Russian Academy of Sciences

Program Systems Institute

MC#: a Language for Concurrent Disributed Programming for Cluster and GRID Architectures

Introduction

MC# is a high-level object-oriented programming language based on the .NET platform created for developing complex industrial software systems intended for running on multiprocessor architectures.

In contrast with the wide-spread MPI (Message Passing Interface) approach in MC# programs there is no need to

(a) explicitly manage the distribution of computational processes over cluster nodes or GRID machines, and

(b) manually program the serialization of complex objects (data) for sending them to another processors.

The MC# language is the adaptation of the base idea of the Polyphonic C# language (nowadays also known as Cω — Benton N., Cardelli L., Fournet C., Microsoft Research Laboratory, Cambridge, UK) for the case of concurrent distributed computations.

Asynchronous methods of Polyphonic C# in the MC# language can be scheduled for execution on other machines in two ways:

- automatically (typically a machine with the least workload is taken);
- manually through explicit indication by a programmer.

Such methods in the MC# language are called as **movable methods**. Interaction of movable methods running on different machines is realized through channels and channel message handlers. Synchronization of channels and handlers is implemented by the **chords** in the Polyphonic C# style.

Implementation

The MC# Programming System includes

- Compiler and
- Runtime system.

They are intended for running on the Mono platform [www.mono-project.com] — a free implementation of the .Net framework for Unix-like systems.

Novel Constructs of MC#: Movable Methods, Channels, and Handlers

Movable methods are the methods that can be scheduled for execution on remote machines. Such methods in the code are labeled with the special keyword **movable**; for example:

```
movable Compute( int x, double y ){
  // method body
}
```

There are two forms of calling movable methods:

- object.method(arguments) — in this case the node where movable method will be executed is scheduled by the runtime system,
- machine@object.method(arguments) — explicit indication of node where this method must be executed.

Channels and handlers are tools for interaction among distributed objects. Typically, they are defined with the chord construct as in the following example where channel sendInt for sending single integers and corresponding handler are declared:

```
chandler getInt int() & channel sendInt( int x ){
  return x;
}
```

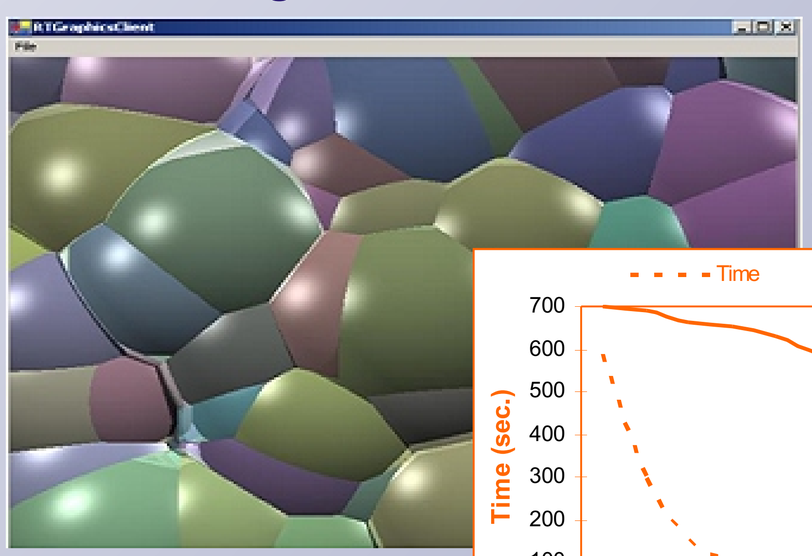
The body of the chord is executed only if all methods in its header were called. Such mechanism makes it possible to implement the synchronization of messages coming from the different channels:

```
chandler equals bool() & channel c1( int x ) & channel
c2( int y ) {
  if ( x == y )  return true;
  else  return false;
}
```

