

Universal Computational Cluster

Overall system activities, communication protocol and plugin design

Universal Computational Cluster (UCC) is meant for computing NP-hard problems by distributive exact algorithms. This document presents overall system activities on UML Activity diagrams, specifies the communication protocol by showing sequence diagrams and giving the XML Schema files for all the types of messages used by the system and as an appendix gives the crucial abstract class libraries for the Task Solvers.

1. System activities

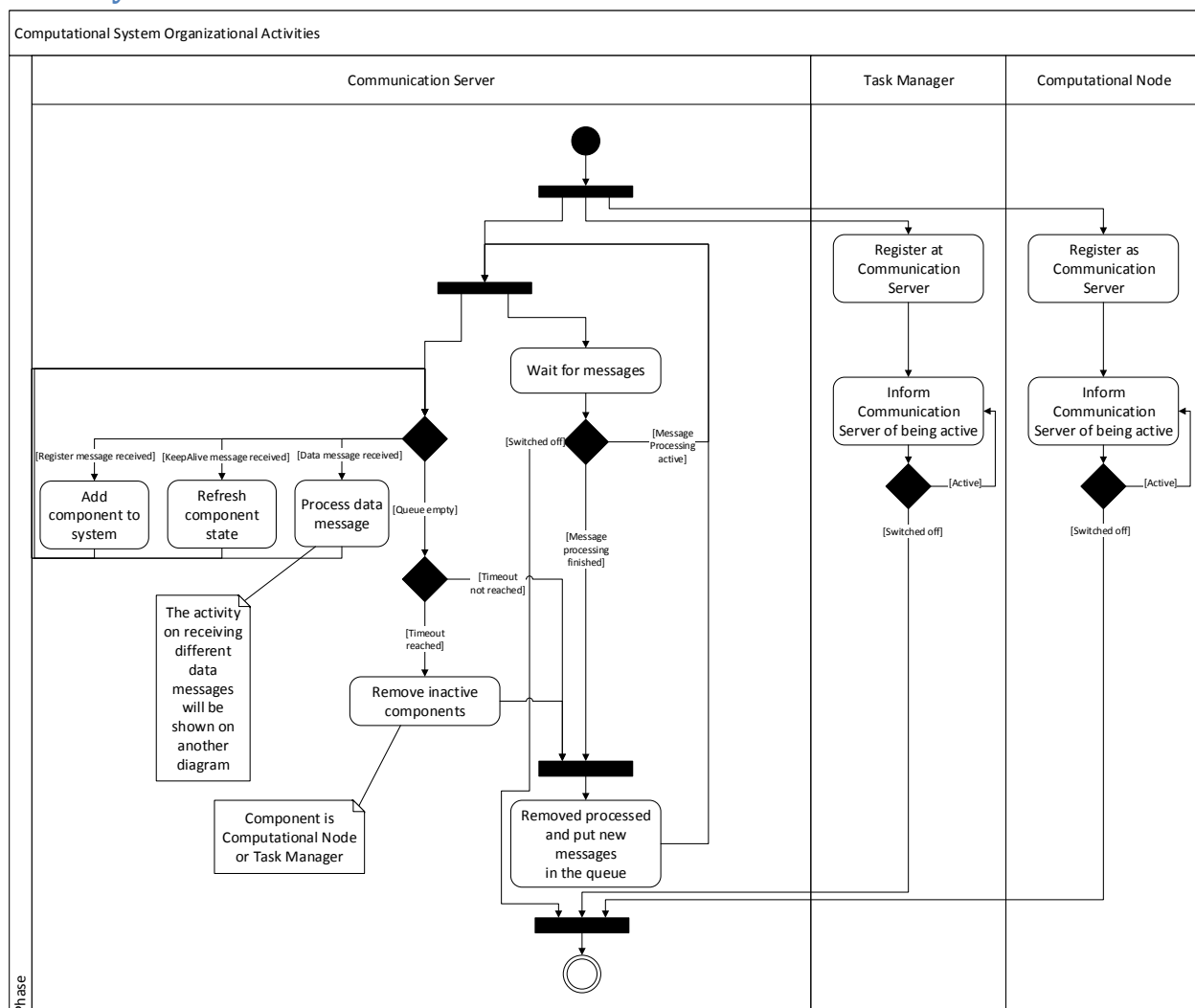


Figure 1 Overall system activities concerning the setup of the cluster

Fig. 1 presents how the components connect with each other and what activities are maintained even in idle state of the cluster. Task Manager (TM) and Computational Node (CN) register to the Communication Server (CS) and constantly inform CS of their state (thus maintaining the TCP/IP connection open).

Fig. 2 shows all the additional activities of the system after Computational Client (CC) connects to it and requests computations for a given problem to be performed. Both figures represent only one instance of each of the CN, TM and CC but possibly there could (and should) be many of them.

The important thing is that CS is always awaiting new messages (which are read almost instantly), regardless of the state of various problems send for computation. The processing of each of the message could be done separately and in parallel by different threads and the current state of problems is stored in the memory and not as a state of the system.

2. Communication protocol design

The system uses the TCP/IP protocol in order to send messages. Only the CS has a listens on a given port and all the other components start with connecting to CS. All the messages are a text XML messages.

