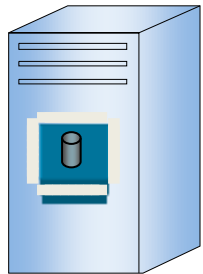


Преимущества распараллеливания на CPU и GPU в MATLAB

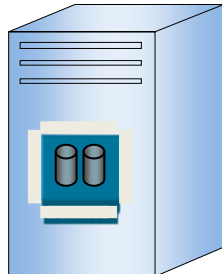
Михаил Шпак

Инженер MathWorks

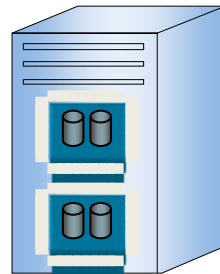
Доступные высокопроизводительные устройства



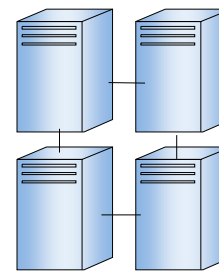
**Один
процессор**



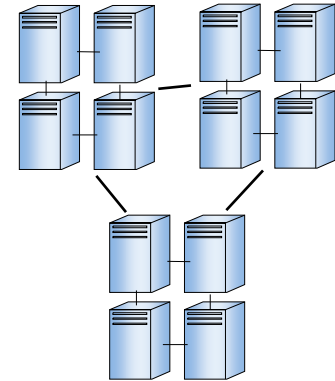
**Много
ядер**



**Много
процессоров**



