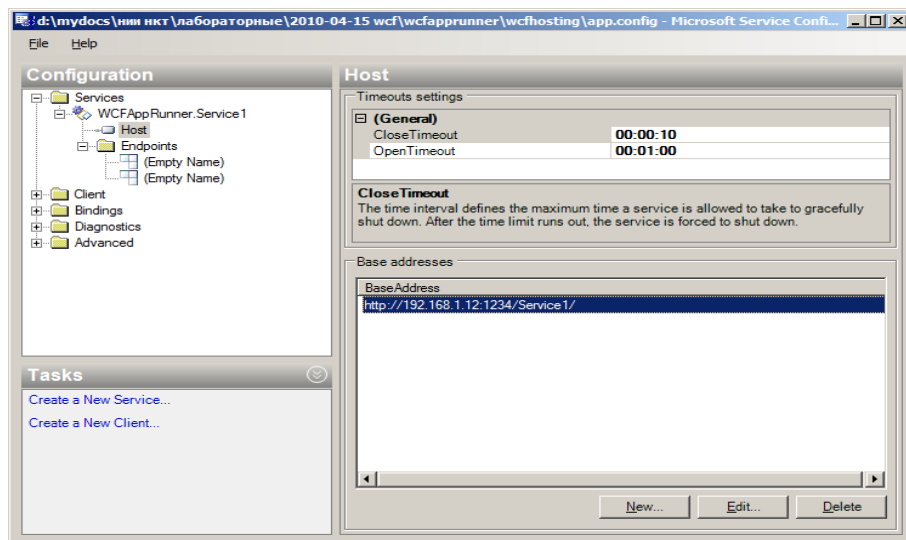
<схема> : // <сервер> : <порт> / <путь>

http
https
net.tcp
msmq

localhost
IP-адрес
DNS-имя

80
8080

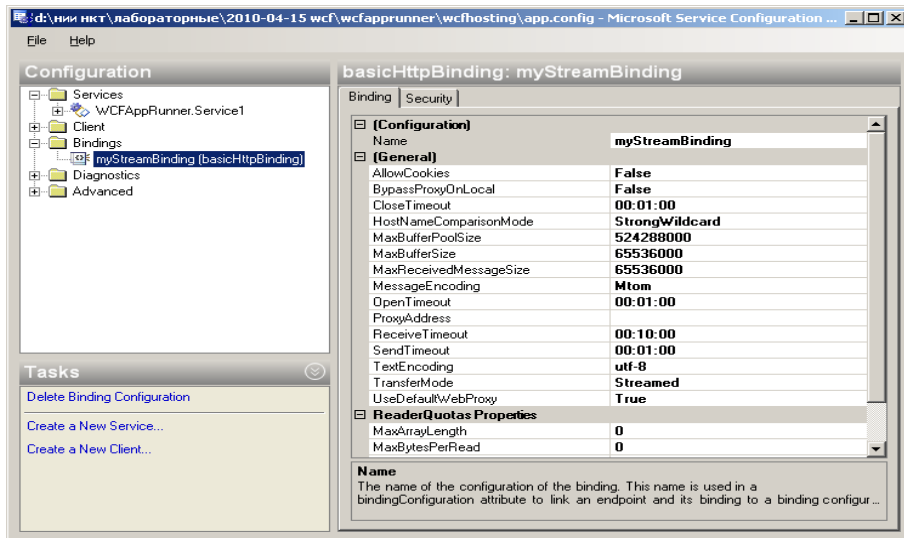
base
+
endpoint



```
<services>
  <service behaviorConfiguration="WCFAppRunner.Service1Behavior"
    name="WCFAppRunner.Service1">
    <clear />
    <endpoint address="mex" binding="mexHttpBinding"
      contract="IMetadataExchange" listenUriMode="Explicit" />
    <endpoint address="MyService" binding="basicHttpBinding"
      bindingConfiguration="myStreamBinding"
      contract="WCFAppRunner.IService1" />
    <host>
      <baseAddresses>
        <add baseAddress="http://192.168.1.12:1234/Service1/" />
      </baseAddresses>
    </host>
  </service>
</services>
```

XML-конфигурация

Привязки



- Базовые конфигурации:
BasicHttpBinding,
WSHttpBinding,
NetPeerTcpBinding,
MsmqIntegrationBinding и пр.

