



**Институт программных систем
Российской академии наук и К^о**

**Орен TS: архитектура и
реализация среды для
динамического
распараллеливания вычислений**

Абрамов С. М., Московский А. А., Роганов В. А.,
Парамонов Н. Н., Шевчук Е. В.,
Шевчук Ю. В., Чиж О. П.

Новороссийск, Абрау-Дюрсо, 2005-09-20





План доклада

- ❑ Короткое само-представление
- ❑ Open TS: обзор архитектуры
- ❑ Сравнение подходов: MPI vs Open TS
- ❑ Приложения, написанные на OpenTS
- ❑ Закругляясь:
 - ★ Что осталось за рамками доклада?
 - ★ Планов наших громадье...
 - ★ Благодарности



**Институт программных систем
Российской академии наук и К°**

1. Short Self-Introduction





Орен TS: архитектура и реализация

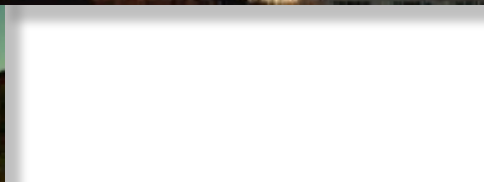
ИПС РАН, Переславль-Залесский





Орен TS: архитектура и реализация

МГУ им. М.В.Ломоносова





Партнеры

- ❑ ИПС РАН
- ❑ МГУ им. М. В. Ломоносова
- ❑ ОИПИ НАН Беларуси
- ❑ наши пользователи:
 - ★ ЧелГУ
 - ★ НИИ мех. МГУ им. М. В. Ломоносова
 - ★ НИИ КС (Хруничев)
 - ★ и др.



**Институт программных систем
Российской академии наук и К^о**

Open TS: Обзор архитектуры





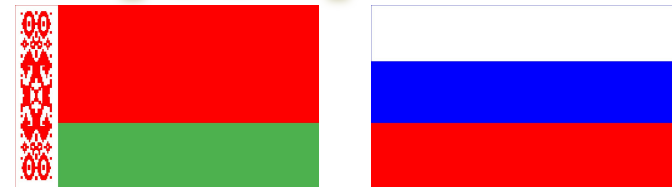
T-Система. История

- ❑ **Середина 80-ых**
Основные идеи T-Системы
- ❑ **1990-ые**
Первая реализация T-Системы
- ❑ **2000-2002, Программа «СКИФ»**
GRACE — Graph Reduction Applied to Cluster Environment
- ❑ **2003-сегодня, Программа «СКИФ»**
Open TS — Open T-system



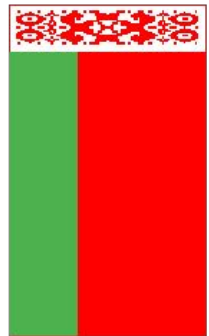
Суперкомпьютерный проект СКИФ Союзного государства

- ❑ 2000-2004
- ❑ 10 + 10 исполнителей
- ❑ \$10М (на 5 лет на 20 предприятий)
- ❑ ИПС РАН — головные по России
- ❑ ОИПИ НАН Беларуси – головные по Российской Федерации
- ❑ Hardware, Software, Applications, Aux.





Выпуск образцов (16)



“Кардиология”
9/5 G
3+1-1U+4U
Intel P-IV-1266

“Первенец”
20/11G
16-3U;
Intel P-III-600

“ВМ-5100”
48/26G
16-2U
Intel P-IV-1500

“Myrin”
89/59G
8-1U
Intel Xeon 2.8

“K-500”
717/415G
64-1U
Intel Xeon 2.8

“K-1000”
2534/2030G
288-1U; IB 4x; AMD
Opteron 248(2.2)

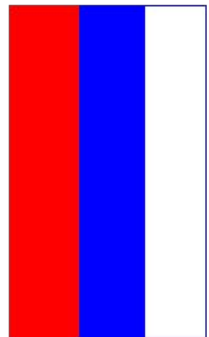
2000

2001

2002

2003

2004



“Первенец”
20/11G
16-3U;
Intel P-III-600

“Студент”
11/6G
9-MiniTower
Intel P-III-600

“ТКС”
403/230G
36-1U
Intel Xeon 2.8

“T-Bridge 8i”
45/37G
4-1U; IB 4x
Intel Itanium 2 (1.4)

“Гибрид”
2.4/1.2G
2-4U+6U;
Intel P-III-800

“Первенец-М”
98/57G
16-3U; AMD
AthlonMP1800+

“T-Forge32”
115/74G
16-2U; AMD
Opteron 224(1.8)

“T-Forge48”
230/184G
24-1U; IB 4x; AMD
Opteron (2.4)

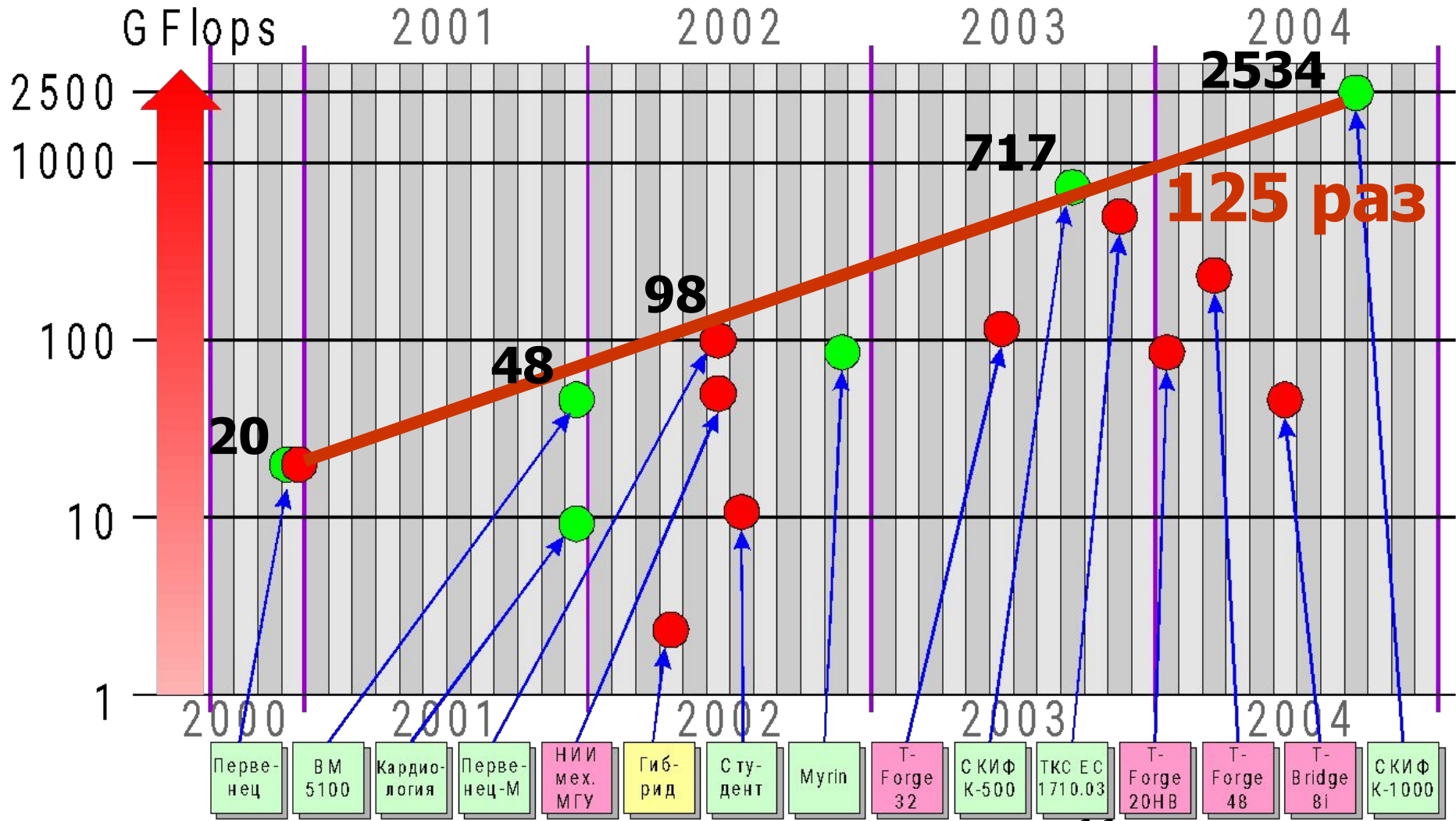
НИИ мех МГУ
49/28G
4+4-4U+5U; AMD
AthlonMP1800+

“T-Forge20HB”
88/70G
10-HB; IB 4x; AMD
Opteron 248(2.2)



