Program Systems Institute
Russian Academy of Sciences

The Open TS dynamic parallelization system approach

Sergei M. Abramov, Alexander Moskovsky,
Vladimir Roganov, Elena Shevchuk
Program Systems Institute RAS,
Russia, Pereslavl-Zalessky

Presentation Outline

- Short self-introduction
- Open TS – T-system with the open architecture
  - Approach in general
  - Benefits
- Applications based on Open TS
- Future work

1. Short Self-Introduction

PSI RAS, Pereslavl-Zalesski

SKIF Supercomputing Project

- Joint of Russian Federation and Republic of Belarus
- 2000-2004
- 10 + 10 organizations
- PSI RAS is lead organization from Russian Federation
- Hardware and Software

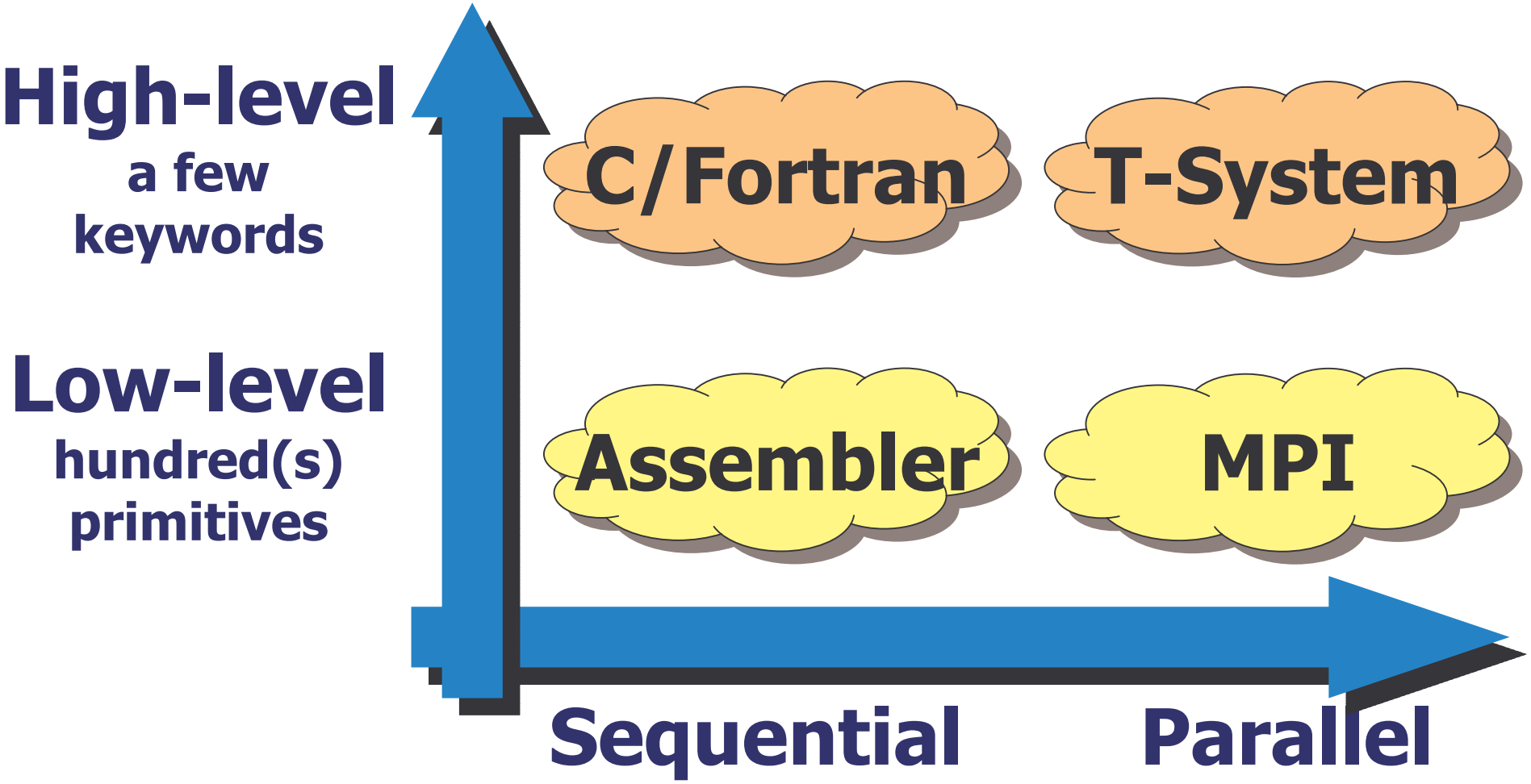
Flagship "SKIF K-1000"

- **Peak performance**
  2,5 Tflops
- **Linpack-performance**
  2,0 Tflops
- **Efficiency ratio**
  80.1 %

- November 2004 : **The most powerful** supercomputer in ex-USSR, **Rank 98** in Top500

Open TS Overview

Comparison: T-System and MPI



T-System in Comparison(1)

Related work	Open TS differentiator
Charm++	FP-inspired approach
UPC, CxC	Implicit parallelism
Glasgow Parallel Haskell	Allows C/C++ based low-level optimization
OMPC++	Provides both language and C++ templates library
Cilk	Supports SMP, MPI, PVM, and GRID platforms

T-System in Comparison(2)

Related work	Open TS differentiator
uSystem	Implicit parallelism
TStreams (HP)	Variables, C-based programming model
Mentat/Legion	Multiple assignment variables
Open MP	Function-level parallelism, not loop-based

T-System History

- **Mid-80-ies**
  Basic ideas of T-System
- **1990-ies**
  First implementation of T-System
- **2001-2002, "SKIF"**
  GRACE — Graph Reduction Applied to Cluster Environment
- **2003-current, "SKIF"**
  Open TS — Open T-system

Open TS/T-system

- Parallel programming tool
- Combination of ideas
  - High-level programming tool
  - Functional/imperative programming
  - Dynamic parallelization
  - → Coarse-grain dataflow (but not 100%)
- C++ (or another) language extension
- Multicore, SMPs, clusters, grids etc..

