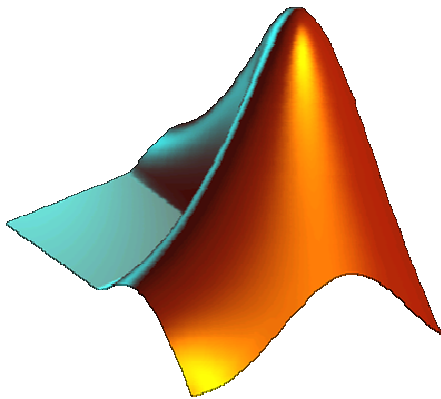


Параллельные вычисления в MATLAB



Аркадий Туревский

Несколько вопросов для рассмотрения

- Хотите ли вы ускорить ваши алгоритмы?
- Работаете ли вы с объемами данных слишком большими для вашего компьютера?

Если так...

- Есть ли к вас многоядерный или многопроцессорный компьютер?
- Есть ли у вас доступ к кластеру?

Решение сложных технических проблем

Проблемы

Вы могли бы...

Решения

Длительное
выполнение

Сложные
вычисления

Ждать



Запуск схожих
заданий на
независимых
процессорах
параллельно

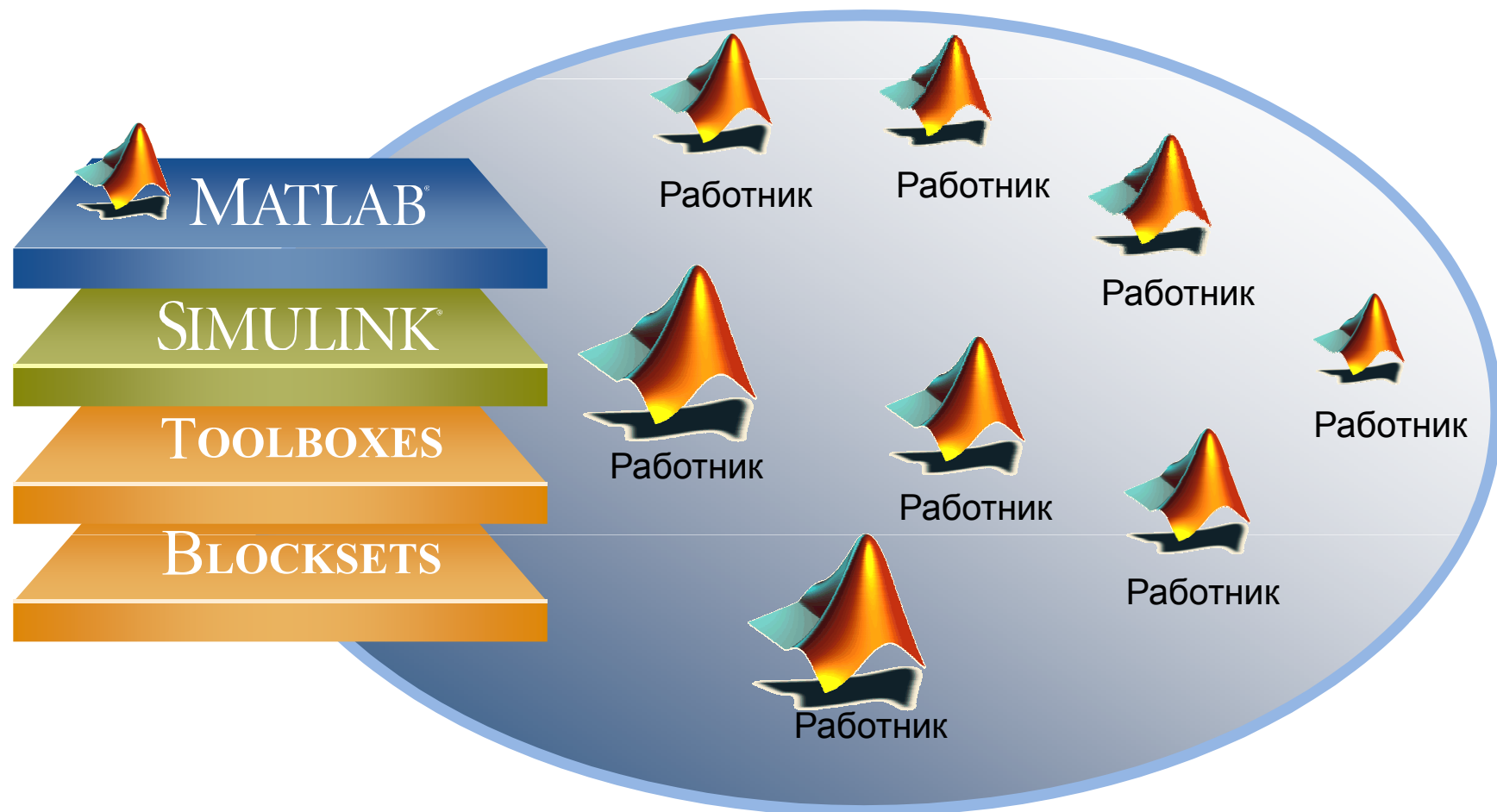
Большие
объемы
данных

Уменьшить
размер задачи



Загрузка *данных*
на нескольких
компьютерах,
которые работают
параллельно

Параллельные вычисления в MATLAB



Программирование параллельных приложений

Уровень управления

Необходимые усилия

Минимальный

Никаких

Небольшой

Непосредственные

Высокий

***Полное
вовлечение***



Программирование параллельных приложений

Уровень управления

Минимальный

Небольшой

Высокий

Необходимые усилия

**Поддержка, встроенная
в toolbox**

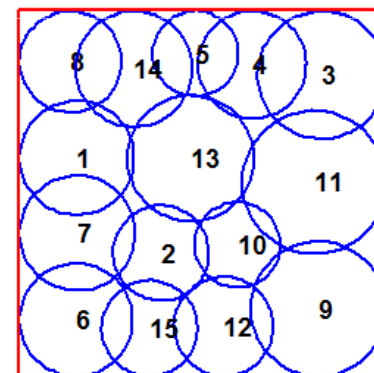
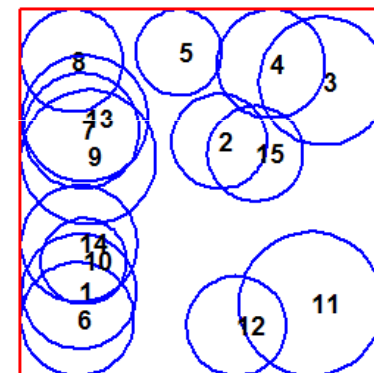
Непосредственное