Орен TS: архитектура и реализация

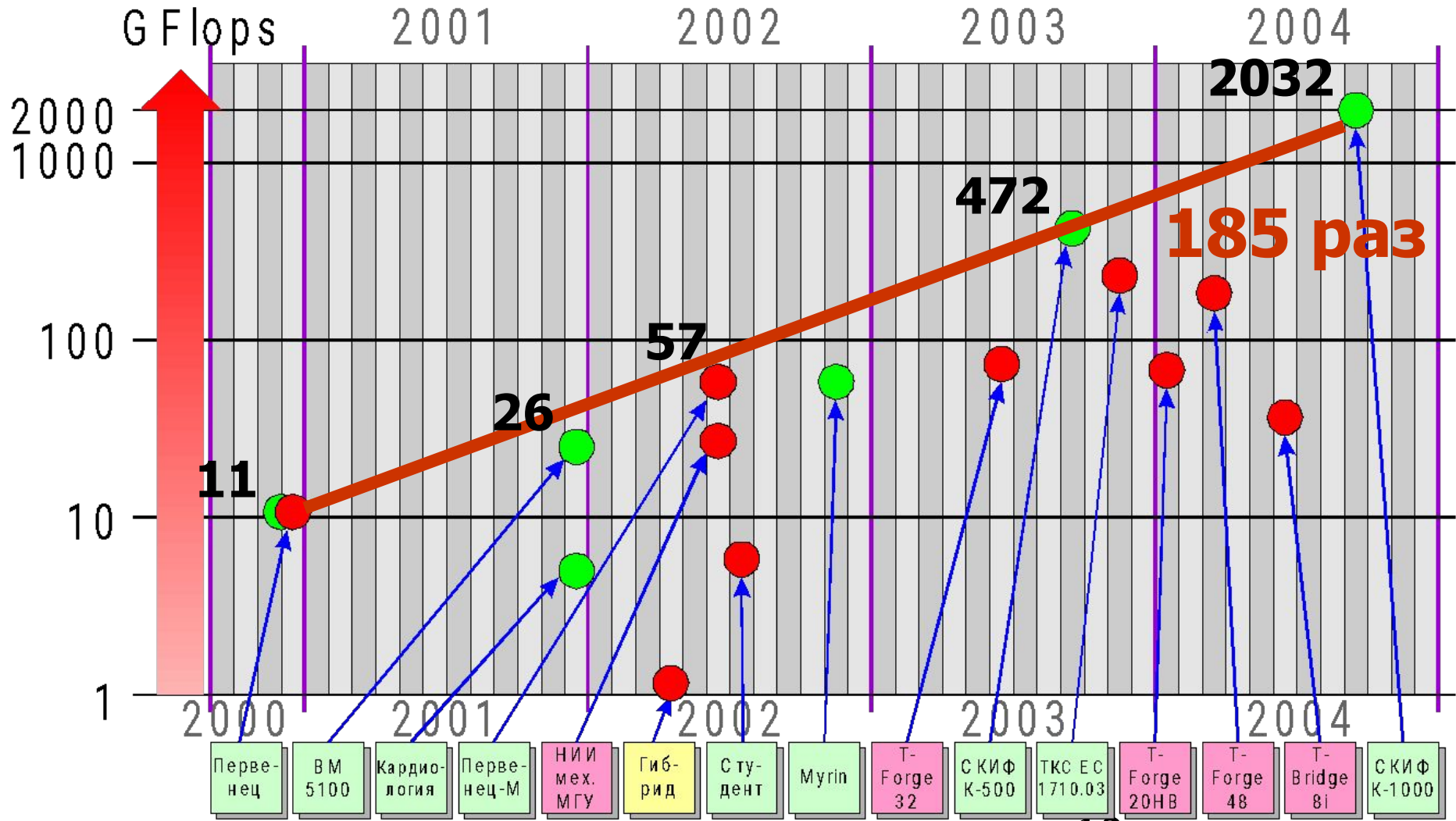
Пиковая производительность образцов





Open TS: архитектура и реализация

Linpack-производительность образцов





Темпы развития отрасли (Linpack)





Флагман: «СКИФ К-1000»



- ▣ **Пиковая производительность: 2,5 Tflops**
- ▣ **Linpack-производительность: 2,0 Tflops**
- ▣ **КПД=80.1 %**
- ▣ **Ноябрь 2004: Наиболее мощная машина на территории СССР**
- ▣ **Ноябрь 2004: № 98 в Top500**



Сравнение: T-Система и MPI

High-level

a few
keywords

C/Fortran

T-System

Low-level

hundred(s)
primitives

Assembler

MPI

Sequential

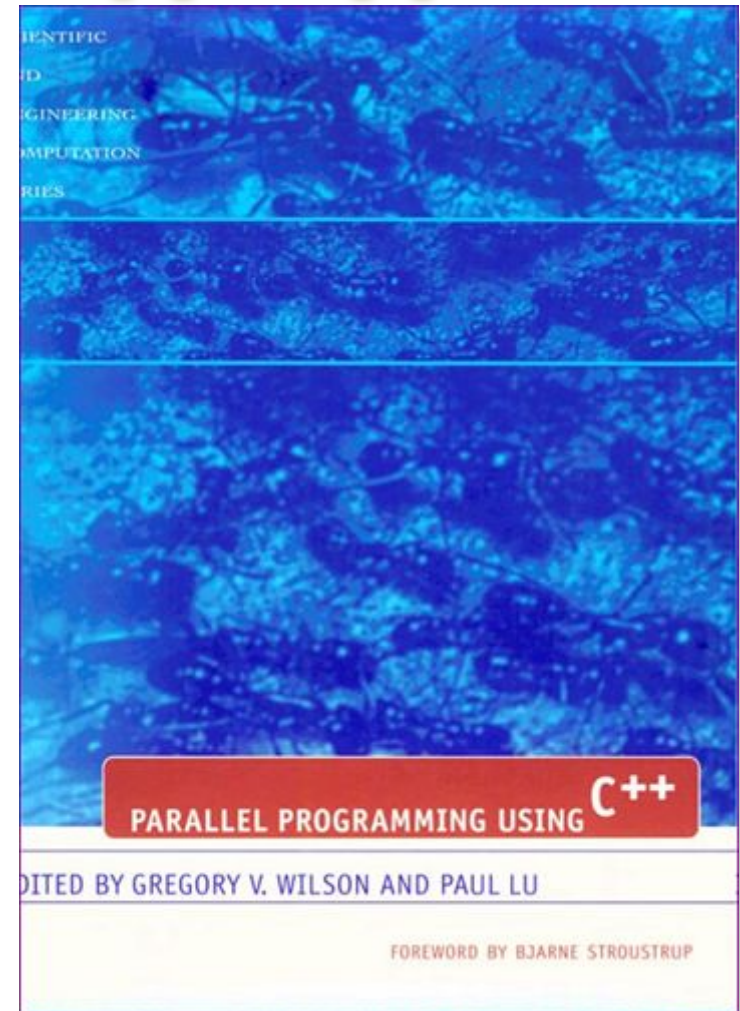
Parallel



Подобные подходы

- Parallel Programming Using C++ (Scientific and Engineering Computation) by Gregory V. Wilson (Editor), Paul Lu (Editor)

ABC++, Amelia, CC++, CHAOS++, COOL, C++//, ICC++, Mentat, MPC++, MPI++, pC++, POOMA, TAU, UC++





T-Система в сравнении

Related work	Open TS differentiator
Charm++	TS основана на FP
UPC, mpC++	В TS неявный параллелизм
Glasgow Parallel Haskell	TS допускает низкоуровневые C/C++/ASM оптимизации
OMPC++	TS дает и язык и библиотека C++ шаблонов
Cilk	TS поддерживает SMP, MPI, PVM, и (в планах) GRID



Open TS: на уровне лозунгов

- ❑ Наша цель — HPC
- ❑ Автоматическое динамическое распараллеливание программ
- ❑ Сочетание функциональной и императивной парадигм (и ООП)
- ❑ Высокоуровневое программирование
- ❑ T++ язык: «параллельный диалект» C++ (незабытое старое: популярно с 90-ых)



T-Подход

- ❑ «Чистые» функции (tfunc) — их вызовы способны породить гранулы параллелизма
- ❑ T-Программы:
 - ★ Функциональны – на верхнем уровне
 - ★ Императивны – на нижнем уровне (C/C++/ASM оптимизации)
- ❑ C-совместимая модель исполнения
- ❑ Неготовые значения, многократные присваивания
- ❑ Гладкое расширения языков: C, Fortran, Рефал



T++ новые ключевые слова

- ❑ **tfun** — T-функция
- ❑ **tval** — T-переменная (T-значение)
- ❑ **tptr** — T-указатель
- ❑ **tout** — Выходной параметр (аналог &)
- ❑ **tdrop** — Разорвать связь поставщик-потребитель (сделать готовым)
- ❑ **twait** — Редкое: ждать готовности
- ❑ **tct** — T-контекст