Open TS: an Outline

- High-performance computing
- "Automatic dynamic parallelization"
- Combining functional and imperative approaches, high-level parallel programming
- T++ language: "Parallel dialect" of C++ — an approach popular in 90-ies

  May express it in many ways (e.g. C++ templates)

T-Approach

- "Pure" function (**tfunction**) invocations (may) produce grains of parallelism
- T-Program is
  - Functional – on higher level
  - Imperative – on low level (optimization)
- C-compatible execution model
- Non-ready variables, <u>Multiple assignment</u>
- "Seamless" C-extension (or Fortran-extension)

T++ Keywords

- **tfun** — T-function
- **tval** — T-variable
- **tptr** — T-pointer
- **tout** — Output parameter (like &)
- **tdrop** — Make ready
- **twait** — Wait for readiness
- **tct** — T-context

Sample Program

```
#include <stdio.h>

tfun int fib (int n) {
  return n < 2  ?  n : fib(n-1)+fib(n-2);
}

tfun int main (int argc, char **argv) {
  if (argc != 2) { printf("Usage: fib <n>\n"); return 1; }
  int n = atoi(argv[1]);
  printf("fib(%d) = %d\n", n, (int)fib(n));
  return 0;
}
```

Open TS Approach benefits

- "Seamless" extention to sequential program
- Portability across multiple parallel architectures (multicore, SMP, clusters, heterogeneous clusters etc)
- Flexible load balancing enabled (adapt to dynamically changing environment, like cycle-sharing network)
- Fault-tolerance enabled
- Web-service synthesis

Web Services Generation

```
tfun int fib (int n) {
  return n < 2  ?  n : fib(n-1)+fib(n-2);
}
```

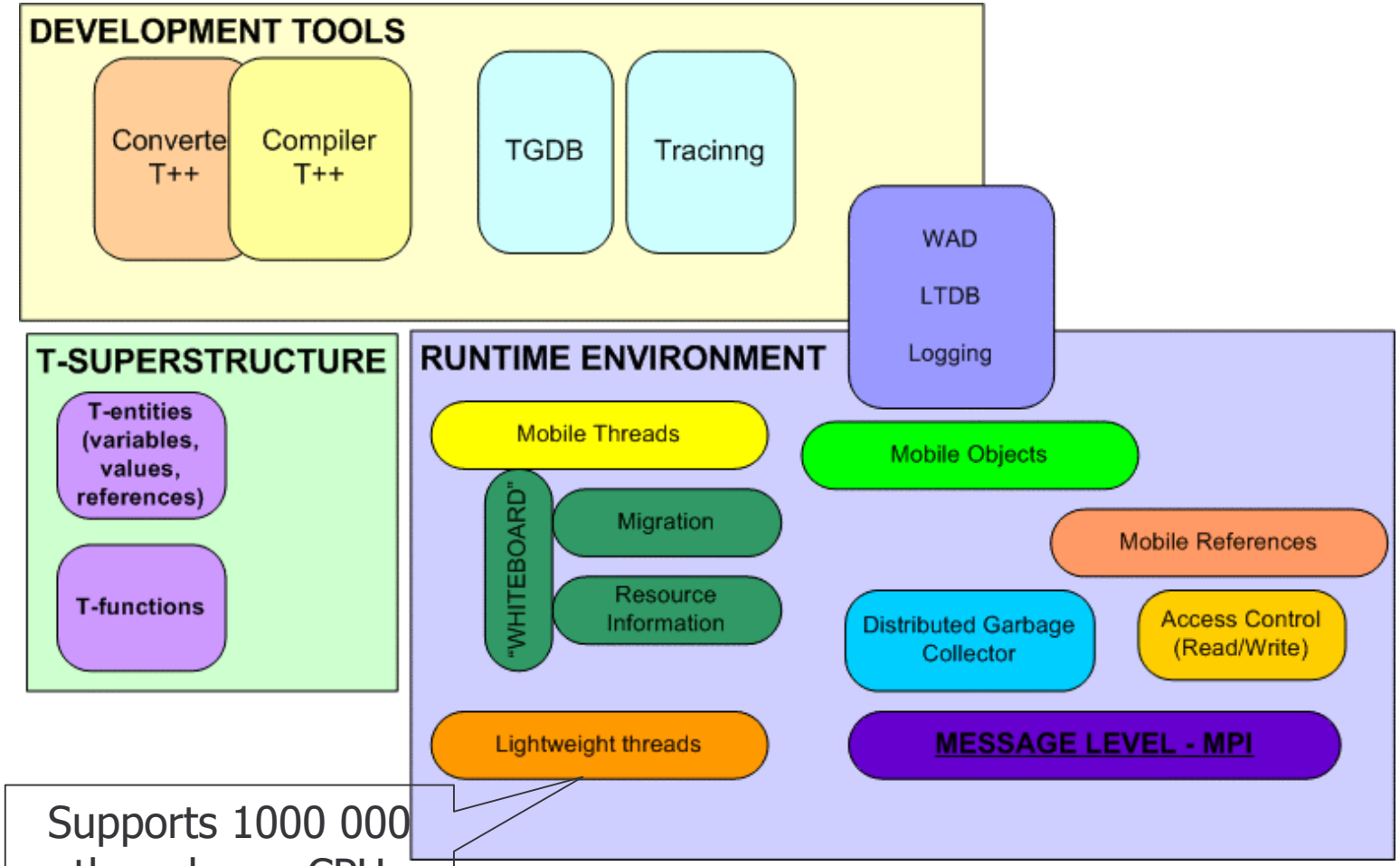
twsgen Perl script

▼

```
<operation name="wstfib">
  <SOAP:operation style="rpc" soapAction=""/>
  <input>
   <SOAP:body use="encoded" namespace="urn:myservice"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
  </input>
  <output>
   <SOAP:body use="encoded" namespace="urn:myservice"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
  </output>
 </operation>
```