– Поддерживают стандартные протоколы (совместимы с сервисами на других платформах)

- Расширение конфигурации

Подключение дополнительных возможностей WCF: работа с потоками, бинарное кодирование данных

Настройка существующих параметров: максимального размера сообщения

```
<bindings>
<basicHttpBinding>
  <binding name="myStreamBinding"
    maxBufferSize="65536000"
    maxBufferPoolSize="524288000"
    maxReceivedMessageSize="65536000"
    messageEncoding="Mtom" transferMode="Streamed" />
</basicHttpBinding>
</bindings>
<services>
  <service behaviorConfiguration="WCFAppRunner.Service1Behavior"
    name="WCFAppRunner.Service1">
    <clear />
    <endpoint address="mex" binding="mexHttpBinding"
      contract="IMetadataExchange" listenUriMode="Explicit" />
    <endpoint address="qqq" binding="basicHttpBinding"
      bindingConfiguration="myStreamBinding" contract="WCFAppRunner.IService1"/>
    <host>
      <baseAddresses>
        <add baseAddress="http://192.168.1.12:1234/Service1/" />
      </baseAddresses>
    </host>
  </service>
</services>
```

Контракт

[ServiceContract]

```
public interface IService
```

```
{
```

[OperationContract]

```
    string GetData(int value);
```

[OperationContract]

```
    CompositeType GetData(CompositeType d);
```

```
}
```

Виды контрактов:

- ServiceContract
 - OperationContract
- DataContract
 - DataMember

[DataContract]

```
public class CompositeType
```

```
{
```

[DataMember]

```
    public List<int> ListValue {get;set;}
```

[DataMember]

```
    public string StringValue {get;set;}
```

```
}
```

*Для определения
контрактов
используются
атрибуты*

Преимущества WCF

1	Большие возможности по расширяемости	<i>Можем заменить или добавить свои элементы</i>
2	Независимость от используемых протоколов передачи: tcp, named pipe, msmq	<i>Один раз пишем код логики, используем (одновременно) разные транспорты.</i>
3	Динамическая конфигурация сервисов и клиентов	<i>Можно с помощью конфигурации менять порты, адреса, etc.</i>
4	Множество вариантов хостинга	<i>Можно обойтись без IIS: консольное приложение или windows-сервис (демон)</i>

Применение сервисно-ориентированной архитектуры для задач моделирования сложных систем

Характеристики сложных систем

Множество составных частей

Сложность связей

Изменчивость систем

Особенности моделирования

Использование готовых моделей

Необходимость синхронизации / сопряжения моделей

Динамическая структура комплексов

SOA

Изолированные и унифицированные сервисы

Стандартные протоколы работы с сервисами

Независимость сервисов, динамическая компоновка

Задача



- Задача

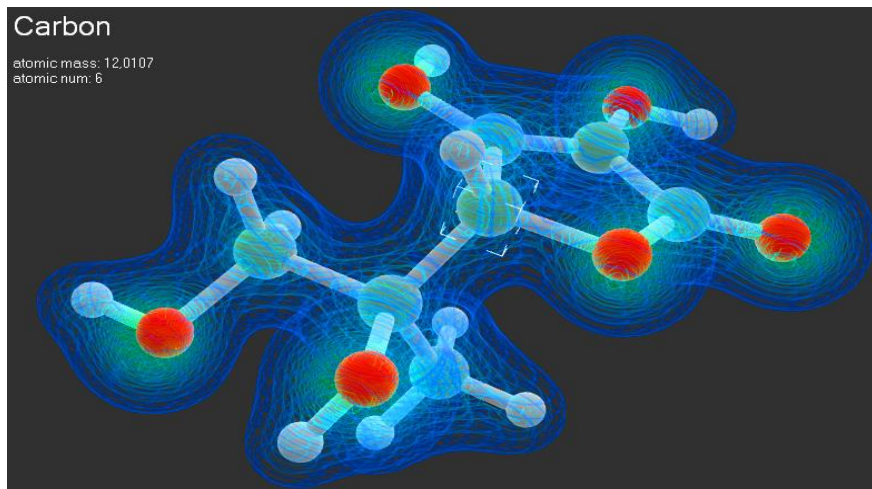
- Расчет электронной плотности с использованием пакета ORCA
- Визуализация полученных результатов