*Полное
вовлечение*

Демонстрация встроенной поддержки

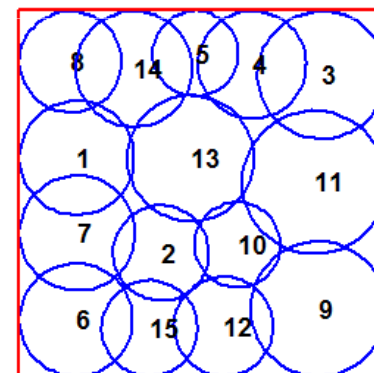
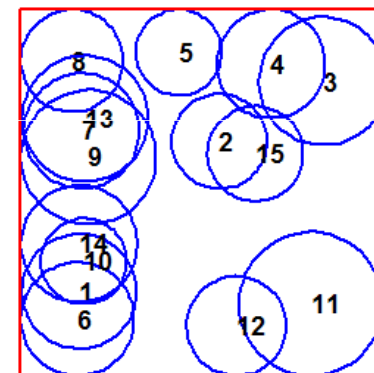
Пример: оптимизация расположения сотовых станций

- Определить расположение
- Максимизировать покрытие
- Минимизировать пересечение



Что мы увидели

- Встроенная поддержка Parallel Computing Toolbox в Optimization Toolbox
- Использование MATLAB работников
- Параллельная оптимизация при помощи **fmincon**

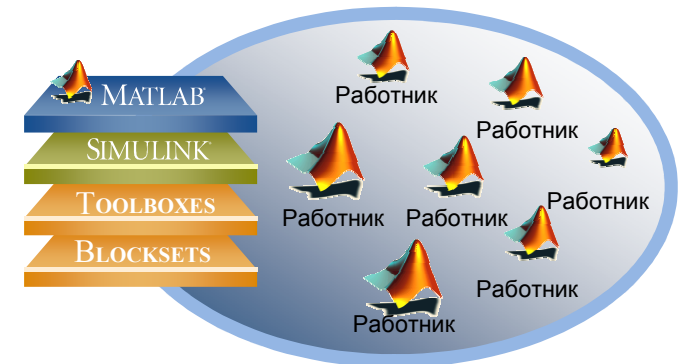


Поддержка параллельных вычислений в Optimization Toolbox

- Функции:
 - **fmincon**
 - Находит ограниченный минимум функции нескольких переменных
 - **fminimax**
 - Находит минимакс решение функции нескольких переменных
 - **fgoalattain**
 - Решает многоцелевую оптимизационную задачу
- Параллелизация вычислений градиентов

Продукты с встроенной поддержкой

- Optimization Toolbox
- Global Optimization Toolbox
- Statistics Toolbox
- SystemTest
- Simulink Design Optimization
- Bioinformatics Toolbox
- Model-Based Calibration Toolbox
- Communications Toolbox



Содержит функции, которые напрямую используют функции Parallel Computing Toolbox

Программирование параллельных приложений

Уровень управления

Минимальный

Небольшой

Высокий

Необходимые усилия

Никаких

Высокоуровневые
конструкции
программирования
(например: `parfor`, `batch`,
`distributed`)

Involved

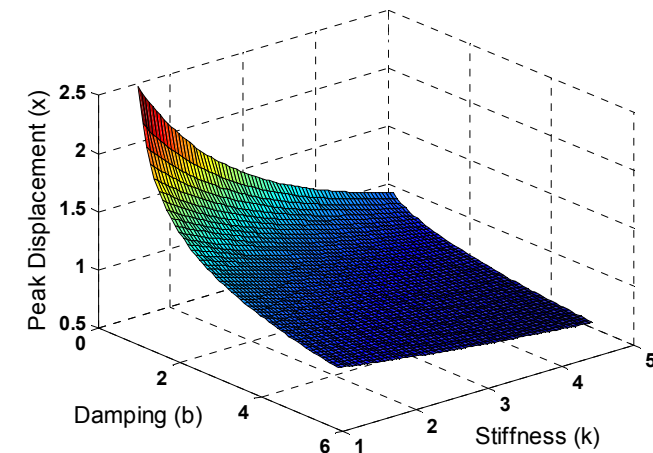
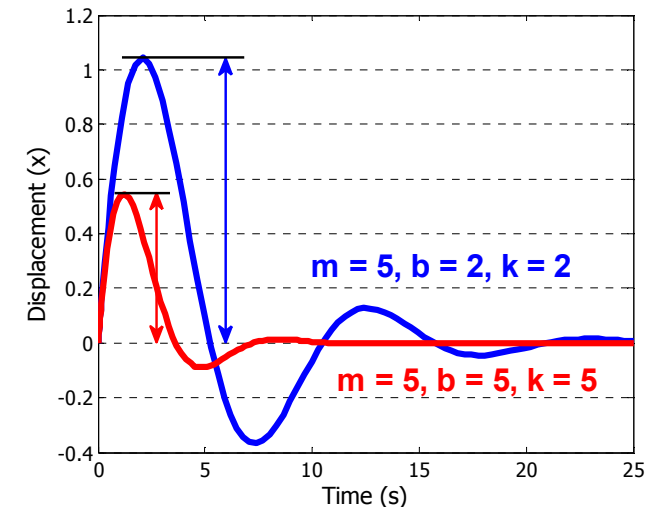
Демонстрация Parfor