gSOAP based server

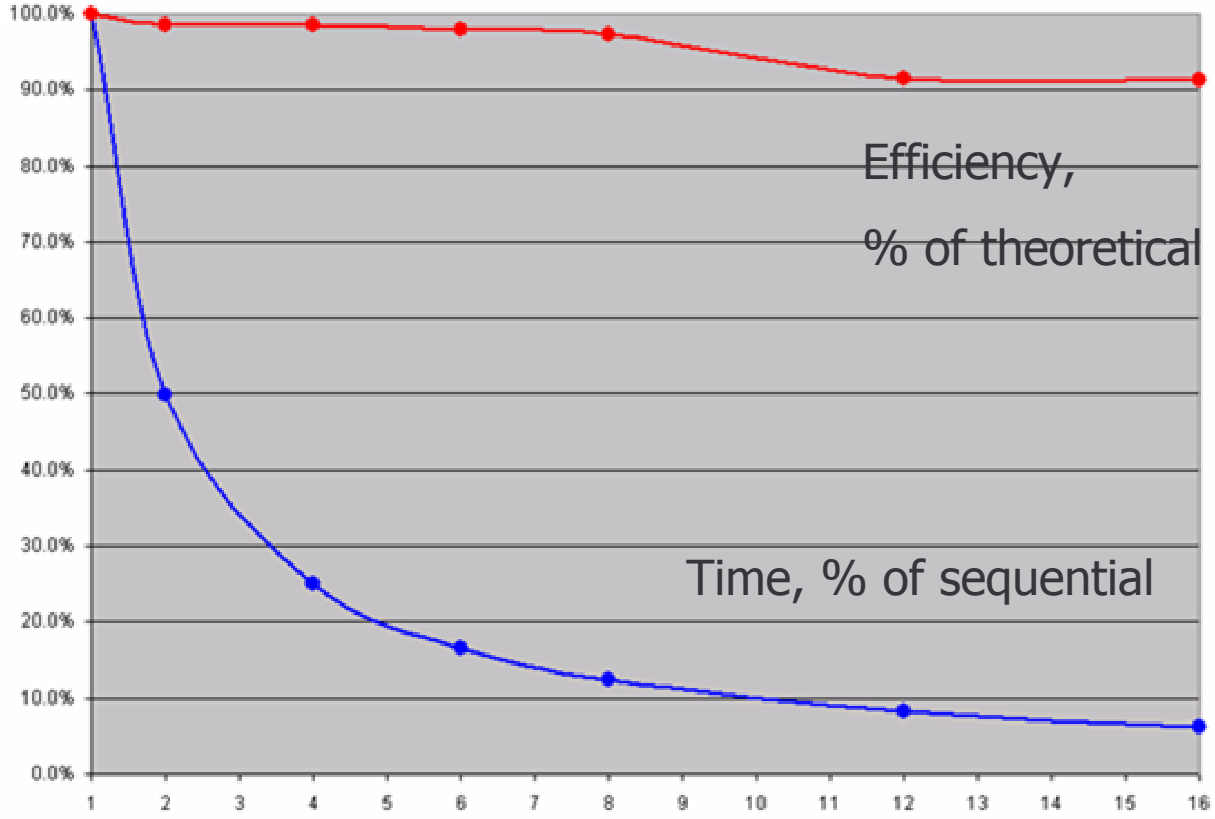
Open TS: Environment



Supports 1000 000 threads per CPU

NPB, Test EP Rewritten @OpenTS

- EP – Embarrassingly Parallel
- NASA Parallel Benchmarks suite
- Speedup = 96% of theoretical maximum (on 10 nodes)



Open TS vs MPI case study Applications

- Popular and widely used
- Developed by independent teams (MPI experts)

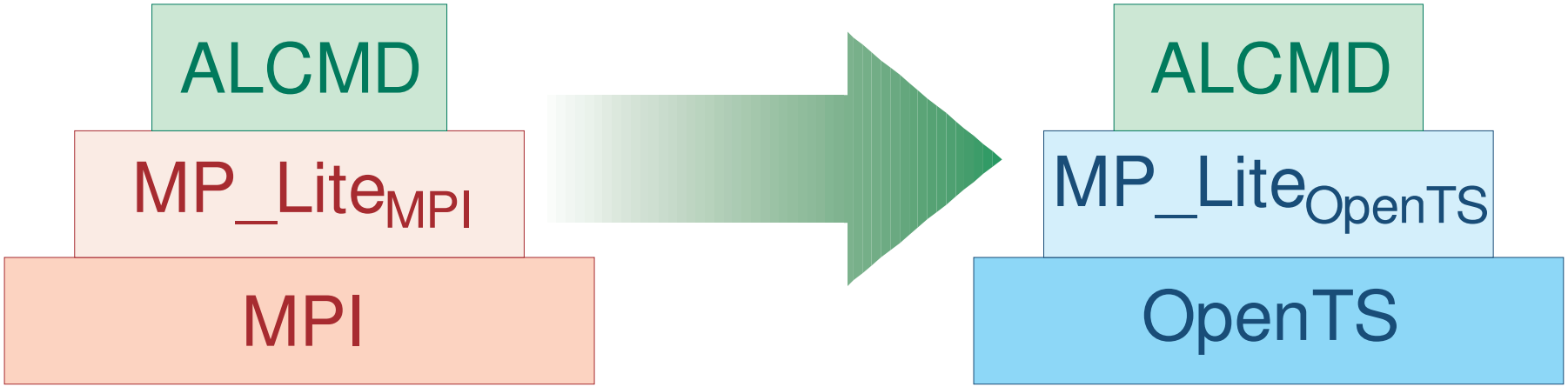
- **PovRay** – Persistence of Vision Ray-tracer, enabled for parallel run by a patch
- **ALCMD/MP_lite** – molecular dynamics package (Ames Lab)

T-PovRay vs MPI PovRay: code complexity

Program	Source code volume
MPI modules for PovRay 3.10g	1,500 lines
MPI patch for PovRay 3.50c	3,000 lines
T++ modules (for both versions 3.10g & 3.50c)	200 lines

ALCMD/MPI vs ALCMD/OpenTS

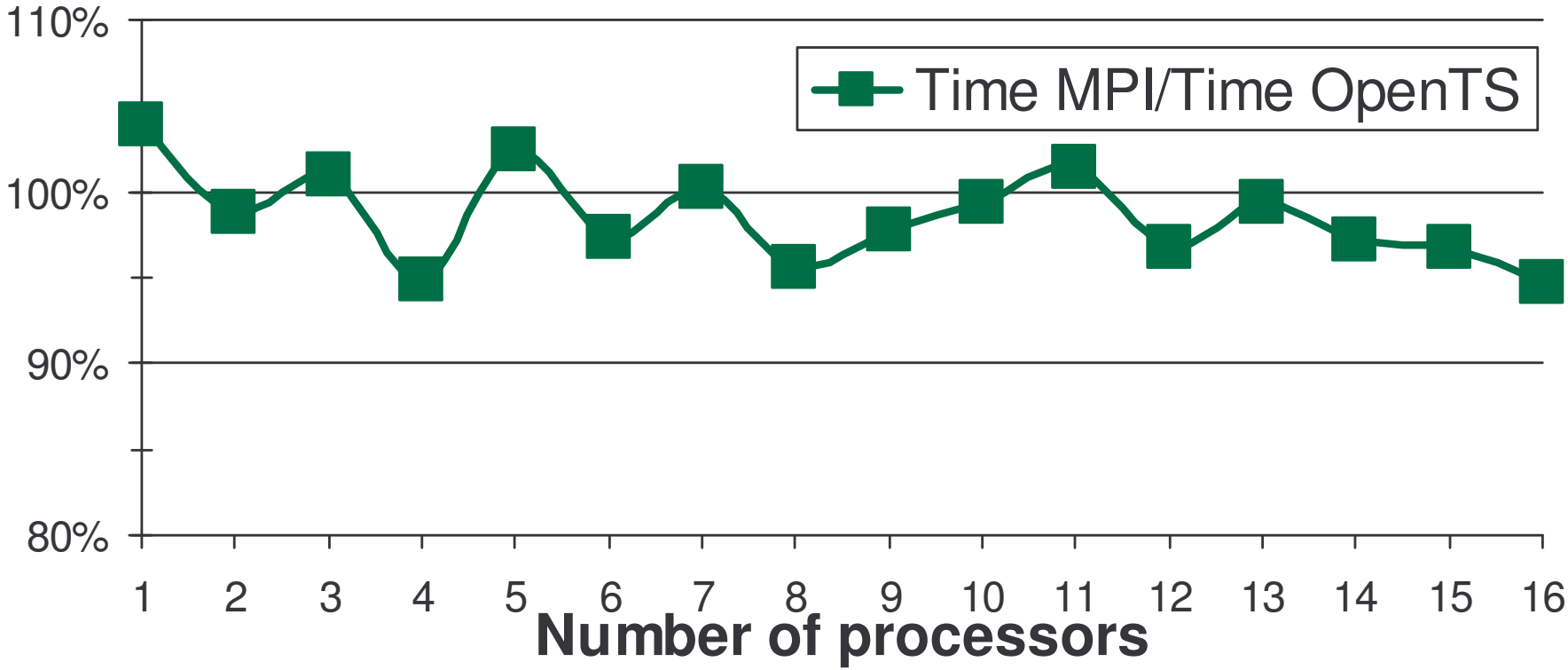
- MP_Lite component of ALCMD rewritten in T++
- Fortran code is left intact