Пример программ

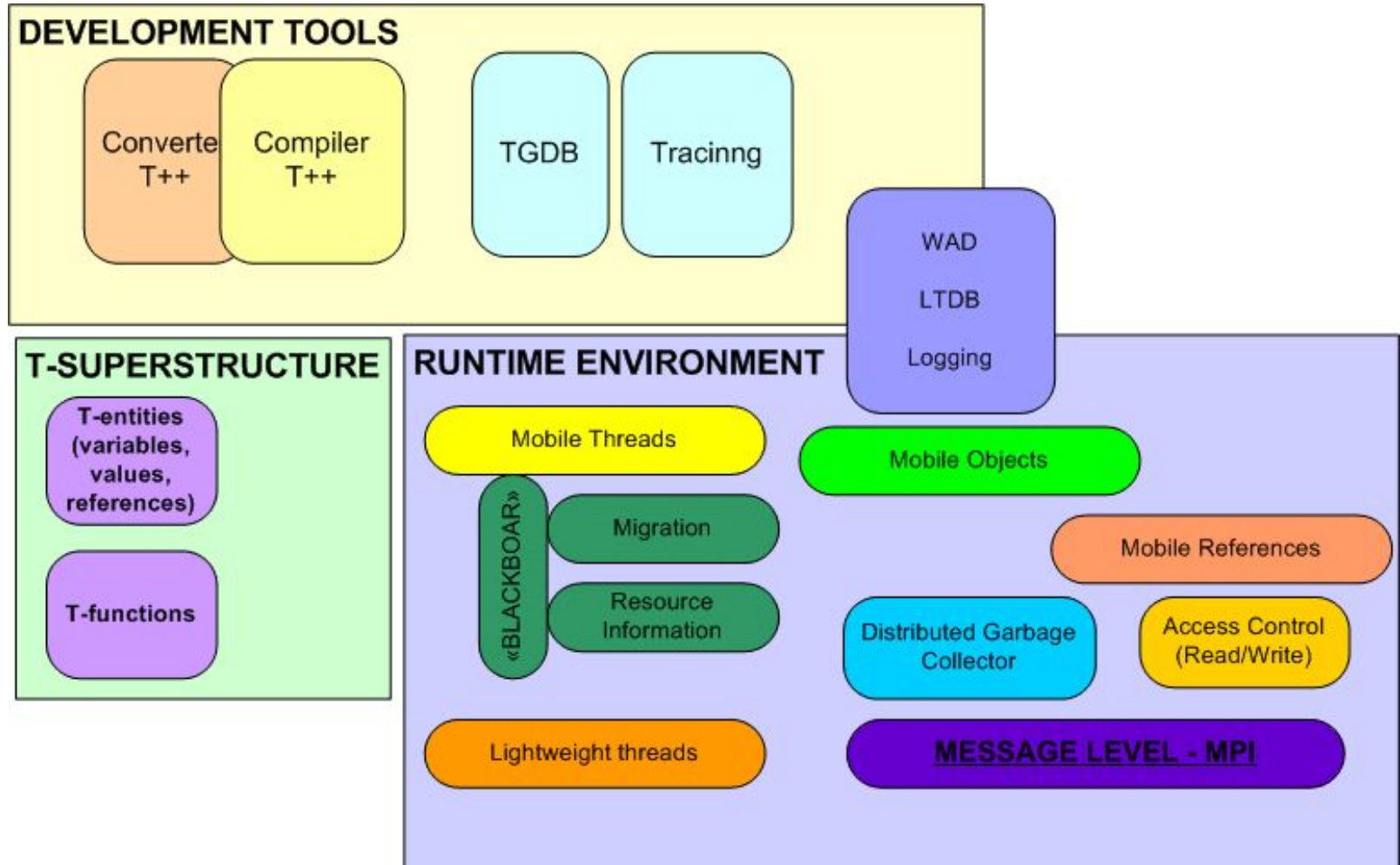
```
#include <stdio.h>
```

```
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;  
}
```



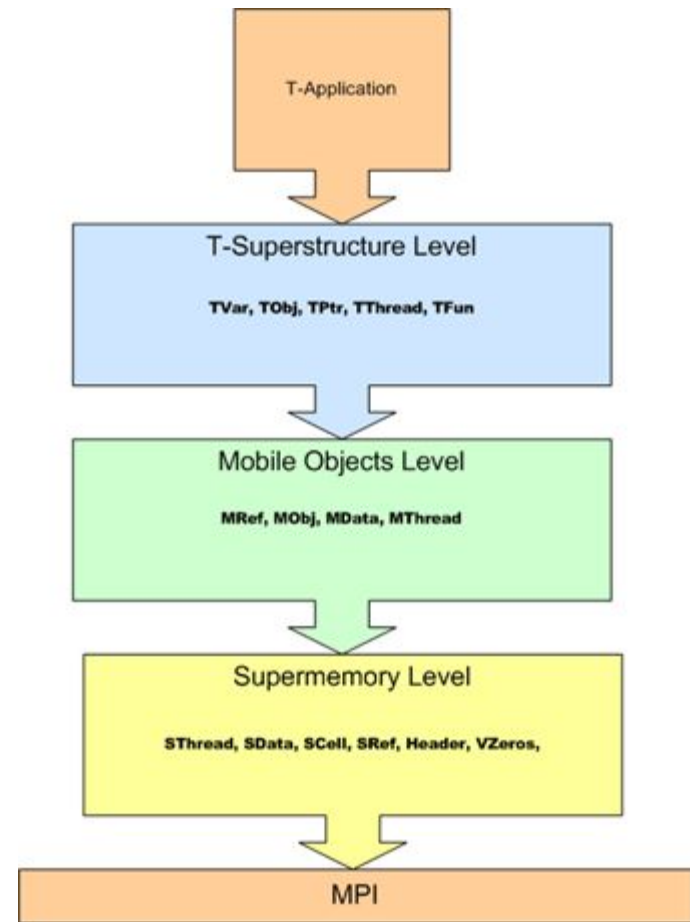
Open TS: Среда





Open TS: Runtime

- Трехслойная архитектура (T, M, S)
- Design: microkernel
Сегодня: 10 расширений
- «Supermemory»
- Lightweight threads
- DMPI: Dynamic MPI
 - ★ auto selection of MPI implementation
 - ★ dynamic loading and linking





Supermemory

- ❑ Object-Oriented Distributed shared memory (OO DSM)
- ❑ Global address space
- ❑ Cell versioning



Multithreading & Communications

- ❑ **Lightweight threads** — провокация
 - ★ PIXELS (1 000 000 threads)
- ❑ **Asynchronous** communications
 - ★ Нити **A** требуется неготовое значение
 - ★ Передается асинхронный запрос (Active messages & Signals) чтобы стимулировать передачу данных к нити **A**
 - ★ Выделяется квант на коммуникации (нет ли чего в нашем процессоре?) и переход (context switch) на другую готовую нить
- ❑ **Latency Hiding** в коммуникациях



DMPI

- ❑ **Dynamic MPI**
 - ★ автоматический подбор реализации MPI
 - ★ динамическая загрузка (dynamic loading and linking)
- ❑ **Семь реализаций MPI поддерживаны:**
 - ★ LAM
 - ★ MPICH
 - ★ SCALI MPI
 - ★ MVARICH
 - ★ IMPI
 - ★ MPICH-G2
 - ★ PACX-MPI
- ❑ **И даже PVM** может быть использован вместо MPI



Debugging: WAD, LTDB

```
[var@skif tests]$ cat fault.txx
void writeNull(unsigned long ptr) {
    *(unsigned long*)ptr = 0;
}

tfun int main (int argc, char *argv[]) {
    writeNull(7);
    return 0;
}
[var@skif tests]$ ./fault
Open T-System Runtime v3.0, 2003-2004, PSI RAS, Russia.
Running under unicomputer MPI on 1-rank cluster:
  [3.3Gf,3322BM,0.86GiB]
Starting tfun main, good luck!

WAD: Collecting debugging information...
WAD: Segmentation fault.
#13 0x08049945 in ?()
#12 0x40170a67 in __libc_start_main()
#11 0x0804bd65 in unsigned long()
#10 0x0804a511 in ts::start_threads(void* (*)(void*), int)()
#9 0x0804b7c1 in ts::thread_worker(void*)()
#8 0x08050445 in ts::Service::doAllWork()()
#7 0x0805e56a in ts::MacroScheduler::work()()
#6 0x08061ffe in ts::MacroScheduler::runLocalTasks()()
#5 0x0805057c in ts::ThrH::yield(ts::ThrH*)()
#4 0x0804ad6f in ts::ThrH::hwSaveRestore(ts::ThrH*, ts::ThrH*, void*)()
#3 0x0804b139 in ts::SThread::newTask(ts::SThread*)()
#2 0x0806b5d9 in ts::TFun<int, tfunmainTFunCtxt>::work()(this=0x8c1583c)
    ~ in , at line 5063
#1 0x0806a28e in tfunmainTFunImpl::body()(this=0x8c1583c)
    ~ in , at line 6
#0 0x0806a1ae in writeNull(unsigned long)(ptr=7)
    ~ in , at line 2
[var@skif tests]$
```



Сбор статистики

```
[alexanderm@skif demos]$ mpirun nl,2,3,4 ./fib 29
Open T-System Runtime v3.0, 2003-2004, PSI RAS, Russia.
Running under LAM MPI on 4-rank cluster:
  ([3.1Gf,3060BM,0.86GiB]+[3.1Gf,3060BM,0.49GiB]+[3.1Gf,3060BM,0.86GiB]*2) ~= [1
2.2Gf,12240BM,3.08GiB]
Starting tfun main, good luck!

fib(29) = 514229
Tasks activated:      [407582/416020/420993]
Tasks exported:      [33/35/40]
Msgs sent:           [1534/1577/1639]
Async Msgs:          [0/0/0]
Msgs size:           [114472/118405/124576]
Taskboard visits:    [408080/416657/421748]
Scheduler time:      [1.188/1.219/1.234]
MPI time:            [0.031/0.033/0.035]
Idle time:           [0.011/0.013/0.015]
Tasks time:          [8.942/9.059/9.190]
Total time:          [16.368/16.375/16.380]
```