Пример: прогон параметров дифференциального уравнения

- Решить дифур 2 порядка

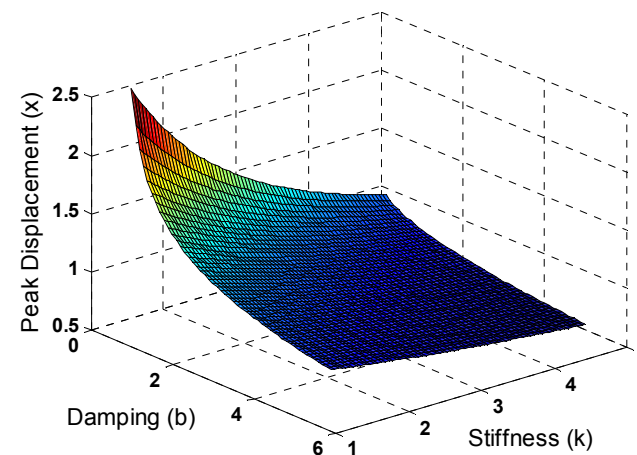
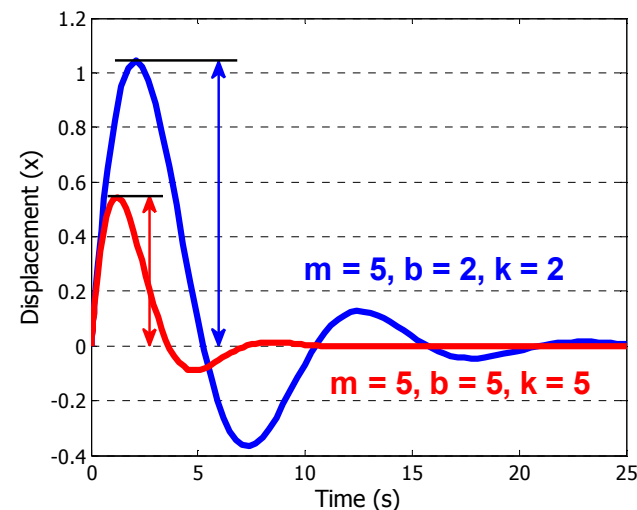
$$\underbrace{m}_{5} \ddot{x} + \underbrace{b}_{1,2,\dots} \dot{x} + \underbrace{k}_{1,2,\dots} x = 0$$

- Имитация с различными величинами b и k
- Сохранить максимум для каждого прогона
- Создать график с результатами

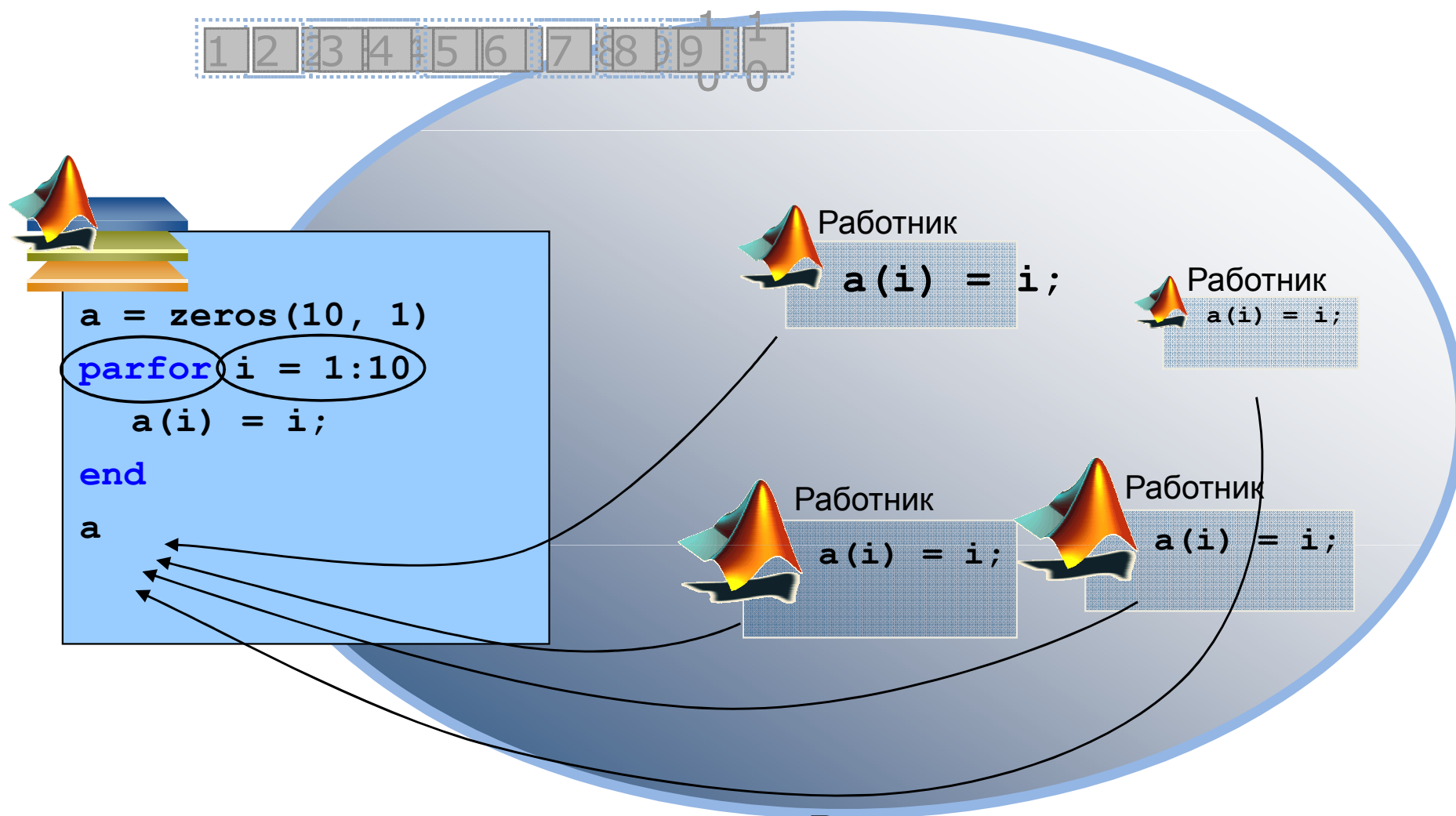


Что мы увидели

- Смешанное параллельное и серийное выполнение кода в одной функции
- Выполнение цикла на работниках MATLAB
- Использование M-Lint для перевода `for`-цикла в `parfor`-цикл