ALCMD/MPI vs ALCMD/OpenTS : code complexity

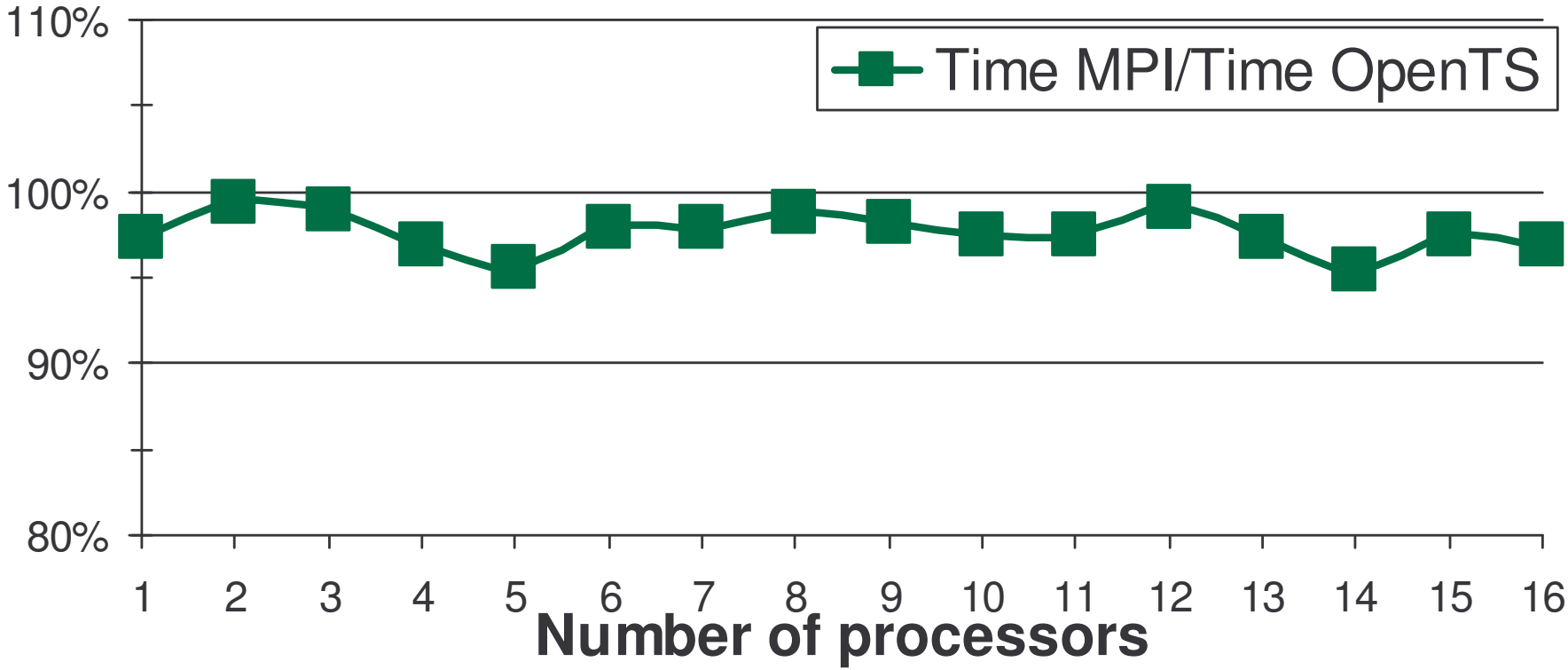
Program	Source code volume
MP_Lite total/MPI	~20,000 lines
MP_Lite,ALCMD-related/ MPI	~3,500 lines
MP_Lite,ALCMD-related/ OpenTS	500 lines

ALCMD/MPI vs ALCMD/OpenTS: performance



2CPUs AMD Opteron 248 2.2 GHz RAM 4GB,
GigE, LAM 7.1.1, Lennard-Jones MD, 512000 atoms

ALCMD/MPI vs ALCMD/OpenTS: performance



2CPUs AMD Opteron 248 2.2 GHz RAM 4GB,
InfiniBand,MVAMPICH 0.9.4, Lennard-Jones MD,512000 atoms