- Требования

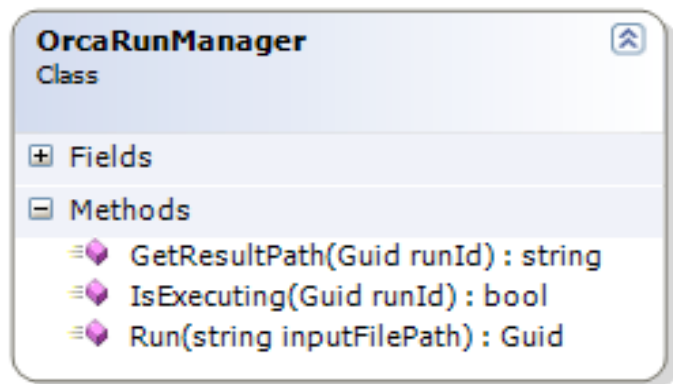
- Удаленный вызов сервисов вычисления и визуализации
- Оптимизация передаваемых данных

- Тестовые примеры

- \Samples*.inp



Шаг 1: подготовка



- Использование вспомогательной библиотеки:
 - OrcaRunner.dll
- Запуск пакета

`OrcaRunManager.Run(<путь к файлу>);`

- Результат – GUID для идентификации запуска (RunID)
- Проверка завершения работы