Сообщения из разных мест

```
var@skif:~/autoInstallOpenTS/openTS/demos
File Edit View Terminal Go Help
[4,0] -> [7]: ts::Data<ts::controlHandler>/readRq @0x8a9906c[60]
[9,0] -> [10]: ts::Data<ts::controlHandler>/readRq @0x8a9906c[60]
[9,0] -> [14]: ts::Data<ts::controlHandler>/readRq @0x8a98164[60]
[4,0] -> [7]: ts::Data<ts::controlHandler>/readRq @0x8a9906c[60]
[9,0] -> [7]: ts::Data<ts::controlHandler>/readRq @0x8a98164[60]
[9,0] -> [8]: ts::Data<ts::controlHandler>/readRq @0x8a98164[60]
[1,0] -> [14]: ts::Data<ts::controlHandler>/readRq @0x8a9948c[60]
[7,0] -> [13]: ts::Data<ts::controlHandler>/readRq @0x8a98d5c[60]
[5,0] -> [15]: ts::Data<ts::controlHandler>/readRq @0x8a98164[60]
[8,0] -> [1]: ts::Data<ts::controlHandler>/readRq @0x8a98164[60]
[1,0] -> [5]: ts::Data<ts::controlHandler>/readRq @0x8a9948c[60]
[7,0] -> [4]: ts::MControlData<ts::Resource> @0x8a9887c[80]
[5,0] -> [7]: ts::MControlData<ts::Resource> @0x8a987ac[80]
[8,0] -> [4]: ts::MControlData<ts::Resource> @0x8a98114[80]
[5,0] -> [8]: ts::MControlData<ts::Resource> @0x8a987ac[80]
[1,0] -> [7]: ts::Data<ts::controlHandler>/readRq @0x8a98f04[80]
[6,0] -> [10]: ts::Data<ts::controlHandler>/readRq @0x8a98114[80]
[5,0] -> [14]: ts::MControlData<ts::Resource> @0x8a987ac[80]
[1,0] -> [6]: ts::MControlData<ts::Resource> @0x8a98f04[80]
[6,0] -> [8]: iteratorTFunImpl @0x8a9d544[328]
[8,0] -> [12]: ts::MControlData<ts::Resource> @0x8a98114[80]
[8,0] -> [13]: ts::MControlData<ts::Resource> @0x8a98114[80]
[6,2] -> [10]: ts::DRCObj::DRCData/retrieve @0x8a98cac[68]
[6,0] -> [8]: ts::Data<ts::controlHandler>/readRq @0x8a9904c[60]
[1,0] -> [7]: ts::MControlData<ts::Resource> @0x8a98f04[80]
[13,0] -> [2]: ts::Data<ts::controlHandler>/readRq @0x8a9892c[60]
```



Open TS на территориально-распределенных установках

- ❑ Meta-cluster messaging support (MPICH-G2, IMPI, PACX-MPI)
- ❑ Customizable scheduling strategies (network topology information used)



**Институт программных систем
Российской академии наук и К°**

Контракт с Microsoft: Open TS vs MPI case study





Приложения

- ❑ Популярные и широко используемые
 - ❑ Разработаны независимыми MPI-экспертами (без порочащих связей с Т-Системой)
-

- ❑ **PovRay** – Persistence of Vision Ray-tracer, C-пакет + C/MPI-patch
- ❑ **ALCMD/MP_lite** – молекулярная динамика (Ames Lab)
Фортран программа + MP_Lite/MPI



Ключевой вопрос:

- Позволяет ли T-Система *удобно* создавать прикладные системы?
- Экономится ли труд программиста?
- Действительно ли более читабельный и более компактный код? (less space for bugs)
- И при этом мы не сильно жертвуем производительностью (до 30% от MPI)?

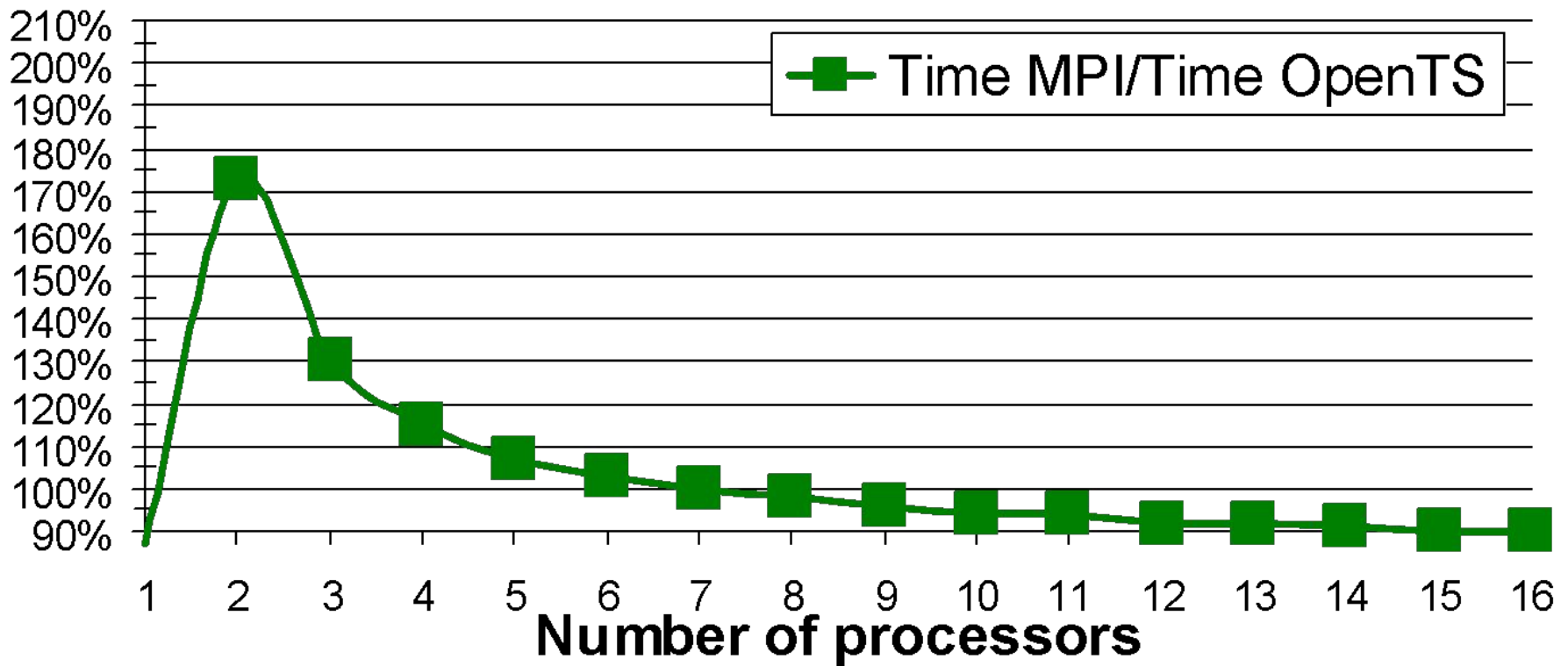


T-PovRay vs MPI PovRay: СЛОЖНОСТЬ КОДА

Программа	Объем кода
MPI modules for PovRay 3.10g	1,500 строк
MPI patch for PovRay 3.50c	3,000 строк
T++ modules (for both versions 3.10g & 3.50c)	200 строк



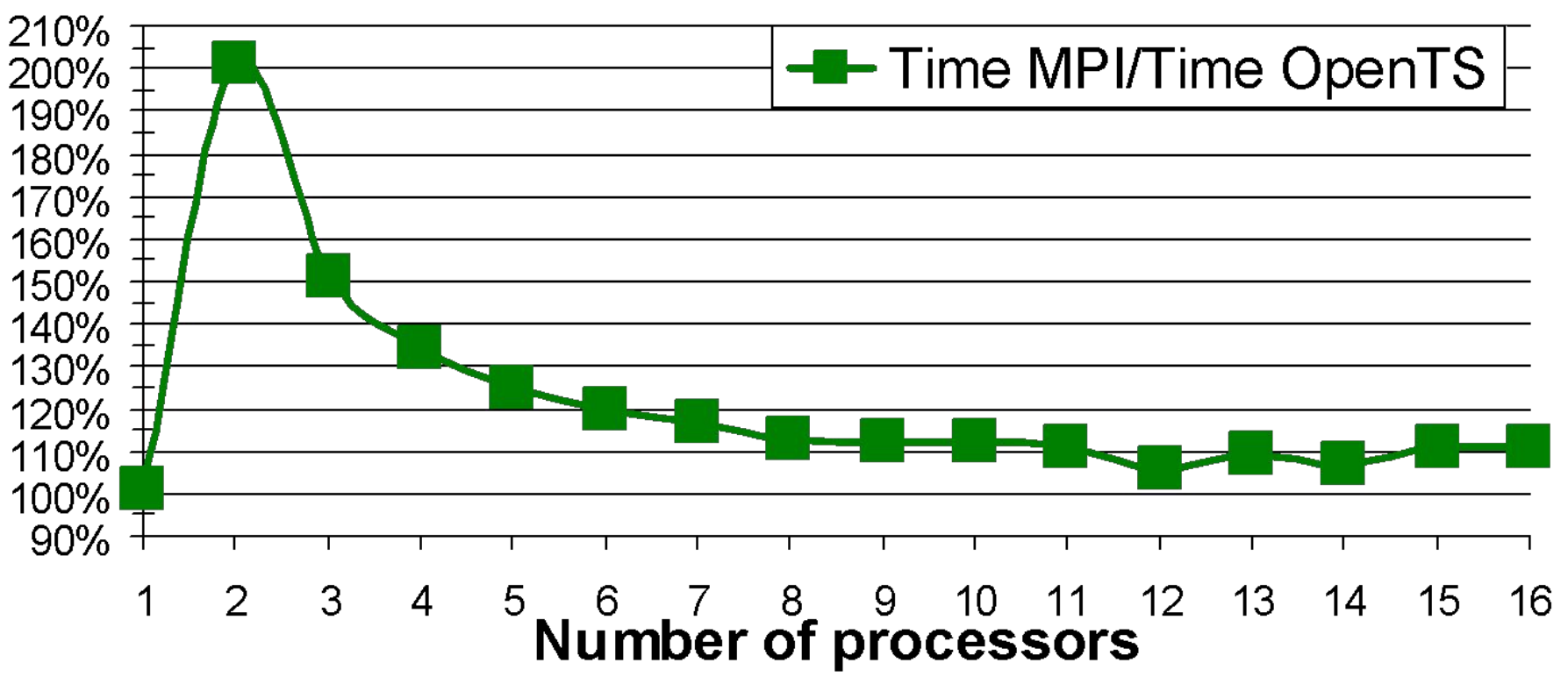
T-PovRay vs MPI PovRay: производительность