Механика циклов `parfor`



Выделенные ресурсы
для «Работников» MATLAB

Преобразование `for` в `parfor`

- Требования для циклов `parfor`
 - Независимость от задач
 - Независимость от порядка
- Ограничения на тело цикла
 - Не может объявлять переменные (e.g., `eval`, `load`, `global`)
 - Не может содержать выражения `break` или `return`
 - Не может содержать другой цикл `parfor`

Советы по преобразованию `for` в `parfor`

- Используйте M-Lint для диагностики проблем с `parfor`
- Если ваш цикл `for` не может быть преобразован в `parfor`, подумайте о преобразовании подмножества тела цикла в функцию
- Прочтите раздел в документации по классификации переменных
- Обратитесь к блогу:
<http://blogs.mathworks.com/loren/2009/10/02/using-parfor-loops-getting-up-and-running/>

Инструменты параллельных вычислений

Параллельные задачи

Длительные вычисления

- Несколько независимых итераций

```
parfor i = 1 : n
```

```
    % do something with i
```

```
end
```

- Серии задач

Задача 1

Задача 2

Задача 3

Задача 4

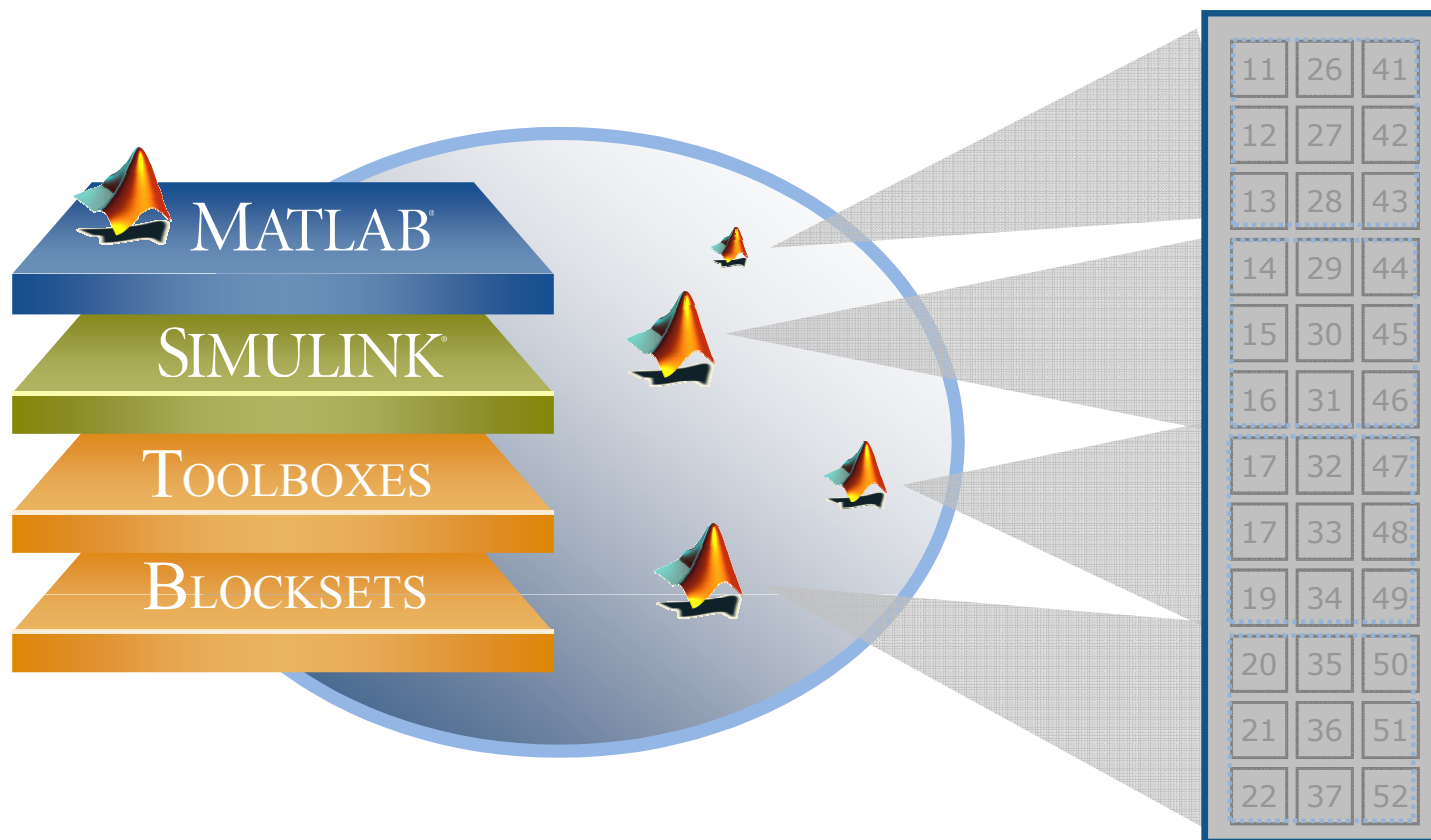
Параллельные данные

Проблема больших объемов данных

1	2	4
1	2	4
1	2	4
1	2	4
1	2	4
1	3	4
1	3	4
1	3	4
1	3	4
1	3	4
1	3	4
2	3	5
2	3	5
2	3	5
2	3	5
2	7	2