Кластер

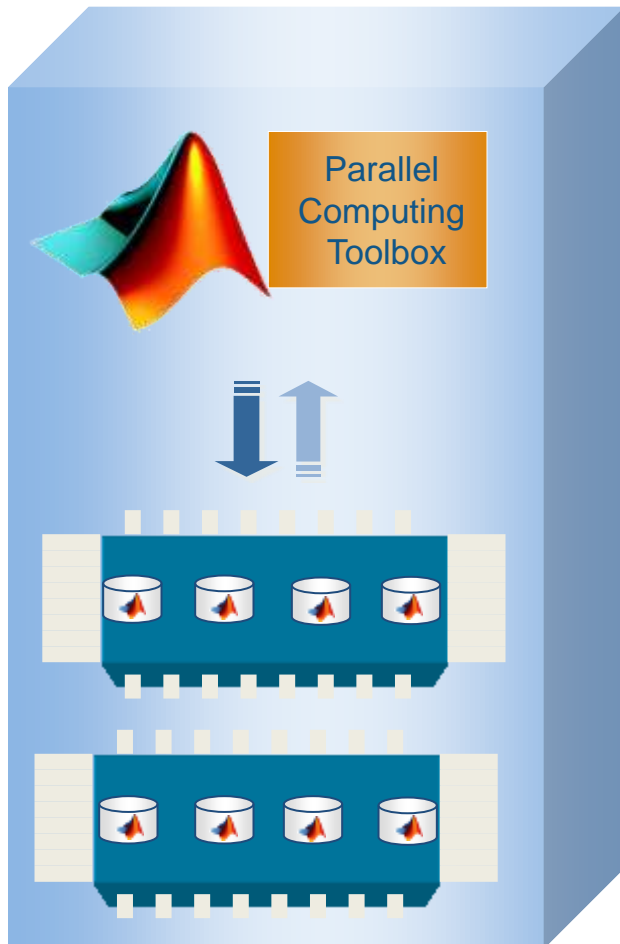


**Грид,
Облако**



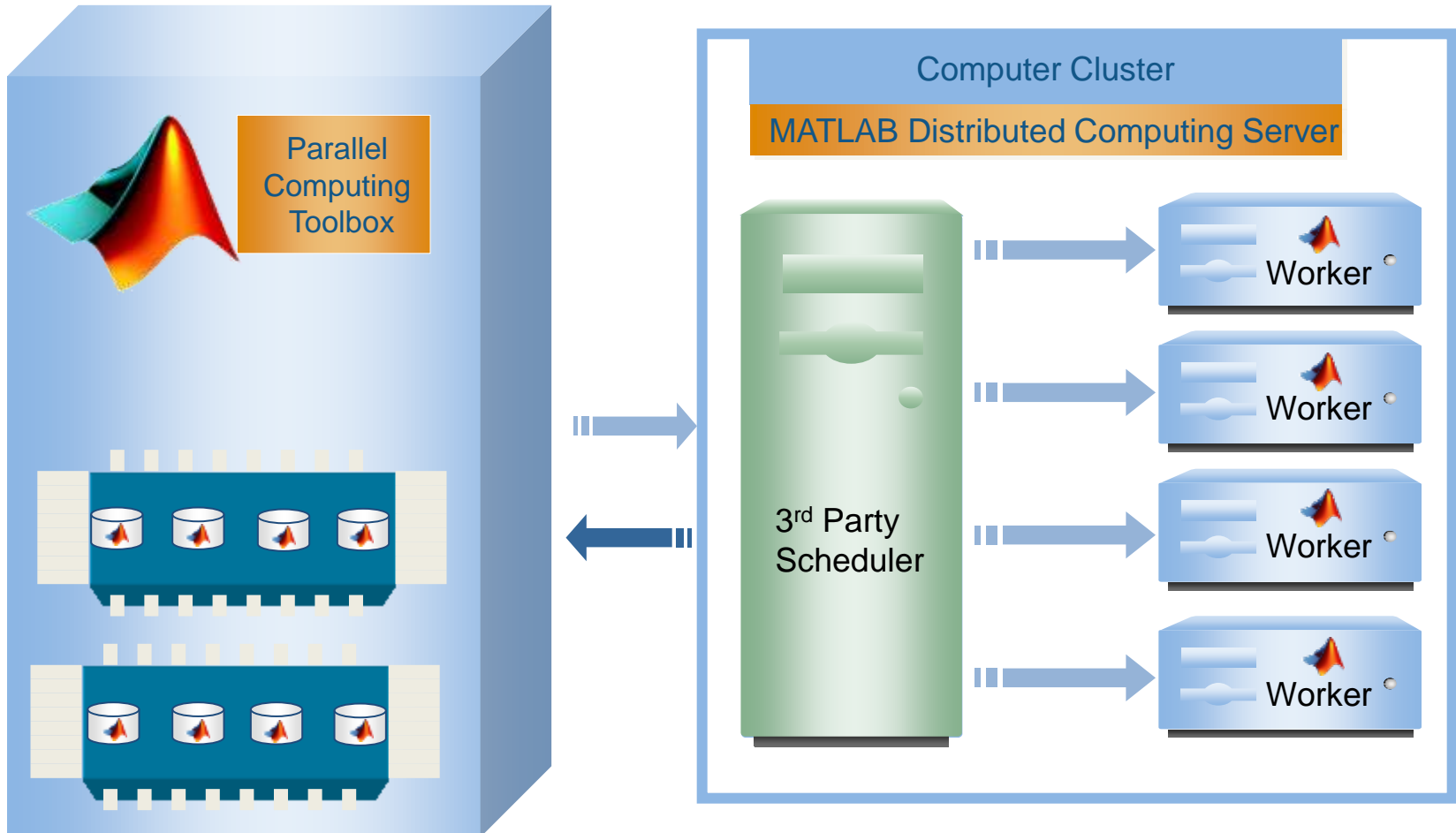
Графический процессор

Parallel Computing Toolbox



- Легко попробовать параллелизм на многоядерных компьютерах
- Быстро разрабатывать параллельные приложения на локальном компьютере
- Воспользоваться всеми преимуществами настольного ПК
- Отдельный расчетный кластер не требуется