Fig. 3 presents a scenario with two TM, two CN and one CC. The scenario shows that each of the TM could be chosen for dividing the problem/merging the solutions and presents what happened when one of the CN crashes.


```

        <xs:restriction base="xs:string">
            <xs:enumeration value="TaskManager" />
            <xs:enumeration value="ComputationalNode" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="SolvableProblems">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" name="ProblemName" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ParallelThreads" type="xs:unsignedByte" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

Register Response message

Register Response message is sent as a response to the Register message giving back the component its unique ID and informing how often it should send the Status message.

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace="http://www.mini.pw.edu.pl/ucc/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="RegisterResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Id" type="xs:unsignedLong" />
                <xs:element name="Timeout" type="xs:time" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>

```

Status message

Status message is sent by TM and CN to CS at least as frequent as a timeout given in Register Response. In the Status message the component reports the state of its threads, what they are computing (unique problem instance id, task id within the given problem instance, the type of problem instance) and for how long.

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace="http://www.mini.pw.edu.pl/ucc/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="Status">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Id" type="xs:unsignedLong" />
                <xs:element name="Threads">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element maxOccurs="unbounded" name="Thread">

```

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="State">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="Idle" />
          <xs:enumeration value="Busy" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element minOccurs="0" name="HowLong">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:pattern value="([0-9]+[.])?[0-9]{2}:[0-9]{2}:[0-9]{2}" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element minOccurs="0" name="ProblemInstanceId"
type="xs:unsignedLong" />
    <xs:element minOccurs="0" name="TaskId" type="xs:unsignedLong" />
    <xs:element minOccurs="0" name="ProblemType" type="xs:string" />
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

Solve Request message

Solve Request message is sent by the CC to CS. It gives the type of the problem instance to be solved, optionally the max time that could be used for computations and the problem data in base64.

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace="http://www.mini.pw.edu.pl/ucc/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="SolveRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ProblemType" type="xs:string" />
        <xs:element minOccurs="0" name="SolvingTimeout">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:pattern value="([0-9]+[.])?[0-9]{2}:[0-9]{2}:[0-9]{2}" />
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="Data" type="xs:base64Binary" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

Solve Request Response message

Solve Request Response message is sent by CS to CC as an answer for the Solve Request. It provides CC with unique identifier of the problem instance.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace=http://www.mini.pw.edu.pl/ucc/
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="SolveRequestResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Id" type="xs:unsignedLong" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Divide Problem message

Divide Problem is sent to TM to start the action of dividing the problem instance to smaller tasks. TM is provided with information about the computational power of the cluster in terms of total number of available threads.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace=http://www.mini.pw.edu.pl/ucc/
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="DivideProblem">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ProblemType" type="xs:string" />
        <xs:element name="Id" type="xs:unsignedLong" />
        <xs:element name="Data" type="xs:base64Binary" />
        <xs:element name="ComputationalNodes" type="xs:unsignedLong" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Partial Problems message

Partial problems message is sent by the TM after dividing the problem into smaller partial problems. The data in it consists of two parts – common for all partial problems and specific for the given task. The same Partial Problems schema is used for the messages sent to be computed by the CN.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace=http://www.mini.pw.edu.pl/ucc/
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="PartialProblems">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ProblemType" type="xs:string" />
        <xs:element name="Id" type="xs:unsignedLong" />
        <xs:element name="CommonData" type="xs:base64Binary" />
        <xs:element minOccurs="0" name="SolvingTimeout"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

<xs:simpleType>
  <xs:restriction base="xs:string">
    <xs:pattern value="([0-9]+[.])?[0-9]{2}:[0-9]{2}:[0-9]{2}" />
  </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="PartialProblems">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" name="PartialProblem">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="TaskId" type="xs:unsignedLong" />
            <xs:element name="Data" type="xs:base64Binary" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

Solutions message

Solutions message is used for sending info about ongoing computations, partial and final solutions from CN, to TM and to CC. In addition to sending task and solution data it also gives information about the time it took to compute the solution and whether computations were stopped due to timeout.

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace="http://www.mini.pw.edu.pl/ucc/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Solutions">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ProblemType" type="xs:string" />
        <xs:element name="Id" type="xs:unsignedLong" />
        <xs:element minOccurs="0" name="CommonData" type="xs:base64Binary" />
        <xs:element name="Solutions">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" name="Solution">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="TaskId" type="xs:unsignedLong" />
                    <xs:element name="TimeoutOccured" type="xs:boolean" />
                    <xs:element name="Type">
                      <xs:simpleType>
                        <xs:restriction base="xs:string">
                          <xs:enumeration value="Ongoing" />
                          <xs:enumeration value="Partial" />
                          <xs:enumeration value="Final" />
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>

```



```

        <xs:element name="ComputationsTime">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:pattern value="([0-9]+[.])?[0-9]{2}:[0-9]{2}:[0-9]{2}" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="Data" minOccurs="0" type="xs:base64Binary" />
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

Solution Request message

Solution Request message is sent from the CC in order to check whether the cluster has successfully computed the solution. It allows CC to be shut down and disconnected from server during computations.

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace="http://www.mini.pw.edu.pl/ucc/"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="SolutionRequest">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Id" type="xs:unsignedLong" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>

```

3. Configuration

In order to work the following parameters should be configured:

- Communications Server: listening port number
- Communications Server: communications timeout (status refresh interval)
- Other components: address and port of the CS