Распределенные массивы со стороны клиента



Удаленное управление массивом
с рабочего компьютера

Распределенный массив
находится на кластере

Демонстрация распределенного массива

Расширенные функции MATLAB для работы с распределенными массивами

Type of Function	Function Names
Data functions	cumprod , cumsum , fft , max , min , prod , sum
Data type functions	cast , cell2mat , cell2struct , celldisp , cellfun , char , double , fieldnames , int16 , int32 , int64 , int8 , logical , num2cell , rmfield , single , struct2cell , swapbytes , typecast , uint16 , uint32 , uint64 , uint8
Elementary and trigonometric functions	abs , acos , acosd , acosh , acot , acotd , acoth , acsc , acscd , acsch , angle , asec , asecd , asech , asin , asind , asinh , atan , atan2 , atand , atanh , ceil , complex , conj , cos , cosd , cosh , cot , cotd , coth , csc , cscd , csch , exp , expm1 , fix , floor , hypot , imag , isreal , log , log10 , log1p , log2 , mod , nextpow2 , nthroot , pow2 , real , reallog , realpow , realsqrt , rem , round , sec , secd , sech , sign , sin , sind , sinh , sqrt , tan , tand , tanh
Elementary matrices	cat , diag , eps , find , isempty , isequal , isequalwithhequalnans , isfinite , isinf , isnan , length , ndims , size , tril , triu
Matrix functions	chol , eig , lu , norm , normest , svd
Array operations	all , and , any , bitand , bitor , bitxor , ctranspose , end , eq , ge , gt , horzcat , ldivide , le , lt , minus , mldivide , mrdivide , mtimes , ne , not , or , plus , power , rdivide , subsasgn , subsindex , subsref , times , transpose , uminus , uplus , vertcat , xor
Sparse matrix functions	full , issparse , nnz , nonzeros , nzmax , sparse , spfun , spones
Special functions	dot

Программирование параллельных приложений

Уровень управления

Необходимые усилия

Минимальный

None

Небольшой

Straightforward

Высокий

Конструкции
низкоуровневого
программирования
(например, работа/задача, MPI)

Функции, основанные на MPI в Parallel Computing Toolbox

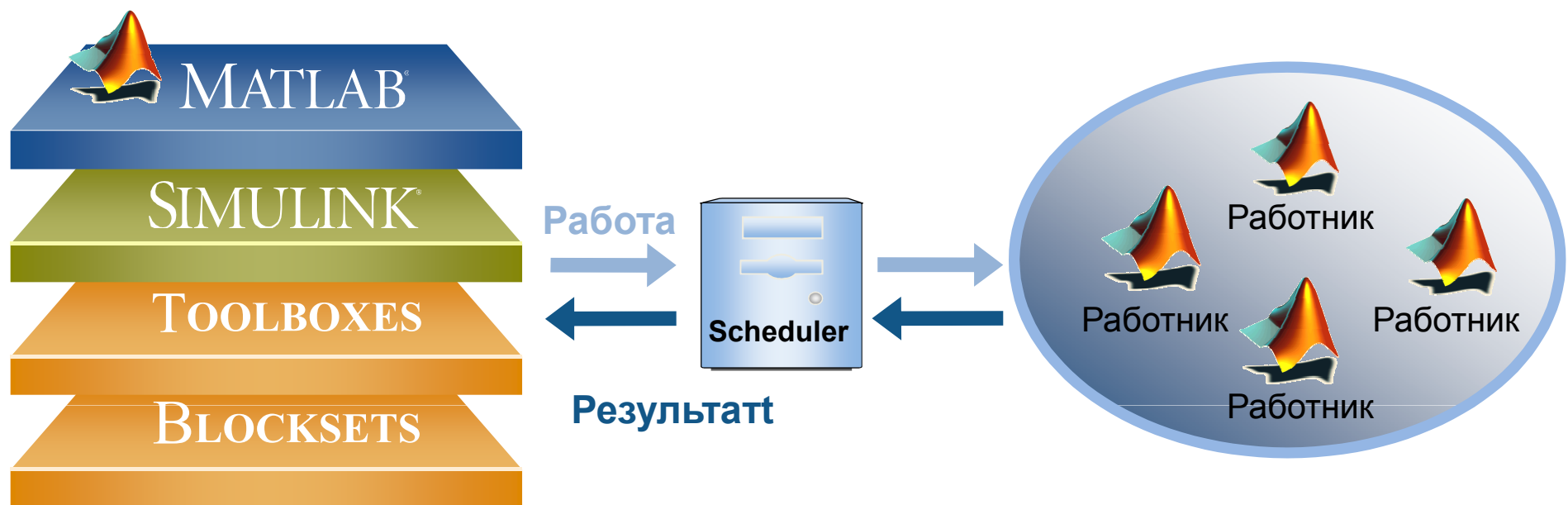
- Используется, когда необходим высокий уровень управления параллельными алгоритмами
- Высокий уровень абстракции от функций MPI
 - `labSendReceive`, `labBroadcast` и другие
 - Передача, прием и трансляция любого типа данных MATLAB
- Автоматический учет использования ресурсов
 - Setup: `communication`, `ranks`, etc.
 - Определение ошибок: тупики и ошибки связи
- Сменные
 - Использование любых реализаций MPI, двоично-совместимых с MPICH2

От интерактивного к планируемому

- Интерактивный
 - Хорошо подходит для прототипирования
 - Немедленный доступ к «работникам» MATLAB

- Планируемый
 - Передает работу другому «работнику» MATLAB (локальному или на кластер)
 - Обеспечивает доступ к большим вычислительным мощностям для повышения производительности
 - Освобождает локальный сеанс MATLAB