Масштабирование на кластер без изменения кода



Создание параллельных приложений

Уровень контроля

Требуемые усилия

Минимальный

**Никаких
усилий**

Поддержка встроена в
тулбоксы

Средний

Средние

Высокоуровневые
конструкции:
(напр. parfor, batch,
distributed)

Обширный

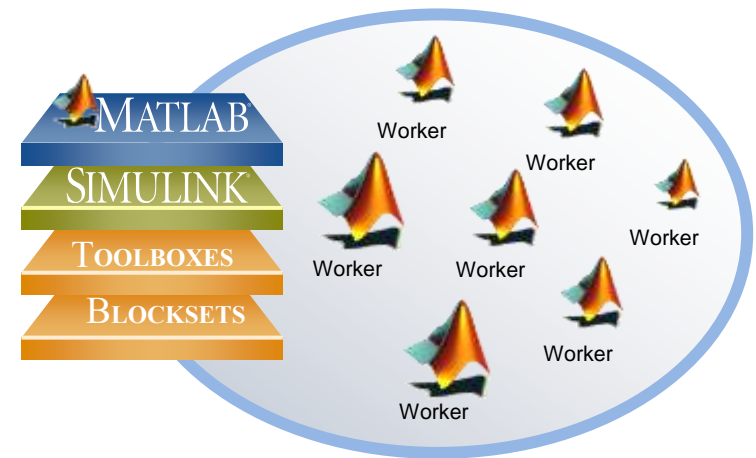
Значительные

Низкоуровневые
конструкции:
(напр. Jobs/Tasks, MPI-based)