16 dual Athlon 1800, AMD Athlon MP 1800+ RAM 1GB,
FastEthernet, LAM 7.0.6



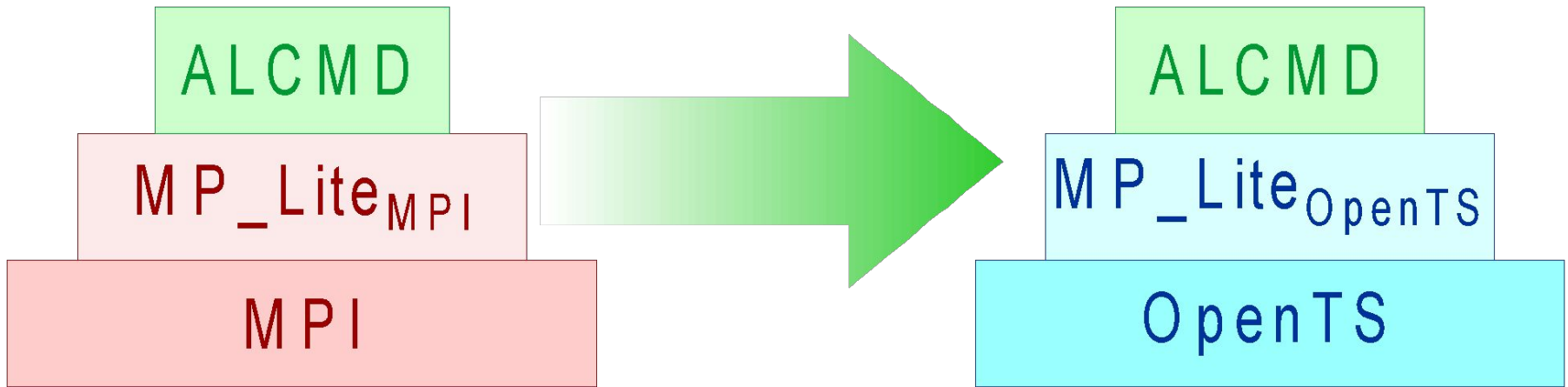
T-PovRay vs MPI PovRay: производительность



2CPUs AMD Opteron 248 2.2 GHz RAM 4GB,
GigE, LAM 7.1.1



ALCMD/MPI vs ALCMD/OpenTS



- ❑ Библиотека MP_Lite (кусочек) переписана на T++
- ❑ Fortran код ***остался нетронутым***



Ключевой вопрос:

- ❑ Позволяет ли T-Система *удобно* создавать **библиотеки (подобные MP_Light)** для дальнейшей разработки прикладных систем?
- ❑ Экономится ли труд программиста?
- ❑ Действительно ли более читабельный и более компактный код? (less space for bugs)
- ❑ И при этом мы не сильно жертвуем производительностью (до 30% от MPI)?



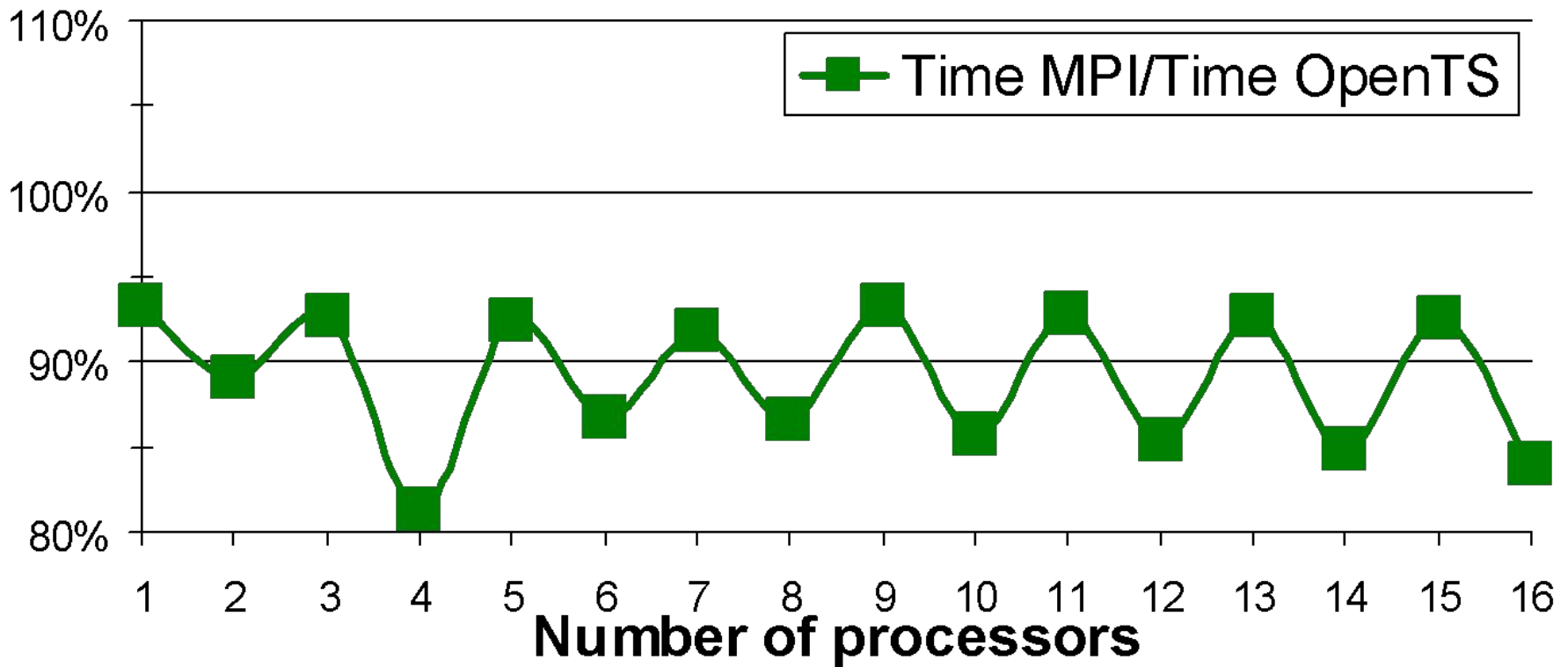
ALCMD/MPI vs ALCMD/OpenTS : code complexity