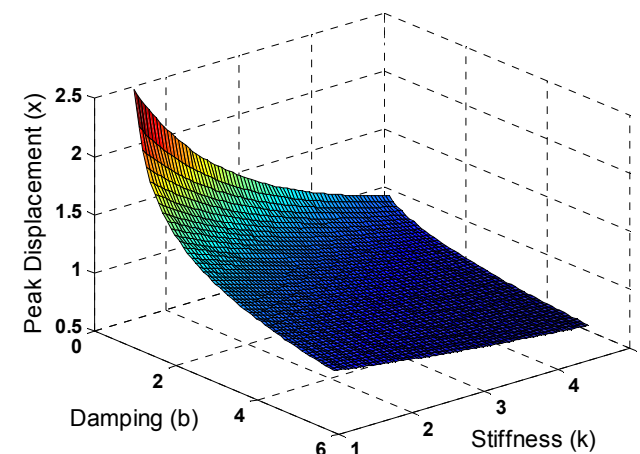
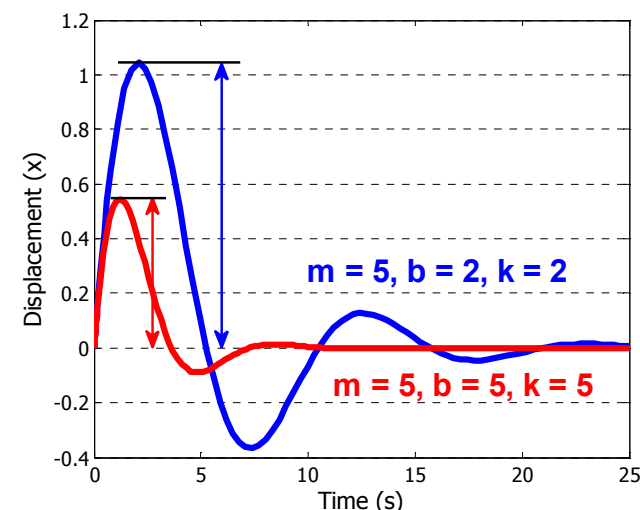
Планировка работы



Демонстрация Batch

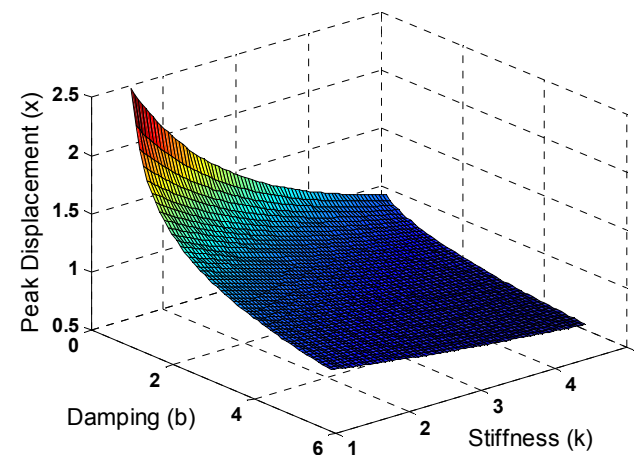
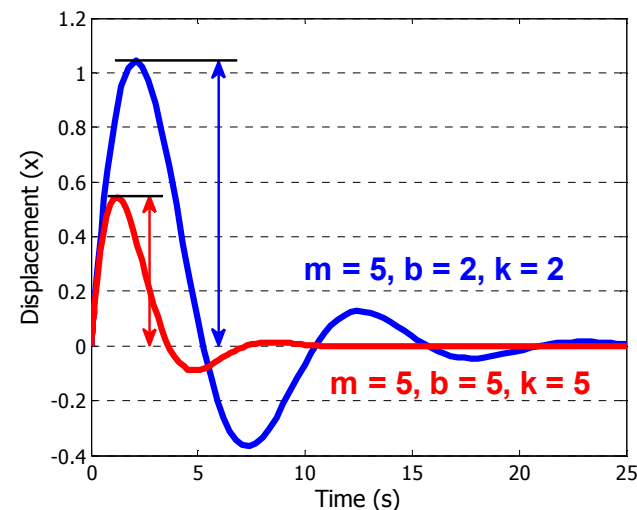
Пример: планирование задания

- Направить прогон на MATLAB работники
- Собрать данные после выполнения расчетов
- Построить график результатов в локальной сессии MATLAB



Что мы увидели

- Использование `batch` для подачи работы
- Использование опции `matlabpool` для подачи работы и параллельного выполнения
- Использование `load` для сбора результатов с работников



Рабочий процесс параллельных задач

- **parfor**

- Множественные независимые итерации
- Легко комбинировать последовательный и параллельный код
- Рабочий процесс
 - Интерактивный, используя **matlabpool**
 - Запланированный, используя **batch**

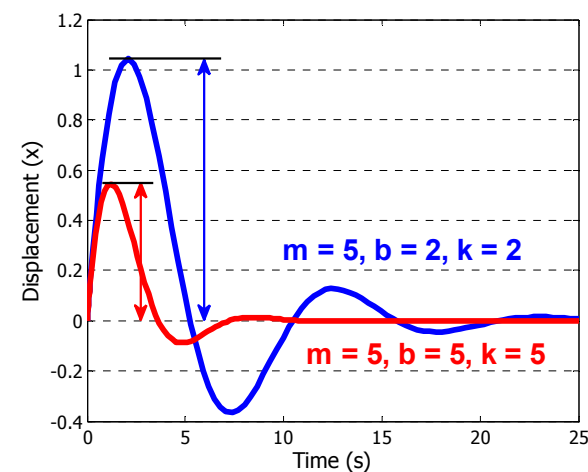
- **createJob** и **createTask**

- Последовательности независимых задач; необязательно итерации
- Рабочий процесс: всегда запланированный