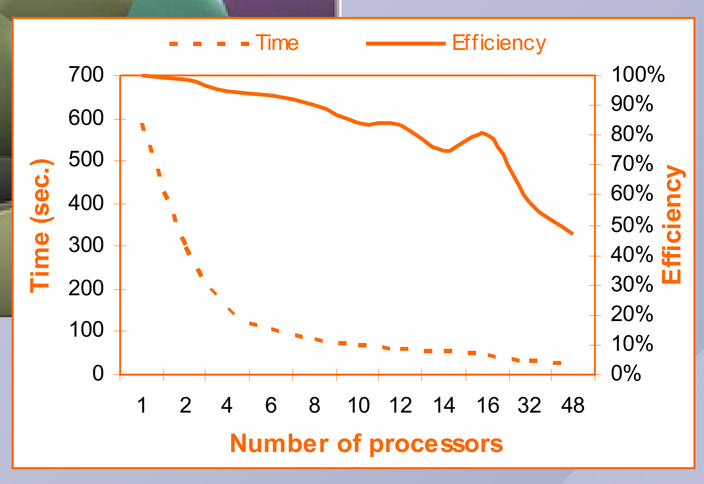
One of the key features of the MC# language is that channels and handlers can be passed to movable methods as arguments separately from the objects they belong to. In this sense they are similar to **delegate** construct of the C# language.

Applications

3D Rendering

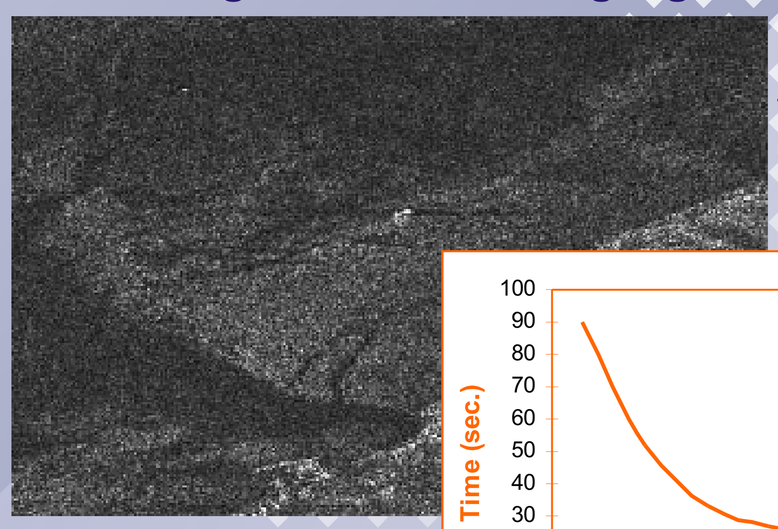


Resulting image for scene with 6000 spheres and 2 light sources.

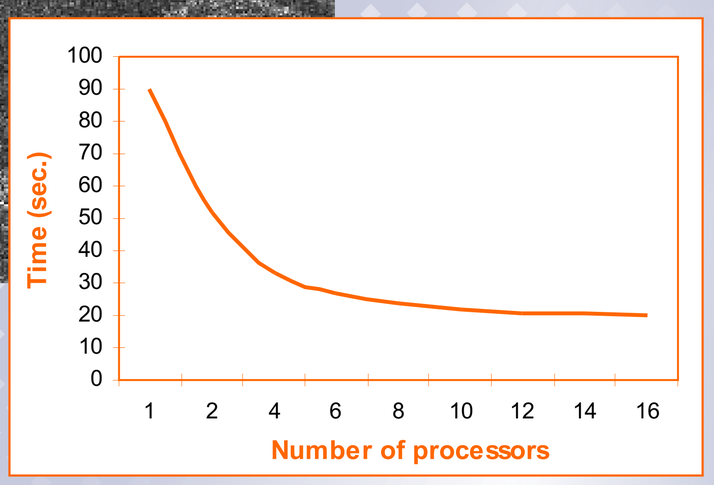


Computations were made on the K-500 cluster (64 nodes with dual Intel Xeon 2.8 GHz processor), Informatics Problems United Institute, National Academy of Sciences, Belarus.

Processing of Radar-tracking Signals



Result of processing of the radiohologram received from a satellite.



Computations were made on the SKIF cluster (16 nodes with the dual AMD Athlon MP 1800+ processor), Program Systems Institute, Russian Academy of Sciences, Pereslavl-Zalessky.

ADDRESS

Artificial Intelligence
Research Center
Program Systems Institute
Russian Academy of Sciences

Pereslavl-Zalessky
Yaroslavl Region
Russia, 152020
Tel/Fax: **+7 (08535) 98064**
E-mail: yury@serdyuk.botik.ru
Web-site: http://u.pereslavl.ru/~vadim/MCSharp/