`OrcaRunManager.IsExecuting(<RunID>);`

- Результат – признак завершения (bool)
- Получение пути к файлу с результатом

`OrcaRunManager.GetResultPath(<RunID>);`

- Результат – путь к файлу с результатом, готовым для визуализации (eldens.cube)
- **Результат**
 - Корректно запускается и отрабатывает ORCA
 - В результате запуска образуется выходной файл

```
C:\Windows\System32\cmd.exe
Resolution ... 100 100 100
Boundaries ... -14.282088 11.824883 <x direction>
               -11.272162 15.136792 <y direction>
               -14.725199 7.376671 <z direction>

*****
* ORCA property calculations *
*****

Active property flags

(<+> Dipole Moment

-----
ORCA ELECTRIC PROPERTIES CALCULATION
-----

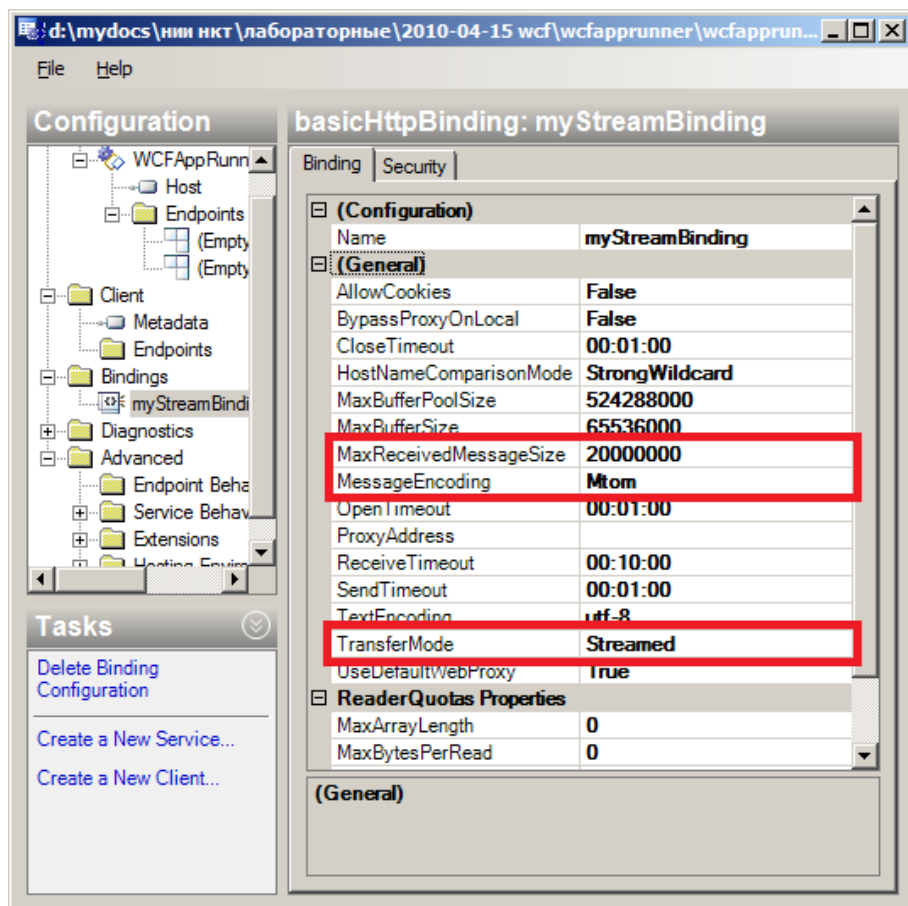
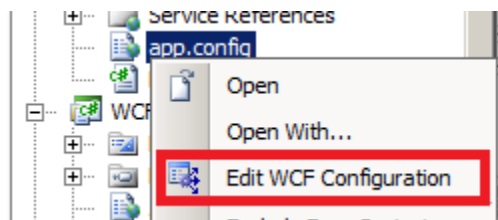
Dipole Moment Calculation ... on