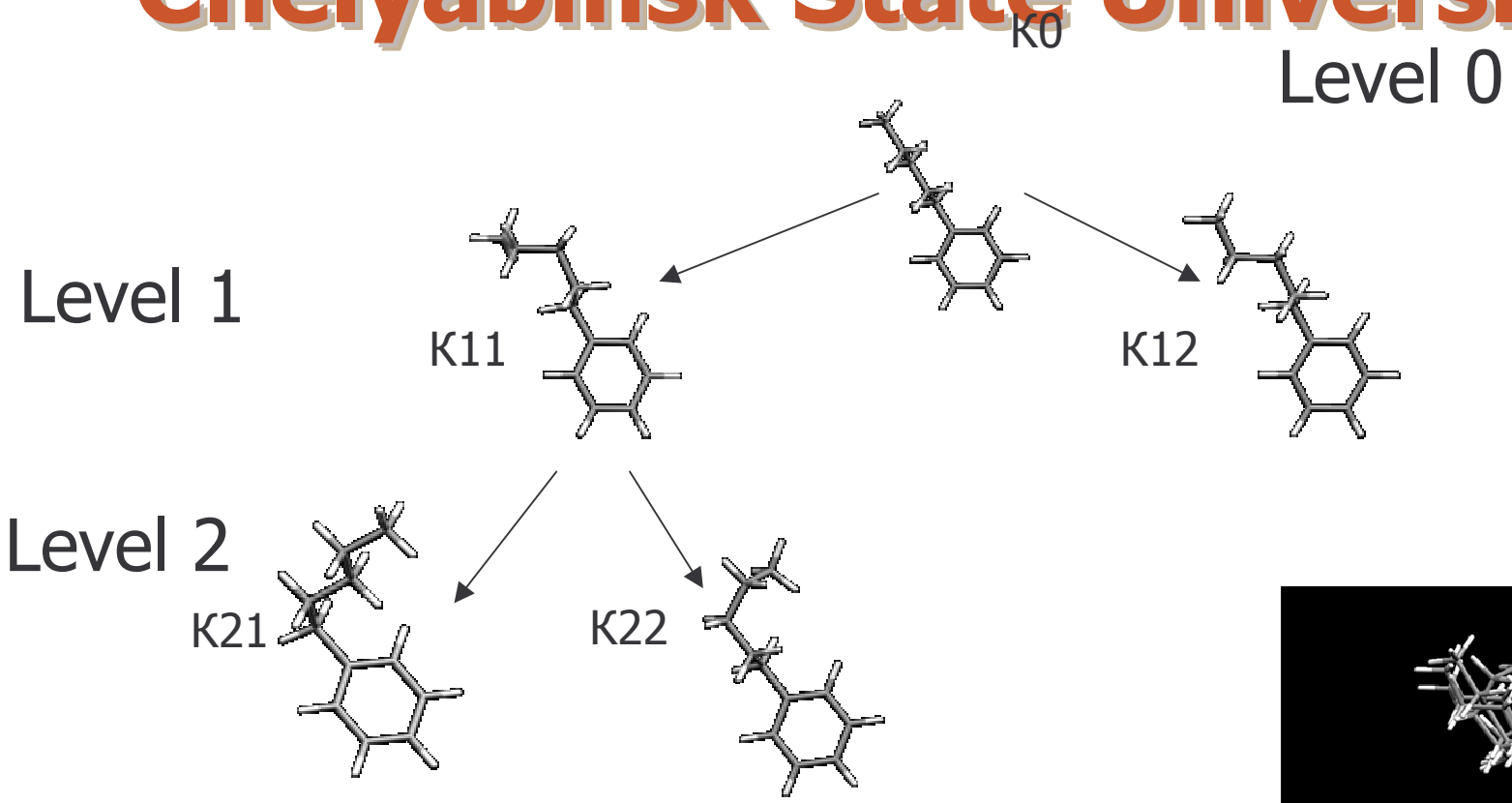
Open TS applications

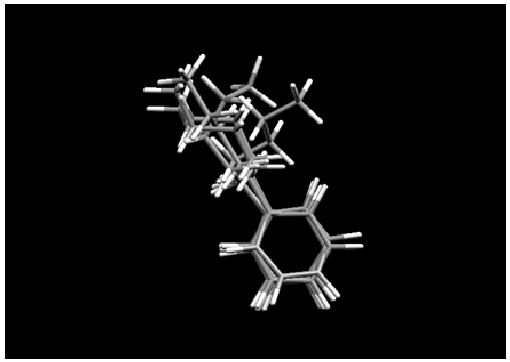
T-Applications

- MultiGen – biological activity estimation
- Remote sensing applications
- Plasma modeling
- Protein simulation
- Aeromechanics
- Query engine for XML
- AI-applications
- etc.

MultiGen

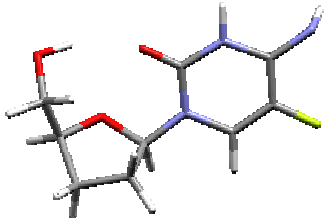
Chelyabinsk State University



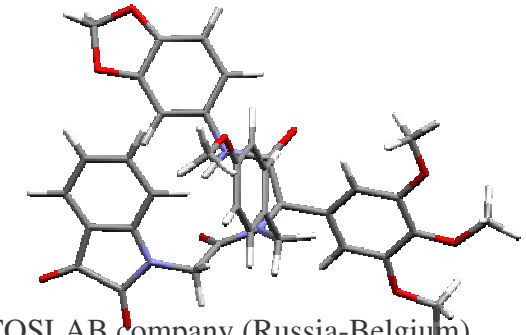


Multi-conformation model

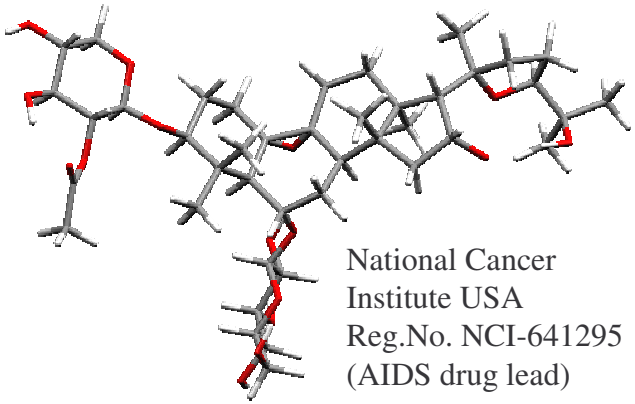
MultiGen: Speedup



National Cancer Institute USA
Reg.No. NCI-609067
(AIDS drug lead)



TOSLAB company (Russia-Belgium)
Reg.No. TOSLAB A2-0261
(antiphlogistic drug lead)