Демонстрация `createJob/createTask`

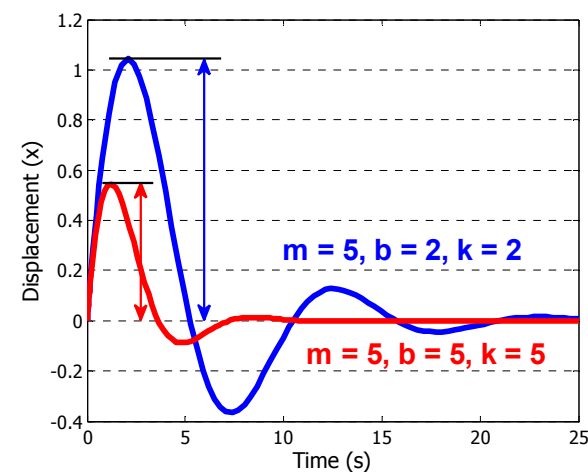
Пример: планирование независимых задач

- Запланировать 3 независимым варианта решения нашего дифуре в предыдущем примере
- Для каждого прогона получить величину перемещения как функцию времени
- Графически сравнить результаты



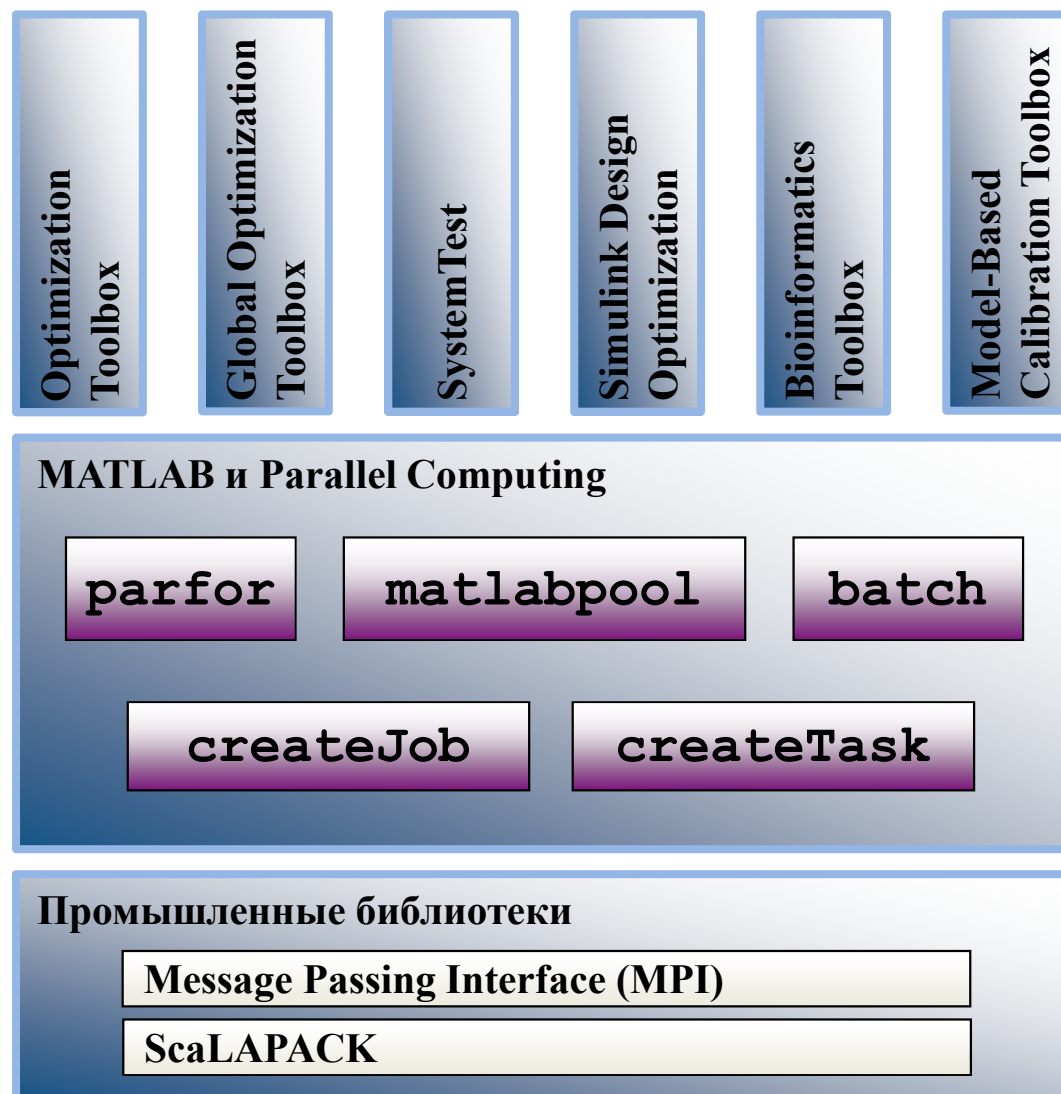
Что мы увидели

- Использование `findResource` чтобы найти scheduler
- Использование `createJob` and `createTask` для создания задачи
- Использование `submit` для подачи работы и параллельного выполнения
- Использование `getAllOutputArguments` для собирания результатов