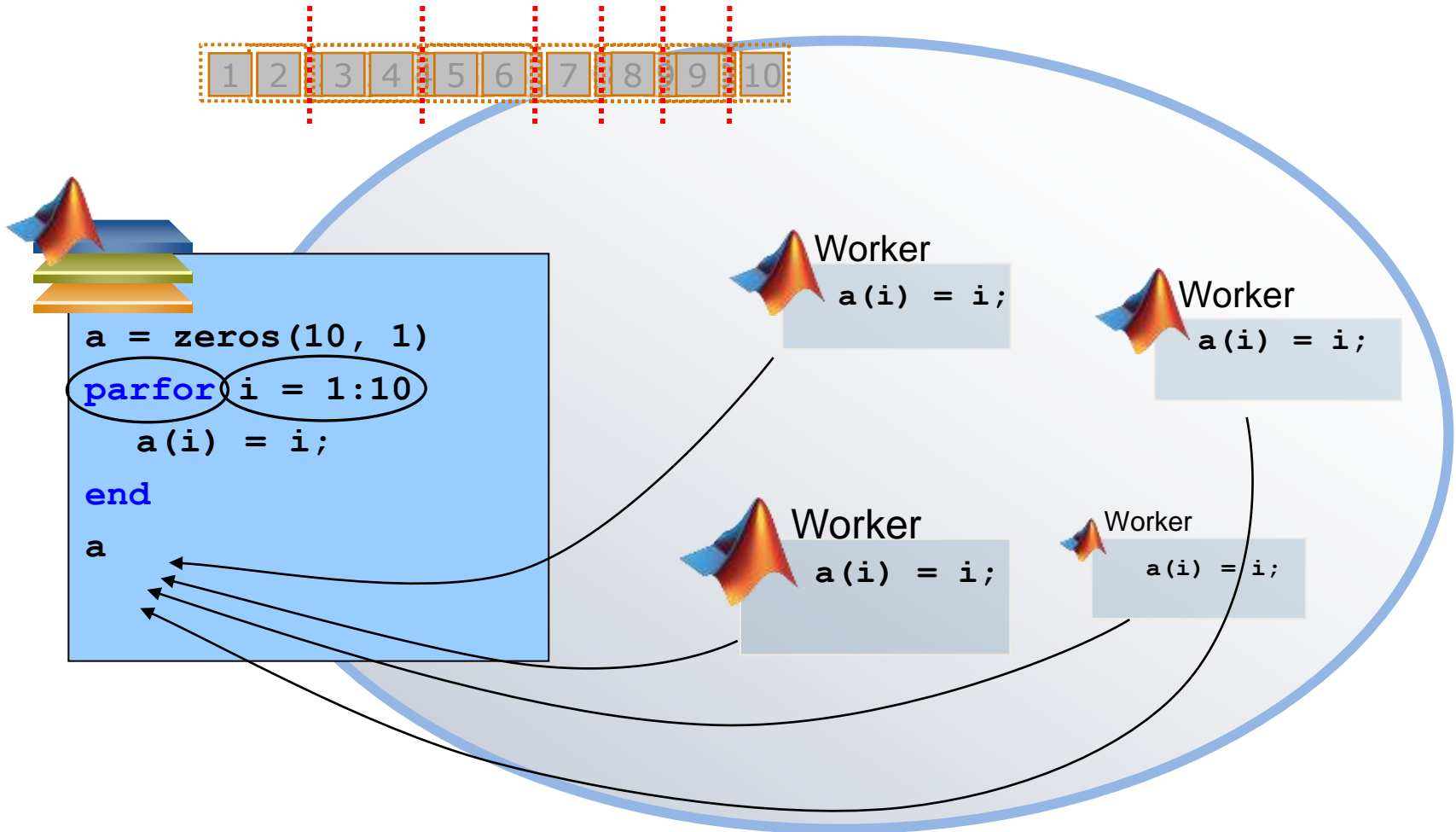


Расширения MATLAB и Simulink со встроенной поддержкой для параллельных вычислений

- Optimization Toolbox
- Global Optimization Toolbox
- Statistics Toolbox
- Bioinformatics Toolbox
- Neural Network Toolbox
- Communications System Toolbox
- Model-Based Calibration Toolbox
- Simulink
- Simulink Design Optimization
- Simulink Code \ Embedded Coder
- И многие другие...