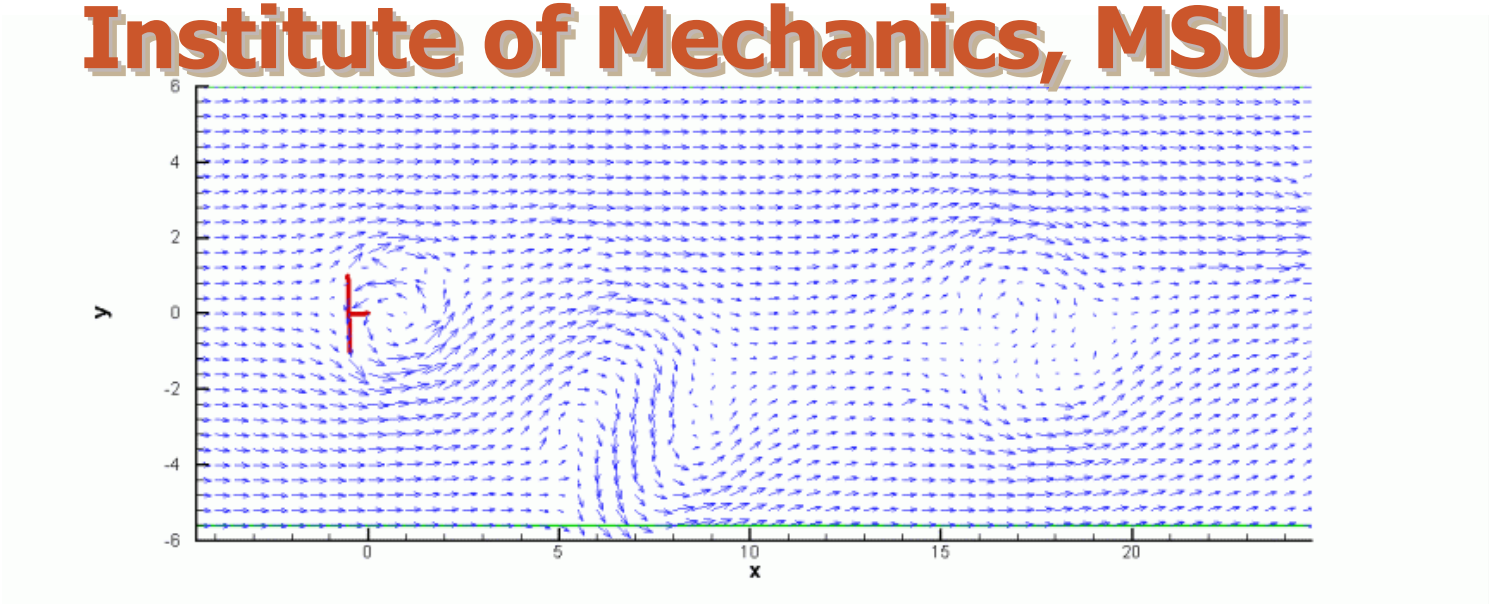
National Cancer Institute USA
Reg.No. NCI-641295
(AIDS drug lead)

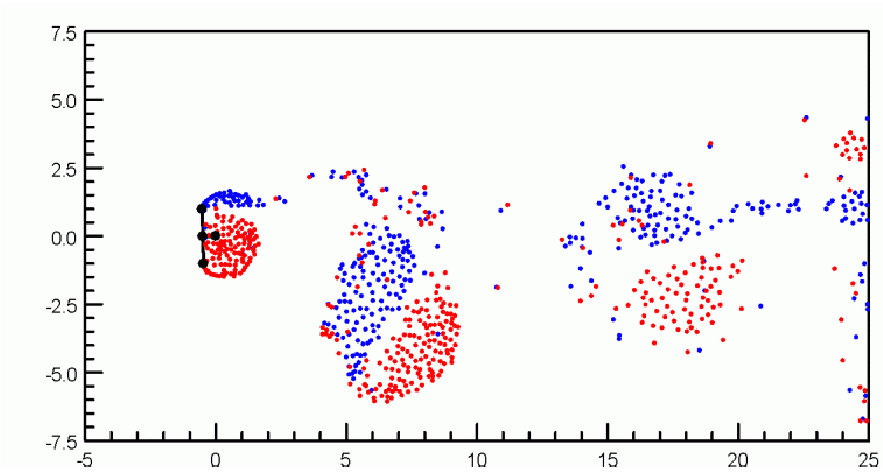
Substance	Atom number	Rotations number	Conformers	Exectution time (min.:c)		
				1 node	4 nodes	16 nodes
NCI-609067	28	4	13	**9:33**	**3:21**	**1:22**
TOSLAB A2-0261	82	18	49	**115:27**	**39:23**	**16:09**
NCI-641295	126	25	74	**266:19**	**95:57**	**34:48**