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



Open TS: архитектура и реализация

ALCMD/MPI vs ALCMD/OpenTS: производительность

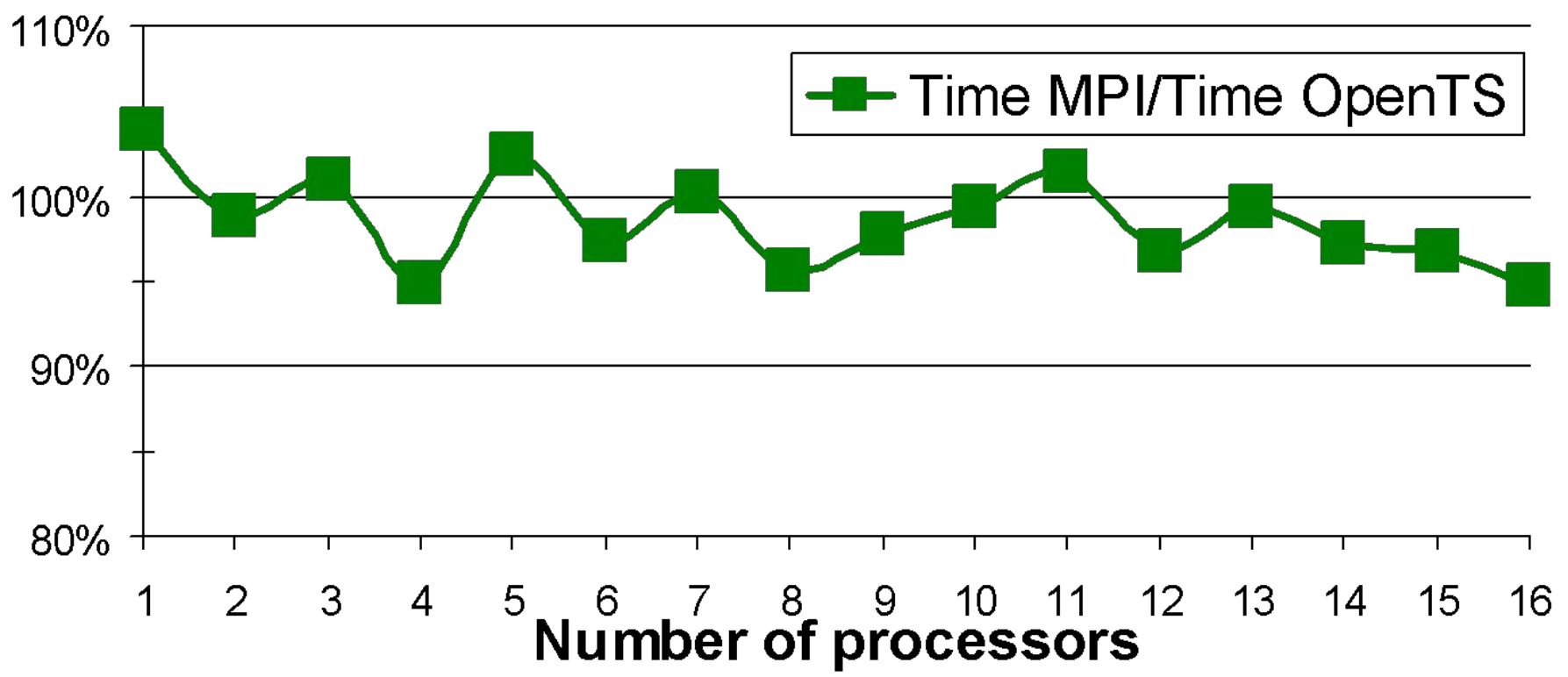


16 dual Athlon 1800, AMD Athlon MP 1800+ RAM 1GB,
FastEthernet, LAM 7.0.6, Lennard-Jones MD₄₀ 12000 atoms



Open TS: архитектура и реализация

ALCMD/MPI vs ALCMD/OpenTS: производительность

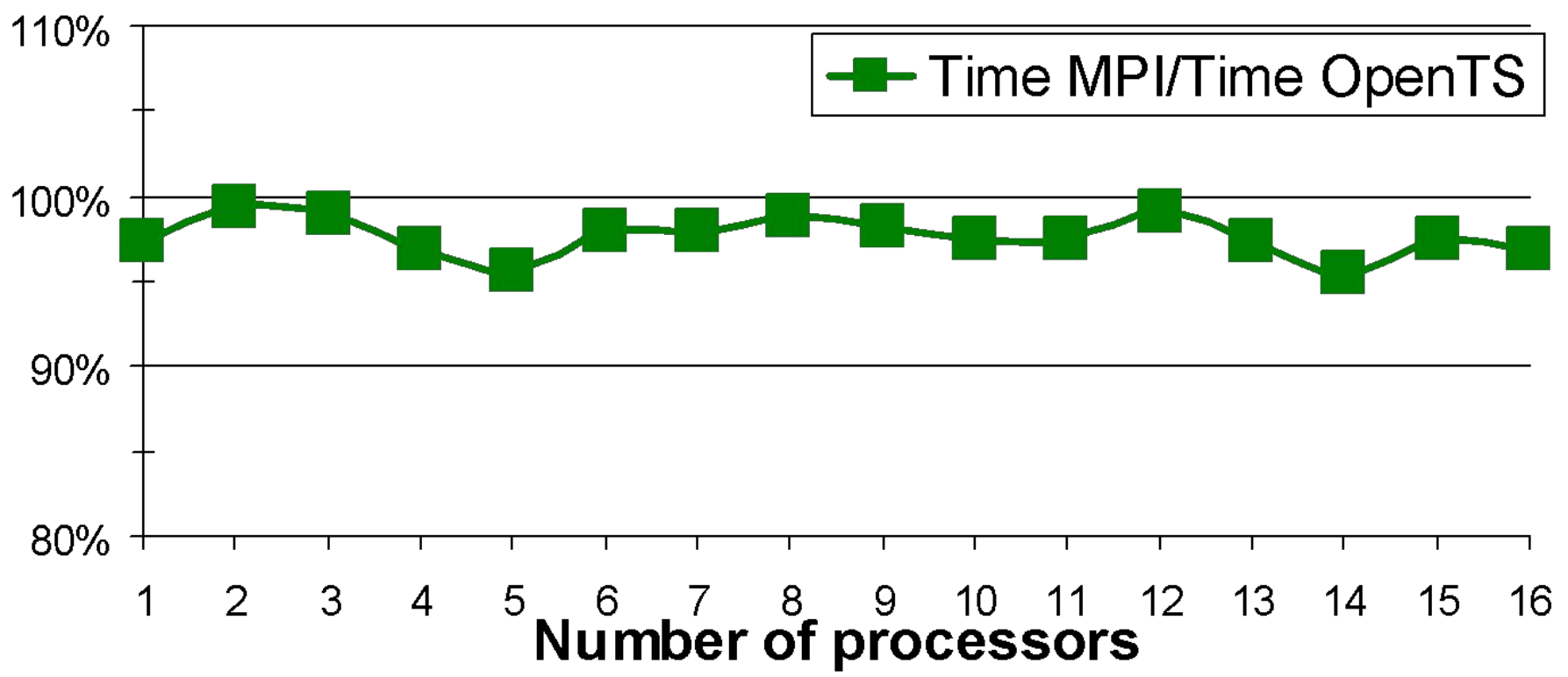


2CPUs AMD Opteron 248 2.2 GHz RAM 4GB,
GigE, LAM 7.1.1, Lennard-Jones MD, 512000 atoms



Open TS: архитектура и реализация

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



**Институт программных систем
Российской академии наук и К°**

Приложения, написанные на Open TS





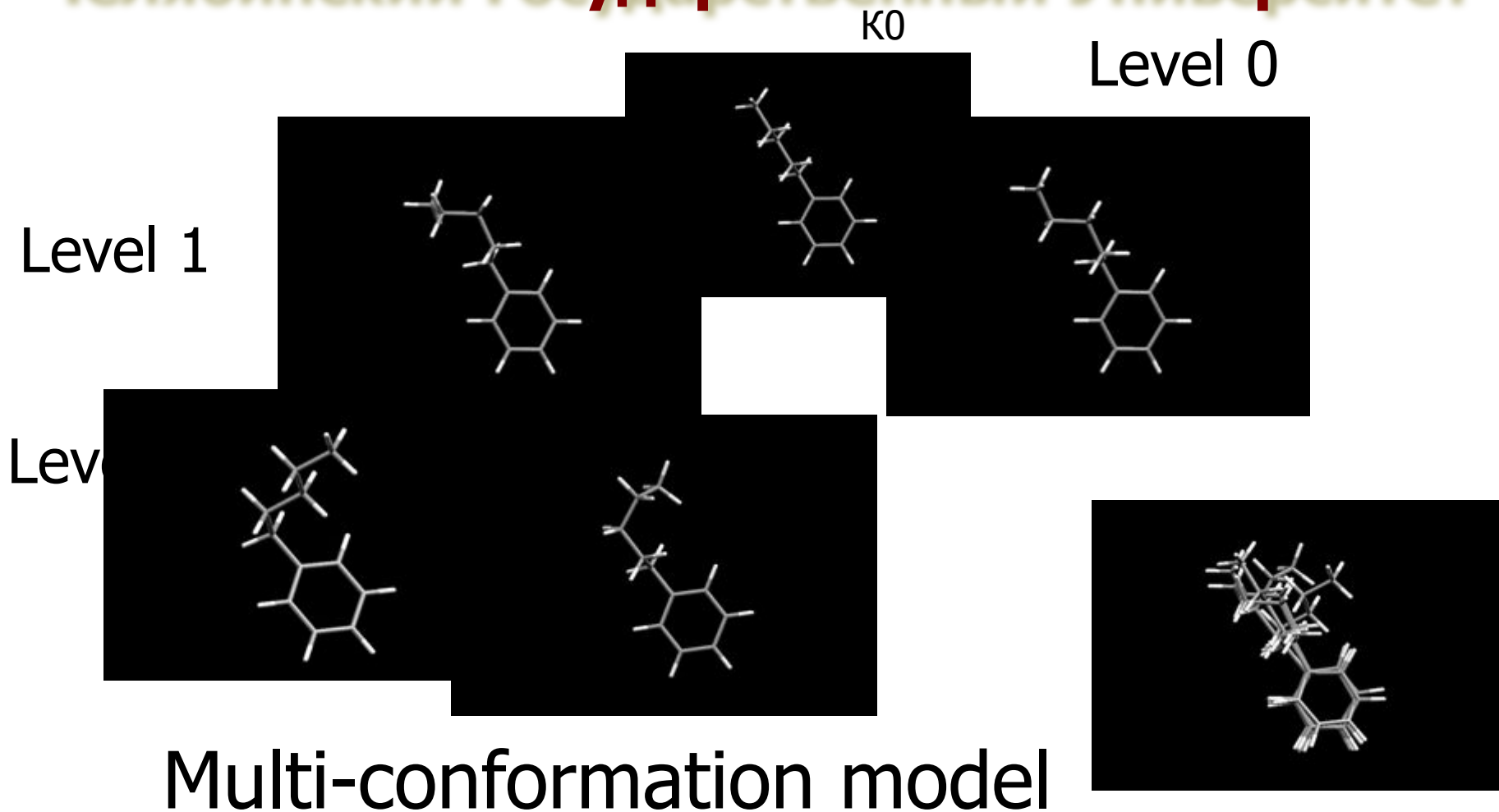
Т-Приложения

- ❑ MultiGen – оценка биологической активности веществ
- ❑ Дистанционное зондирование Земли (ДЗЗ)
- ❑ Моделирование плазмы
- ❑ Моделирование белков
- ❑ Аэромеханика
- ❑ Query engine for XML
- ❑ ИИ-приложения (3 штуки)
- ❑ и др.