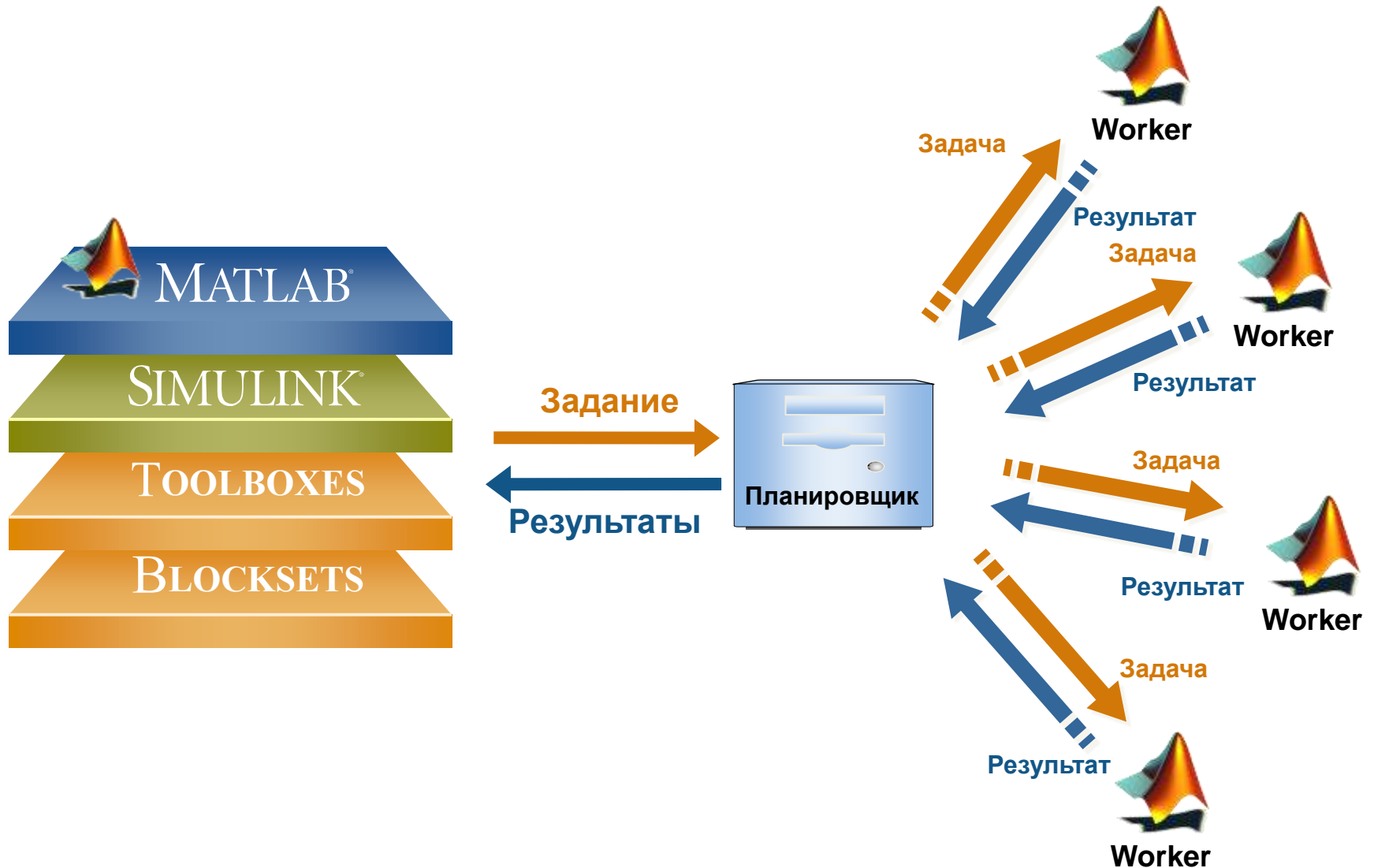


Механизм цикла parfor



Объединение MATLAB Workers

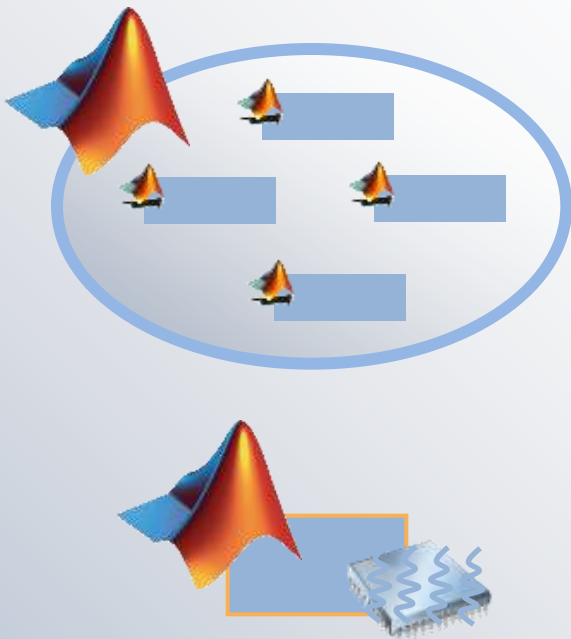
Использование планировщика



Распараллеливание позволяет

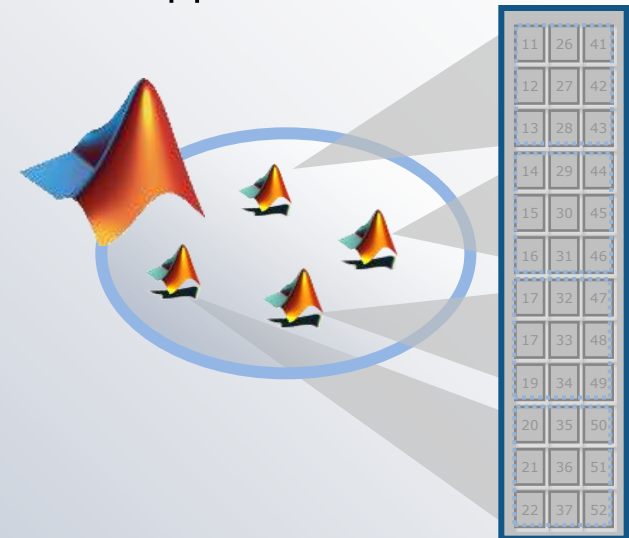
Larger Compute Pool