Quadrupole Moment Calculation ... off
Polarizability Calculation ... off
GBUName ... Samples\UITAMINC.gbw
Electron density file ... Samples\UITAMINC.scfp.tmp

DIPOLE MOMENT

Electronic contribution: 4.11661 -5.81338 6.17804
Nuclear contribution : -2.95547 7.75502 -6.50873
Total Dipole Moment : 1.16113 1.94165 -0.33069
Magnitude <a.u.> : 2.28639
Magnitude <Debye> : 5.81154

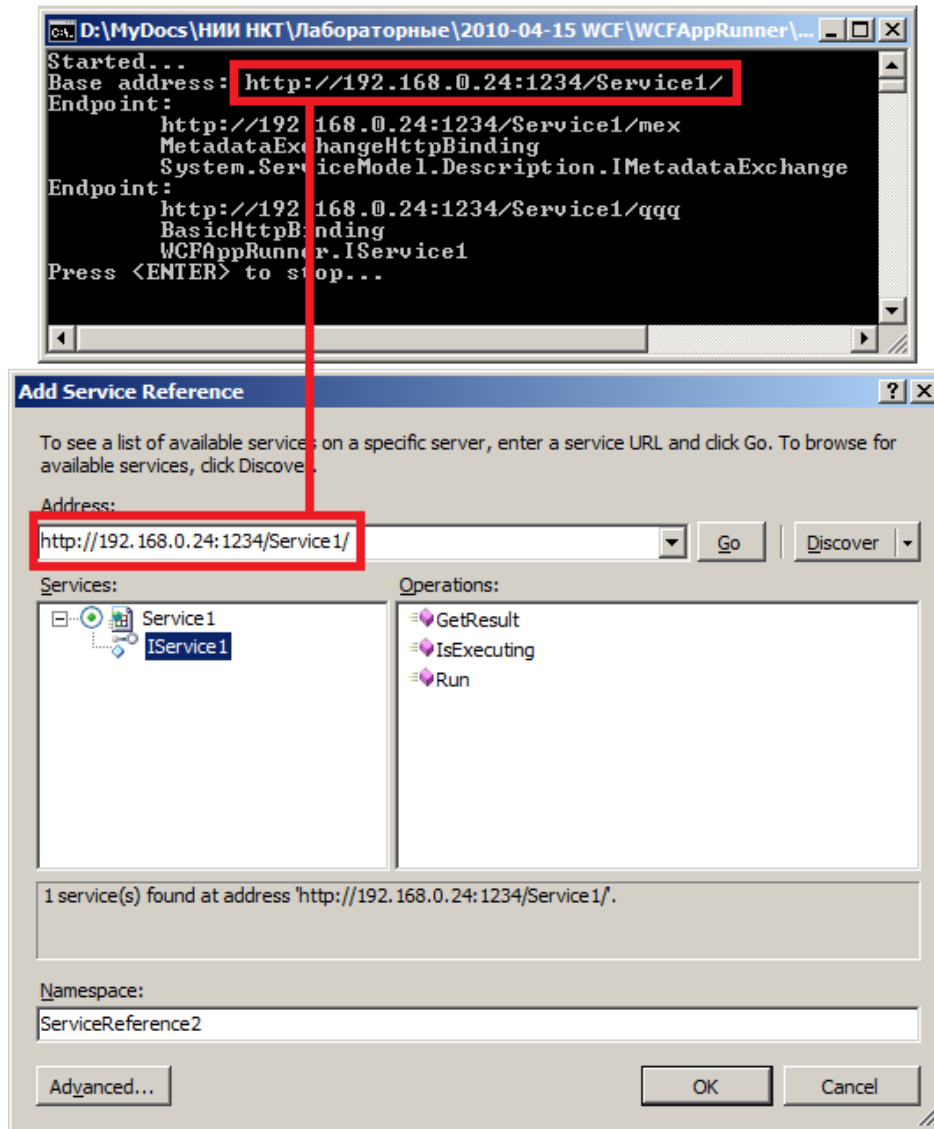
Timings for individual modules:
Sum of individual times ... 224.072 sec (= 3.735 min)
GTO integral calculation ... 0.562 sec (= 0.009 min) 0.3 %
SCF iterations ... 154.879 sec (= 2.581 min) 67.1 %
Orbital Density plot generation ... 68.631 sec (= 1.144 min) 30.6 %
*****ORCA TERMINATED NORMALLY*****
TOTAL RUN TIME: 0 days 0 hours 3 minutes 45 seconds 121 msec
C:\ORCA>
```