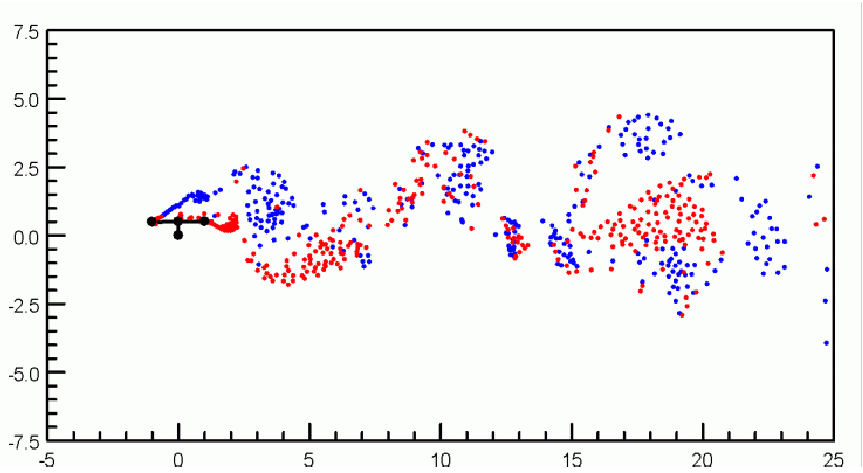
2001/12/10 11:12pm

AEROMECHANICS

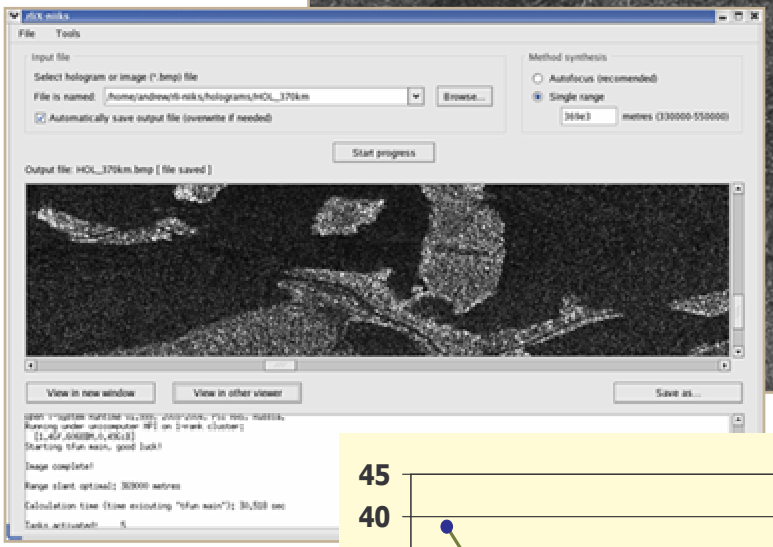
Institute of Mechanics, MSU

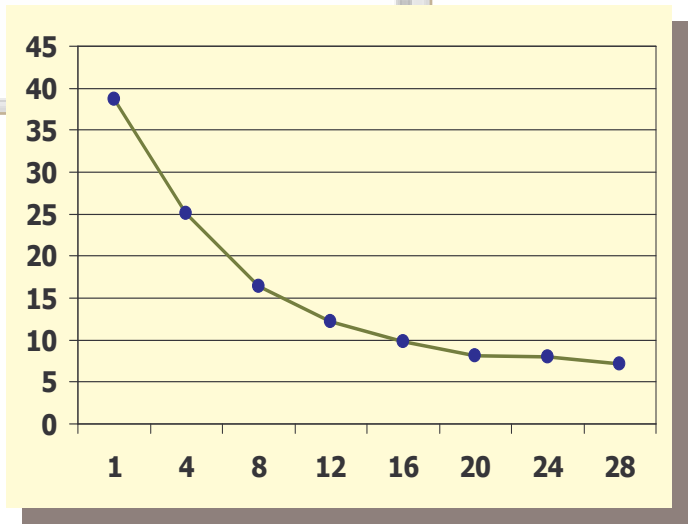


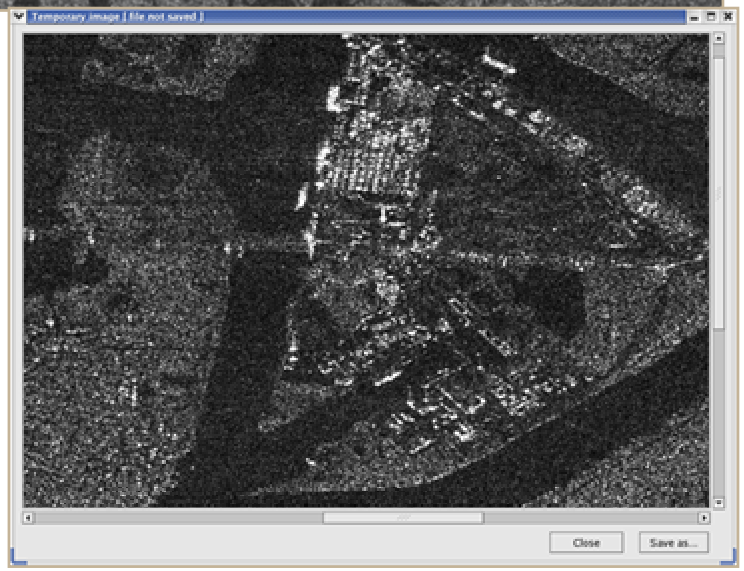




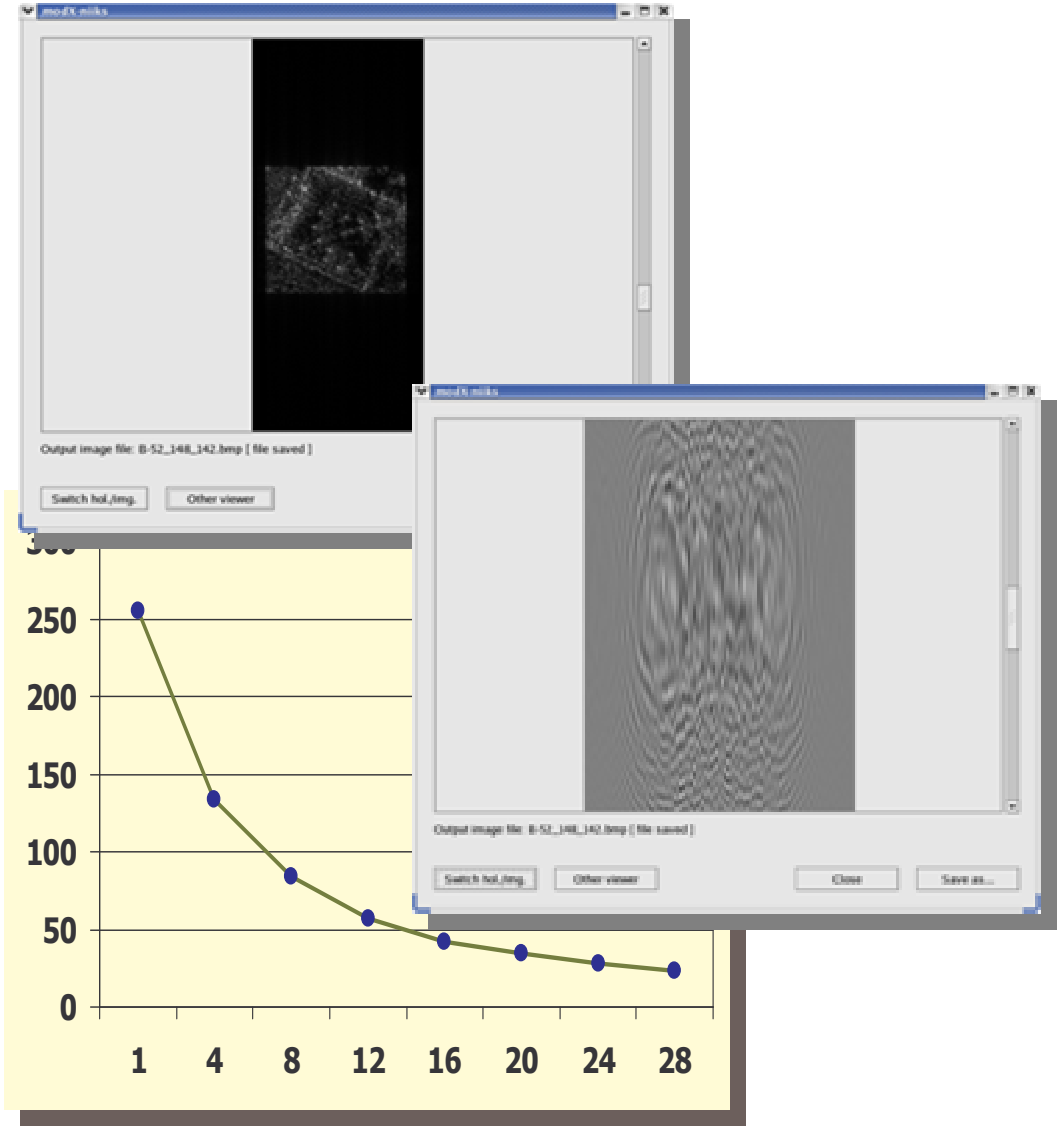
Creating space-born radar image from hologram