Ускорить вычисления



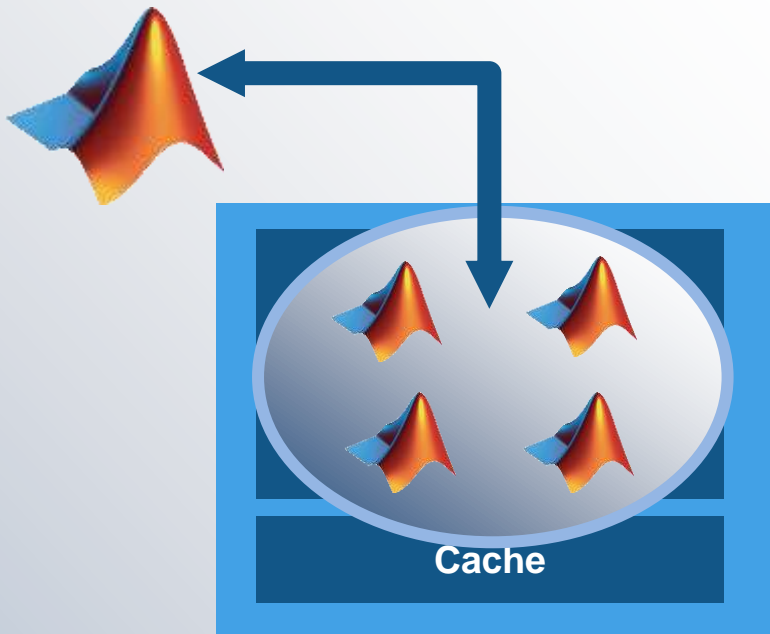
Larger Memory Pool

Работать с большими объемами данных

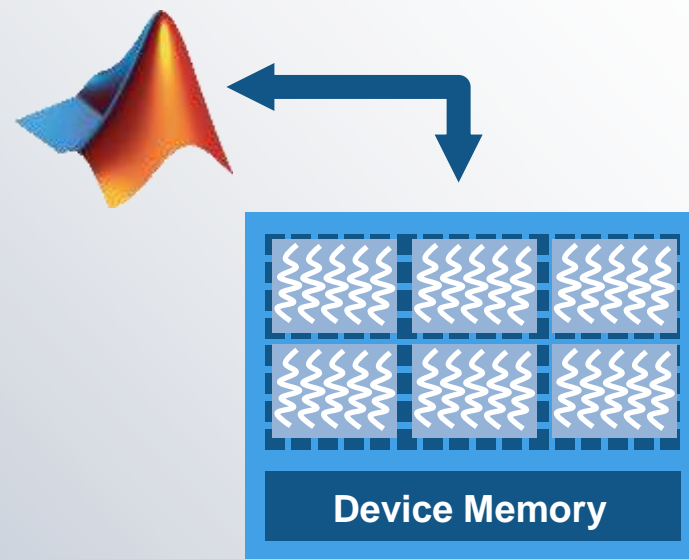


Возможности увеличения производительности

Использование больше ядер (CPUs)