Шаг 2: создание WCF-сервиса



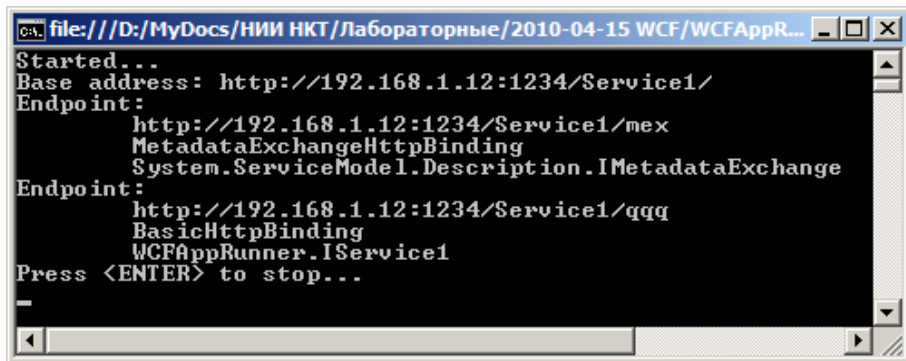
- Новый проект:
 - WCF Service Library
- Настройка сервиса
 - Конфигурация binding
 - TransferMode: Streamed
 - MaxReceivedMessageSize: 20 000 000
 - MessageEncoding: Mtom
- Работа с потоками:
 - Передача потоков в параметрах
 - StreamReader, StreamWriter
- Настройка клиента
 - Add Service Reference...
 - Проверка параметров, полученных от сервиса
- **Результат**
 - Запуск осуществляется через WCF-сервис
 - Работают потоки передачи файлов (передача входного файла и получение выходного)

Шаг 3: использование сервиса



- Перенастройка клиентского приложения
 - Новый ServiceReference
 - Проверка параметров, полученных от сервиса
- **Результат**
 - Использование WCF-сервиса

Шаг 4: хостинг WCF-сервиса

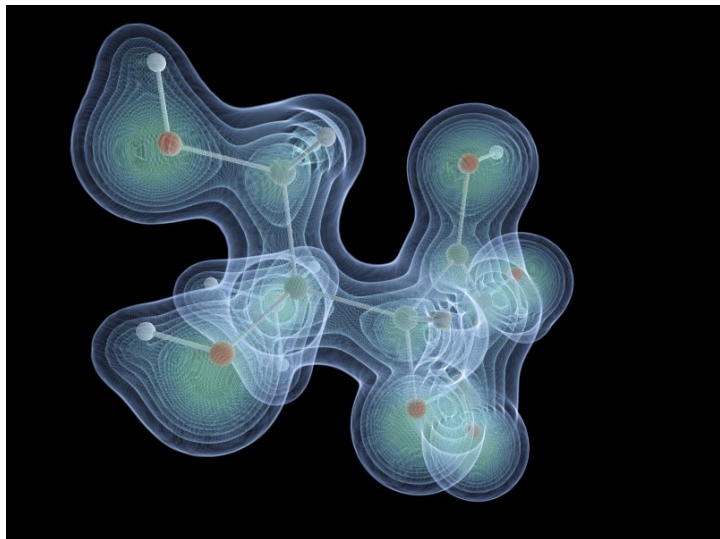


```
file:///D:/MyDocs/НИИ НКТ/Лабораторные/2010-04-15 WCF/WCFAppR...
Started...
Base address: http://192.168.1.12:1234/Service1/
Endpoint:
  http://192.168.1.12:1234/Service1/mex
  MetadataExchangeHttpBinding
  System.ServiceModel.Description.IMetadataExchange
Endpoint:
  http://192.168.1.12:1234/Service1/qqq
  BasicHttpBinding
  WCFAppRunner.IService1
Press <ENTER> to stop...
-
```

```
<service
  behaviorConfiguration=
    "WCFAppRunner.Service1Behavior"
  name="WCFAppRunner.Service1">
  <clear />
  <endpoint address="MyService"
    binding="basicHttpBinding"
    bindingConfiguration="myStreamBinding"
    contract="WCFAppRunner.IService1" />
</host>
  <baseAddresses>
    <add baseAddress=
      "http://192.168.1.12:1234/Service1/" />
  </baseAddresses>
</host>
</service>
```