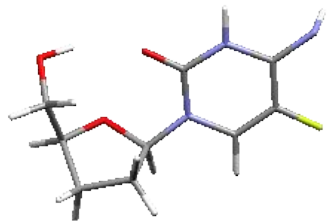
MultiGen

Челябинский Государственный Университет

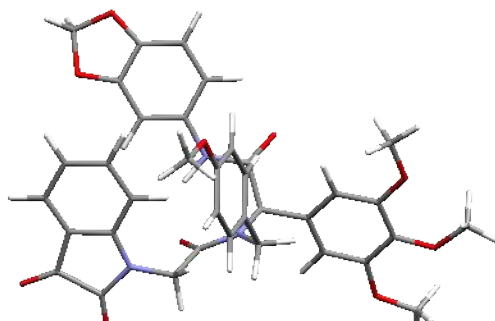




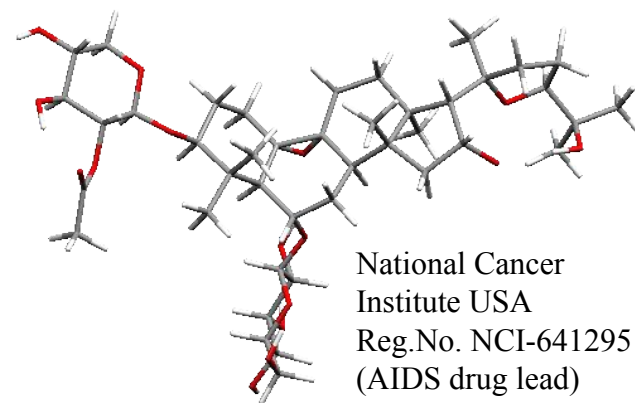
MultiGen: Speedup



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	Execution 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

46



Орен TS: архитектура и реализация

Аэромеханика

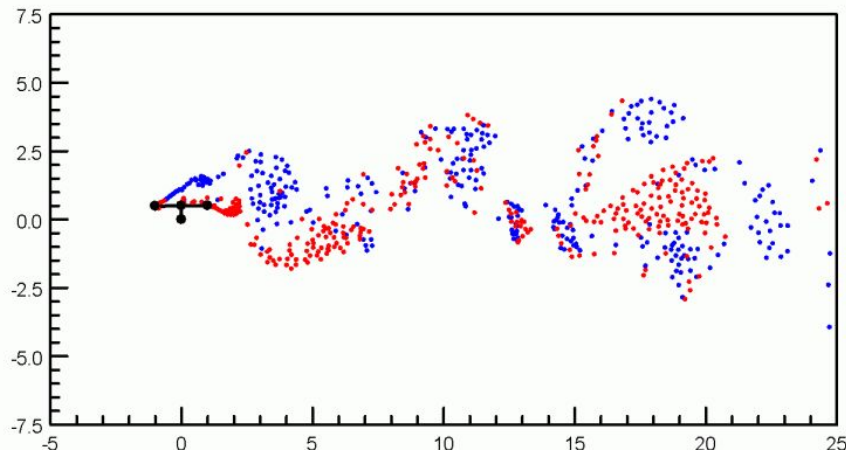
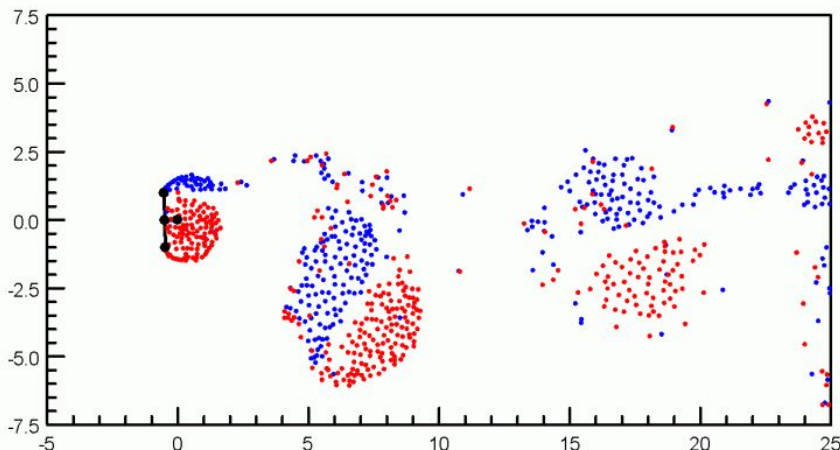
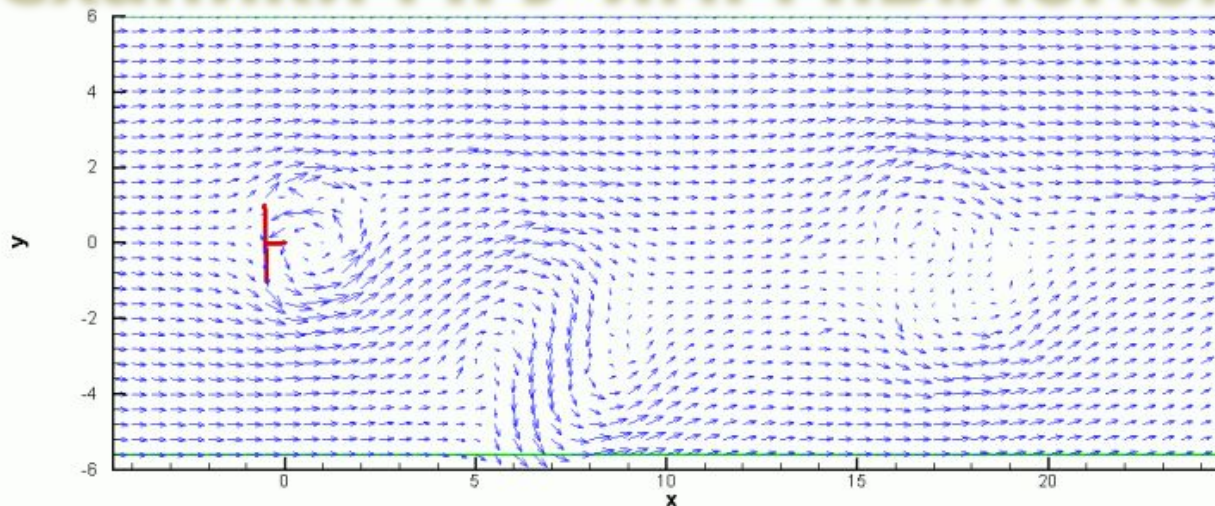
НИИ механики МГУ им. М.В.Ломоносова





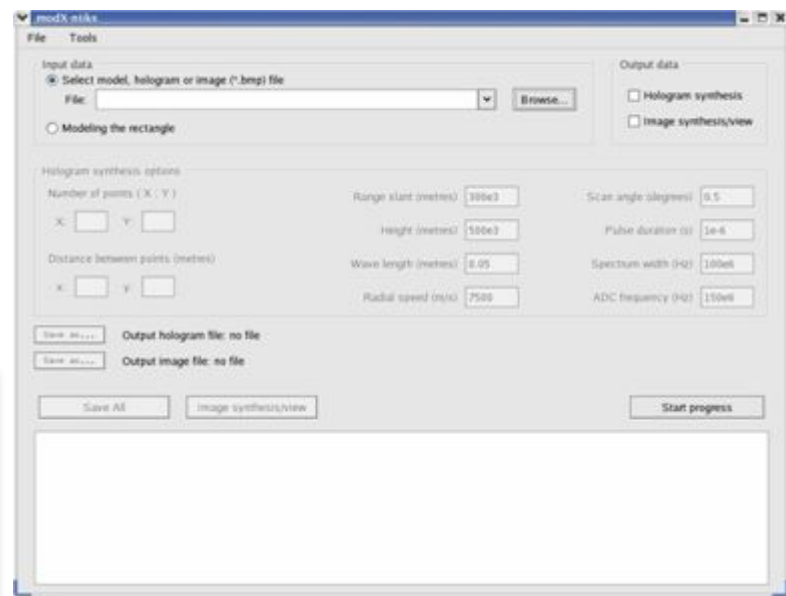
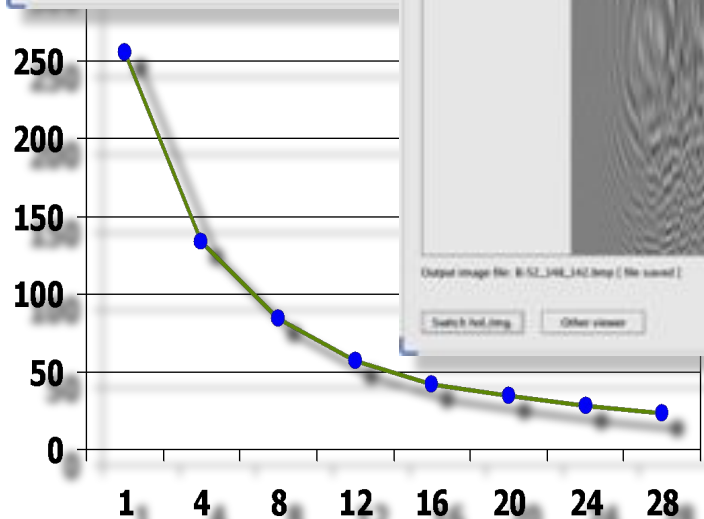
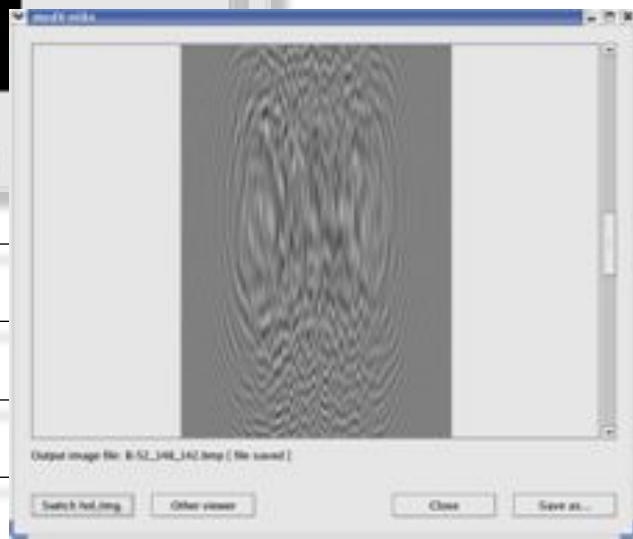
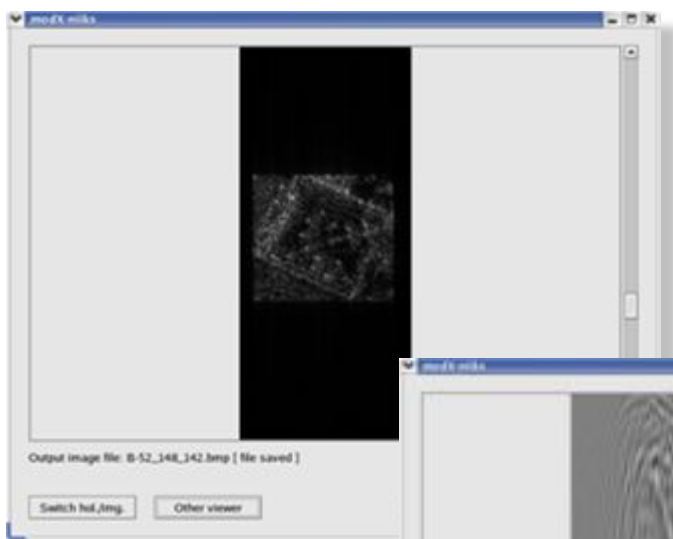
Аэромеханика