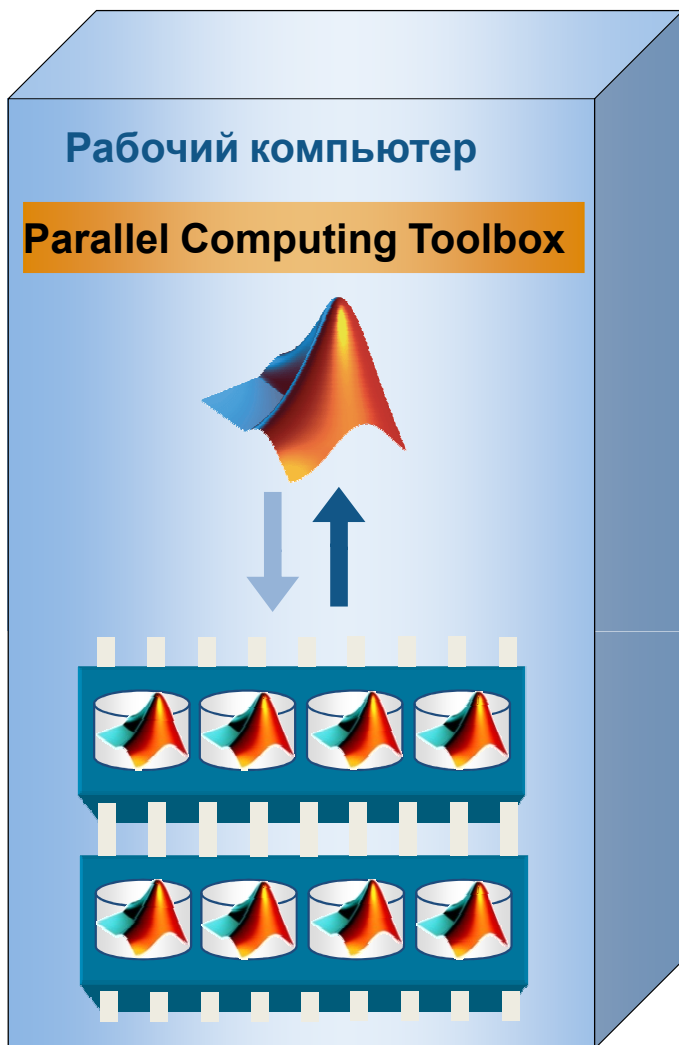


Параллельные вычисления в MATLAB

- Встроенная parallel функциональность распараллеливания в определенных пакетах (необходим Parallel Computing Toolbox)
- Функции высокоуровневого распараллеливания
- Функции низкоуровневого распараллеливания
- Встроенные библиотеки промышленных стандартов

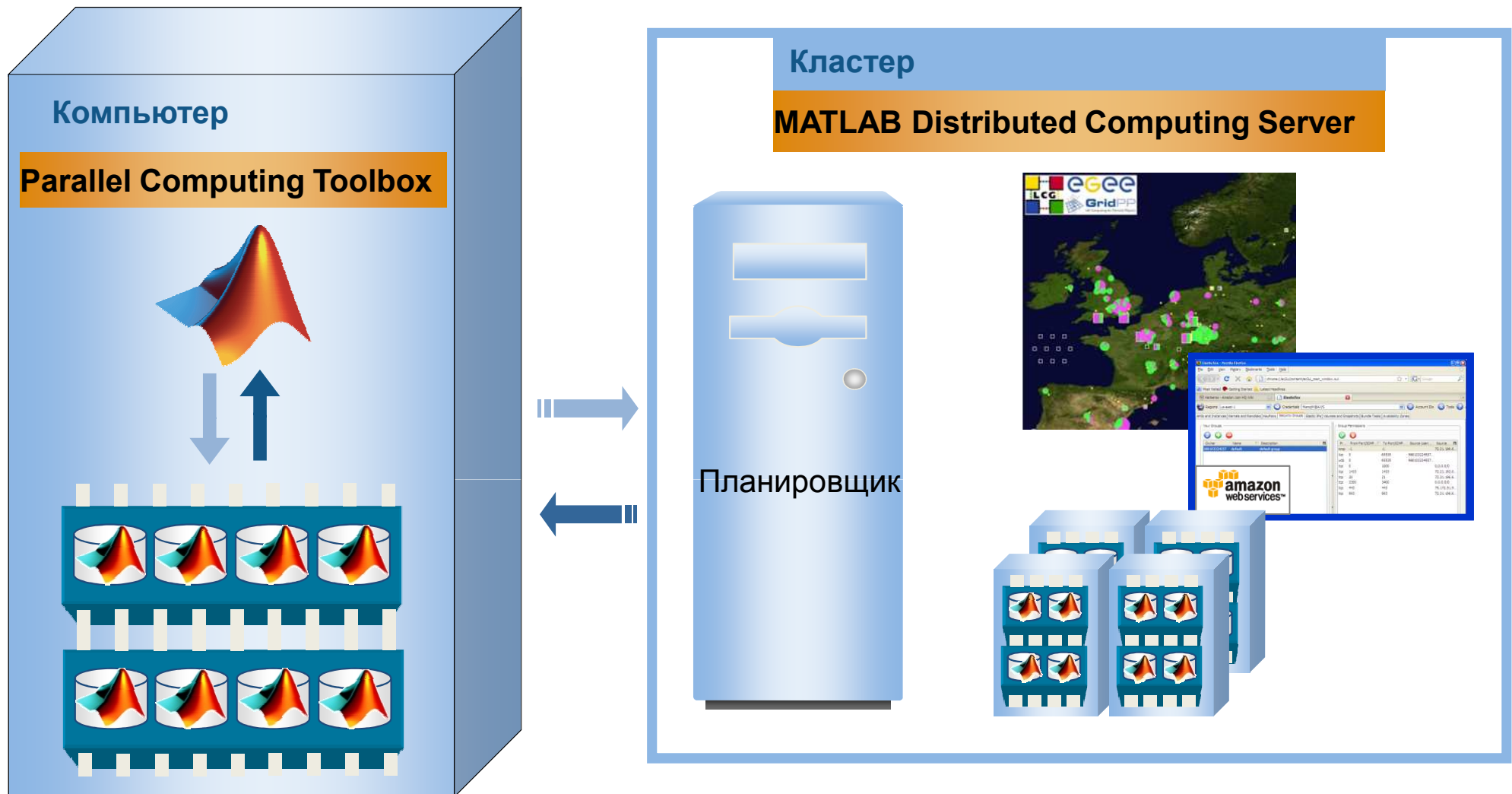


Запуск восьми локальных «работников» на одном компьютере



- Быстрая разработка параллельных приложений на локальном компьютере
- Все преимущества мощности компьютера
- Нет необходимости в отдельном кластере

Расширение до кластеров и сетей



Поддержка планировщиков

Direct Support



Platform™



TORQUE

Open API for others



Программирование параллельных приложений

Уровень управления

Минимальный

Небольшой

Высокий

Необходимые усилия

Поддержка
встроенная в
toolbox

Высокоуровневое
программирование
(например: `parfor`, `batch`,
`distributed`)

Низкоуровневое
программирование
(например: `jobs/tasks`, `MPI`)