- Консольное приложение, запускающее WCF-сервис
`System.ServiceModel.ServiceHost h = new ServiceHost(typeof(...))`
- Ссылка на библиотеку с описанием сервиса
- Конфигурационный файл
 - Application Configuration File
 - baseAddress с указанием IP компьютера
- **Результат**
 - Автономно работающий WCF-сервис

Шаг 5: удаленный WCF-сервис



```
yaw=0;  
roll=-50;  
pitch=0;  
distance=8;  
atom_scale=0.1;  
bond_radius=0.03;  
slice_num=500;  
intens_scale=100;
```

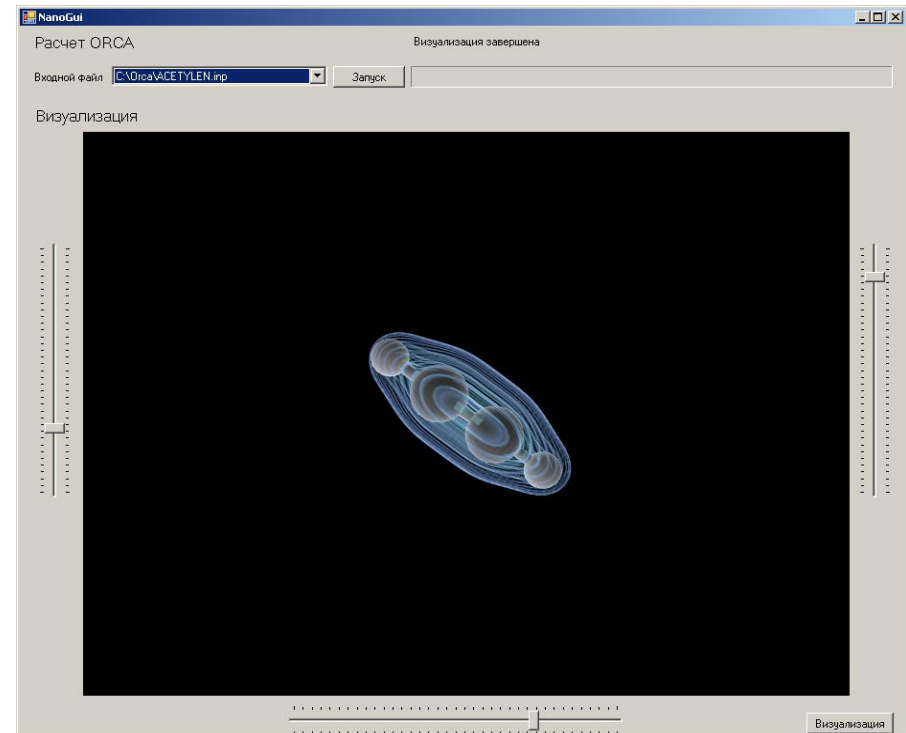
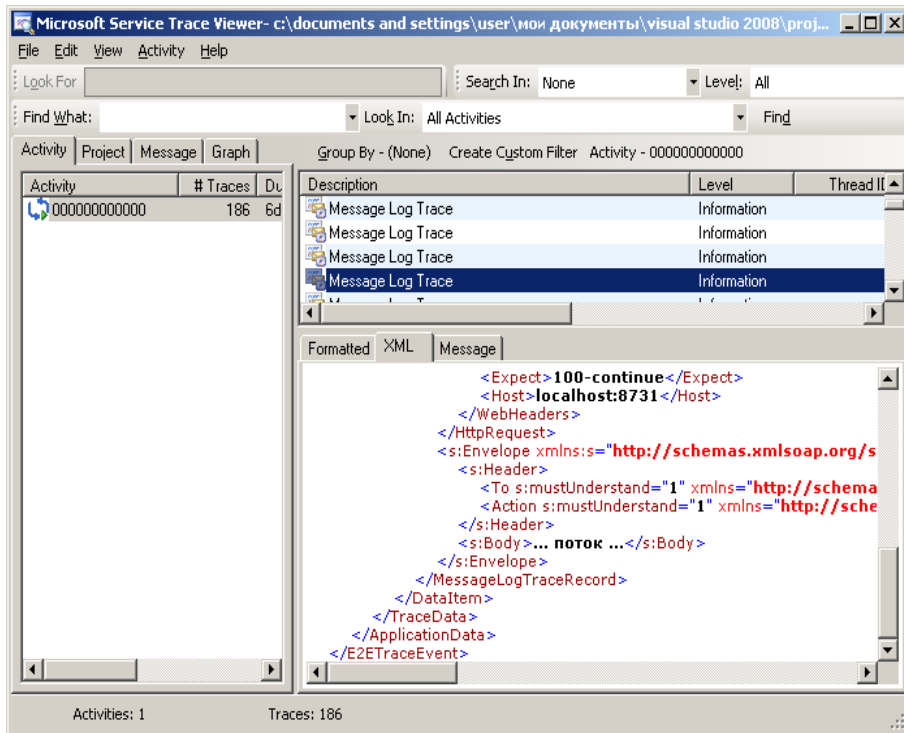
- Удаленный WCF-сервис визуализации
<http://<IP>:8731/VisualizerService/>
- Интерфейс:
`Guid uploadCubeFile(Stream inpFile);`
 - Загрузка файла (*.cube)`Stream visualizeCubeFile(
 Guid fid,
 Dictionary<string, string> param)`
 - Визуализация
- **Результат**
 - Запуск удаленного сервиса визуализации результатов вычислений