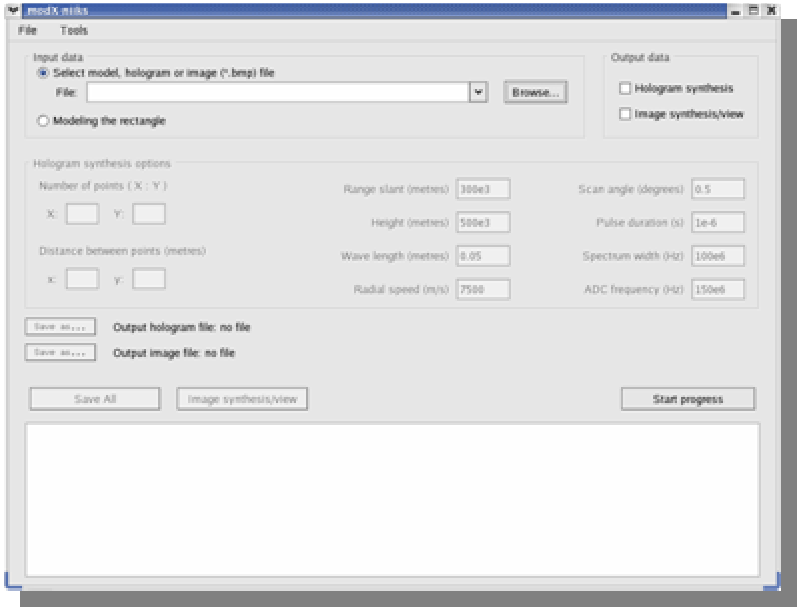






Simulating broadband radar signal





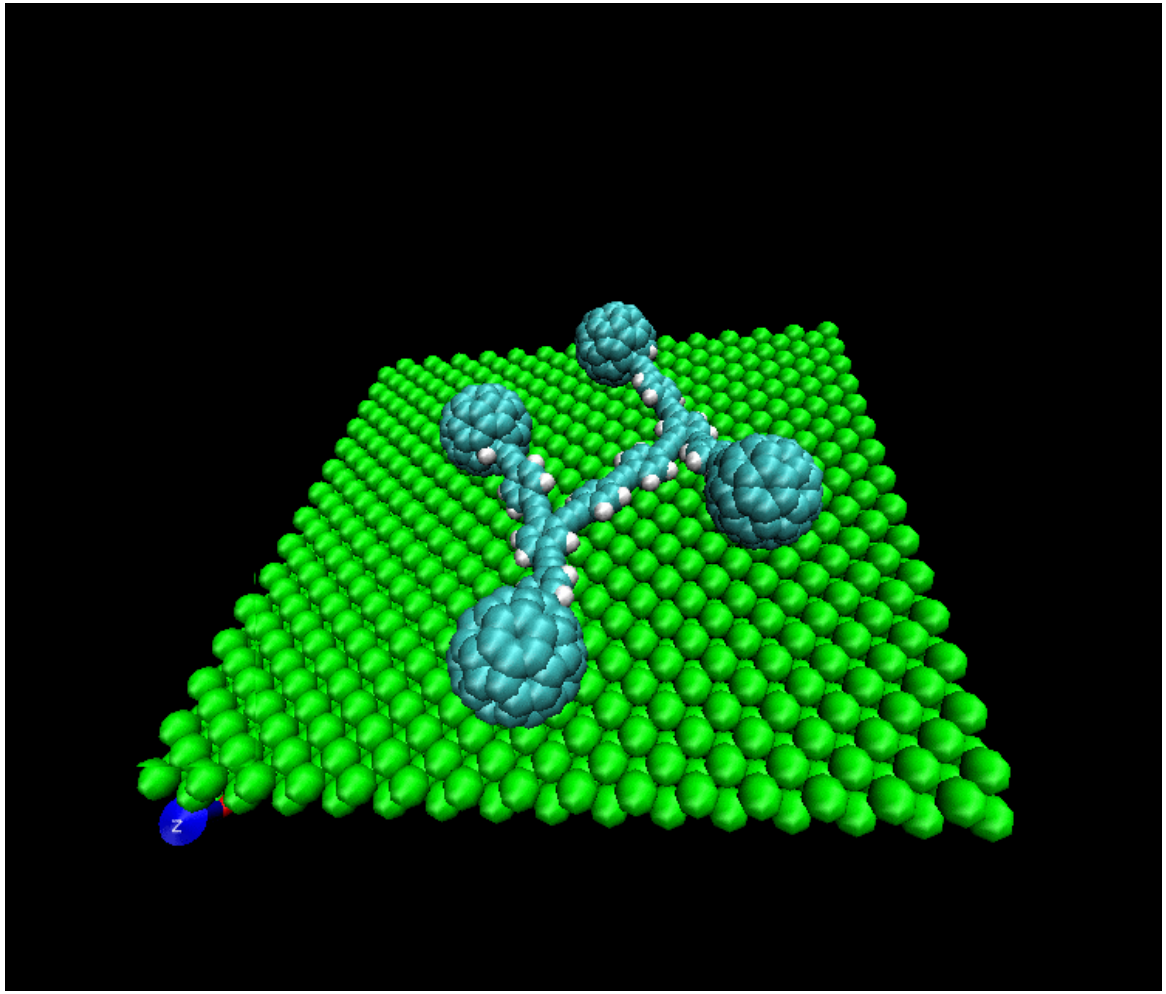
- Graphical User Interface
- Non-PSI RAS development team (Space research institute of Khrunichev corp.)

Rigid body MD

- Goal:

Enable QM/MM molecular dynamics

- Nanocars
- Global optimization



Out of Presentation Scope

- Other T-languages: T-Refal, T-Fortan
- Memoization
- Automatically choosing between call-style and fork-style of function invocation
- Checkpointing
- Heartbeat mechanism
- Flavours of data references: “normal”, “glue” and “magnetic” — lazy, eager and ultra-eager (speculative) data transfer

Future Work

- Windows 64 migration (ongoing)
- Multi-core CPU support
- Heterogenous clusters
- Distributed computing
  - Schedulers
  - Transport
  - Interface to web-services
- Fault-tolerance
- Optimizing for modern CPUs
- Algorithmic skeletons, patterns and high level parallel libraries

HPC Trends

Cycle Harvesting

Data Grids

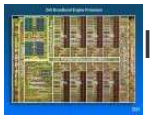
Grids

IBM Blue Gene

Massively Parallel

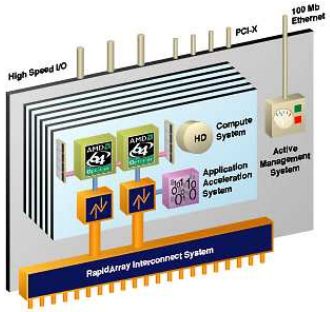
SCALE UP

GPU
PPU



IBM Cell

"Traditional"
High-performance
computing



Cray XD1
(Opteron+FPGA)

SCALE DOWN

Multi-Core
CPUs

Mini-
clusters

ACKNOLEDGEMENTS

- "SKIF" supercomputing project
- Russian Academy of Science grants
  - Program "High-performance computing systems on new principles of computational process organization"
  - Program of Presidium of Russian Academy of Science "Development of basics for implementation of distributed scientific informational-computational environment on GRID technologies"
- Russian Foundation Basic Research "05-07-08005-офи_а"
- Microsoft – contract for "Open TS vs MPI" case study, migration to Windows 64

THANKS

… … ANY QUESTIONS ???… …