НИИ механики МГУ им. М.В.Ломоносова





Open TS: архитектура и реализация Моделирование перспективной широкополосной РЛС



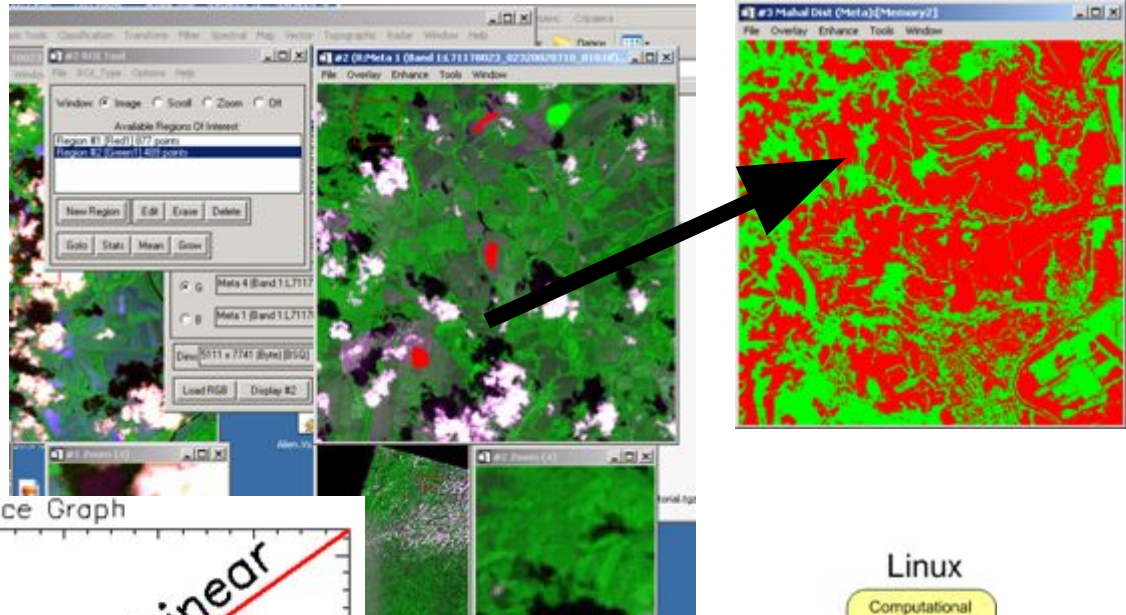
- ❑ Graphical User Interface
- ❑ Non-PSI RAS development team (Space research institute of Khrunichev corp.)



Open TS: архитектура и реализация

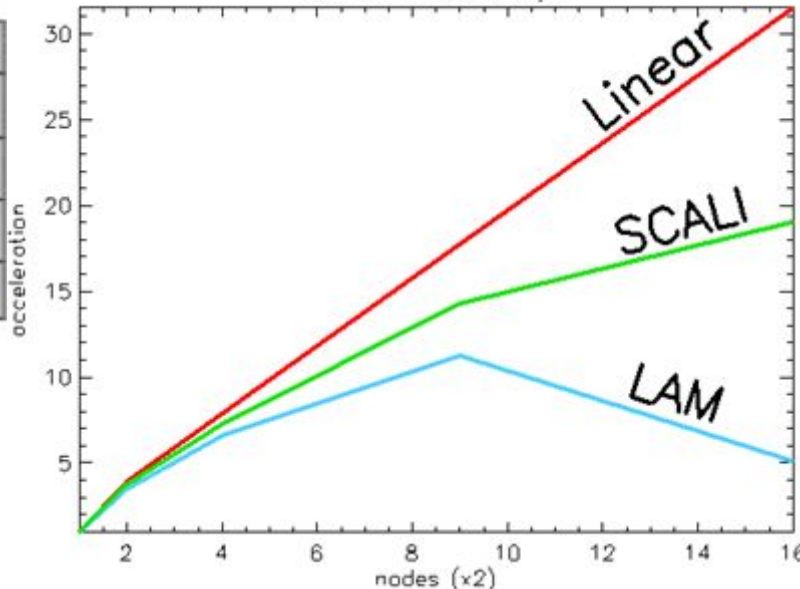
Классификация изображений (Landsat)

Вычислительный Web-сервис

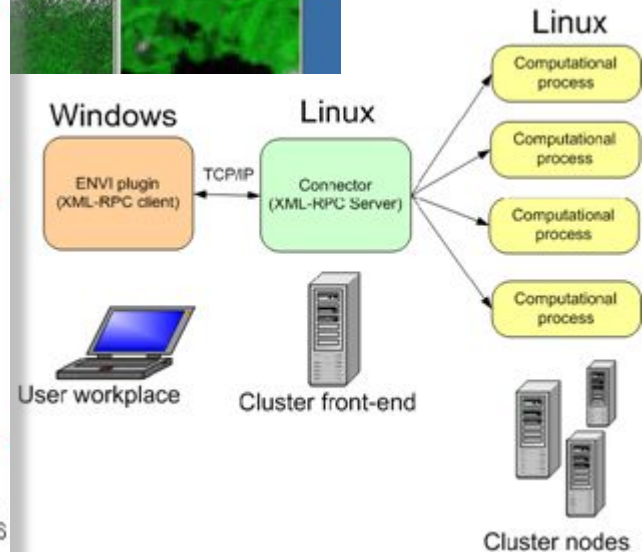


48 regions:
EMVI Mahalanobis Distances:
37 mins = 3420 secs
Mahalanobis Distance at 1 nodes:
313 secs

Performance Graph



	LAM	SCALI
2	256	239
4	136	124
9	80	63
16	176	47





Дальнейшие планы

- ❑ Более глубокая поддержка многоядерных CPU
- ❑ (Территориально-) Распределенные системы
 - ★ Планировщик
 - ★ Другие коммуникационные реализации DMPI
 - ★ Интерфейсы к Web-сервисам
- ❑ Fault-tolerance
- ❑ Оптимизация под различные современные CPU
- ❑ Скелеты алгоритмов, шаблоны и параллельные библиотеки высокого уровня:
 - ★ `sum` = `fold +`
 - ★ `minimum` = `fold min`
 - ★ `prod` = `fold *`



За рамками доклада

- Другие T-диалекты: T-Refal, T-Fortan
- Мемоизация (табулирование) функций
- Автоматическое переключение между **call**-стилем и **fork**-стилем при вызове T-функций
- Checkpointing
- Heartbeat-механизм
- Ароматы (Flavours) **tptr**-указателей: “**normal**”, “**glue**” and “**magnetic**” — ленивые, жадные и супержадные передачи данных



Благодарности

- Суперкомпьютерный проект СКИФ
- Программы РАН
 - ★ ОИВТС: «Высокопроизводительные вычислительные системы с новыми принципами организации вычислительных процессов»
 - ★ Президиум: «Создание основы для внедрения распределенных научных информационно-вычислительной среды на GRID технологиях»
- РФФИ: грант 05-07-08005-офи_a
- Microsoft – контракт «Open TS vs MPI case study»



**Институт программных систем
Российской академии наук и К°**

Спасибо за внимание...

... .. Готов ответить на вопросы