Дополнительно

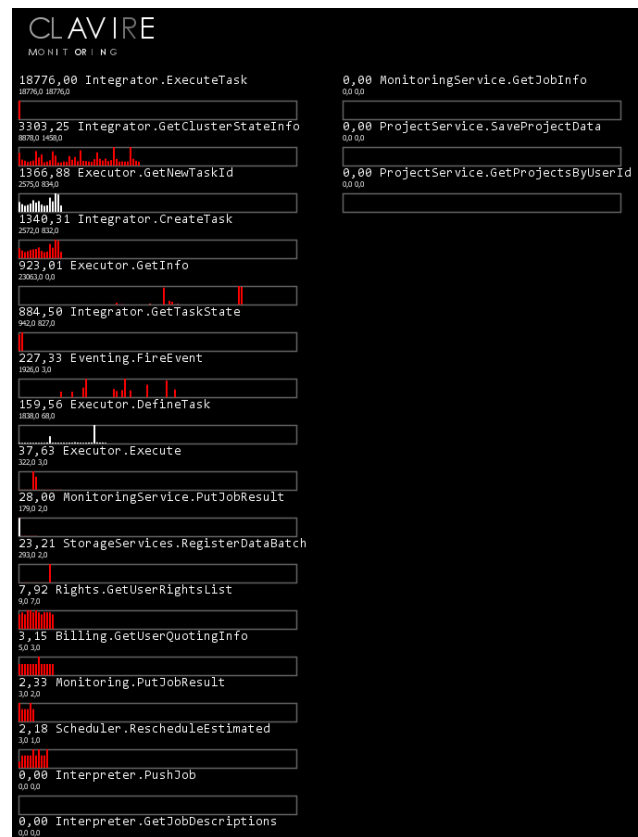
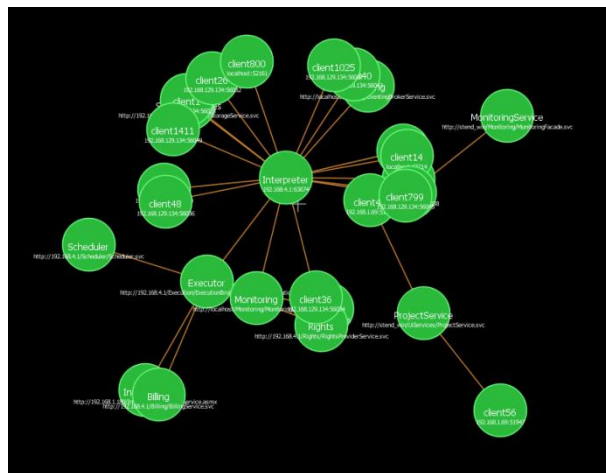
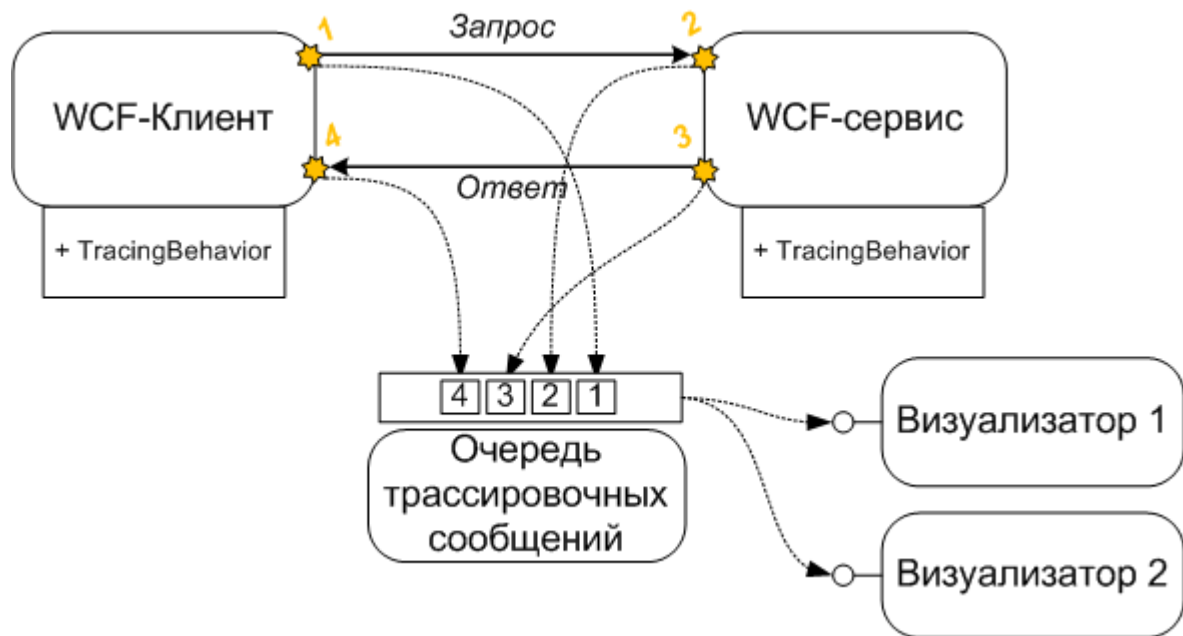
- Что можно сделать еще:
 - Отладка, трассировка
 - SvcTraceViewer.exe, SvcConfigEditor.exe, SvcUtil.exe ...
 - C:\Program Files\Microsoft SDKs\Windows\v6.0A\bin\
 - WcfSvcHost.exe, WcfTestClient.exe ...
 - C:\Program Files\Microsoft Visual Studio 9.0\Common7\IDE\
 - Безопасность
 - Пароль/логин, сертификат, ЭЦП
 - Асинхронный вызов
 - Методы Begin..., End...

Самостоятельная работа

- Трассировка
- Удаленный сервис вычислений (на соседнем ПК)
- GUI-клиент



Демонстрация профилировщика



Источники информации

- MSDN Library (<http://msdn.microsoft.com/en-us/library/>)
- MSDN WCF Portal (<http://msdn.microsoft.com/en-us/netframework/aa663324.aspx>)
- TechDays.ru (<http://www.techdays.ru/>)
- MSDN Cannel 9 (<http://channel9.msdn.com/>)
- <http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/>