Использование GPUs



Поддержка GPU в Parallel Computing Toolbox

- NVIDIA GPUs с вычислительной способностью 1.3 или выше
 - включая Tesla 10-серий и 20-серий (напр., NVIDIA Tesla C2075 GPU: 448 процессоров, 6 Гб памяти)
 - http://www.nvidia.com/object/cuda_gpus.html
- Почему требуется вычислительная способность 1.3
 - Поддерживает doubles (базовый тип данных в MATLAB)
 - Операции соответствуют стандарту IEEE
 - Поддержка кроссплатформенности

Работа с GPU с версии R2010b



Быстро развивается –
используйте
последнюю версию

Возможности при использовании GPUs

Проще в использовании

Использование интерфейса GPU array со встроенными функциями MATLAB

Запуск пользовательских функций над элементами GPU array

Создание ядер из существующего кода CUDA и PTX файлов

Больше контроля

Пример использования:

GPU массив:

```
>> A = someArray(1000, 1000);
```

```
>> G = gpuArray(A); % Push to GPU memory
```

```
...
```

```
>> F = fft(G);
```

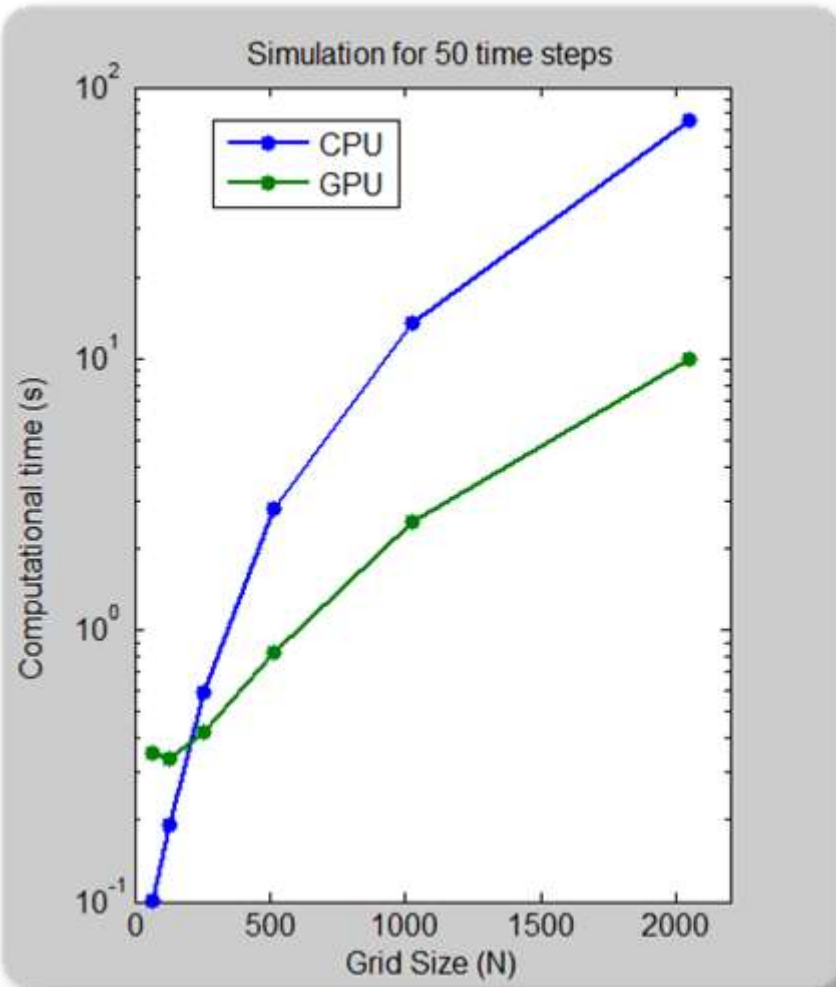
```
>> x = G\b;
```

```
...
```

```
>> z = gather(x); % Bring back into MATLAB
```

Тест: решение 2D волнового уравнения

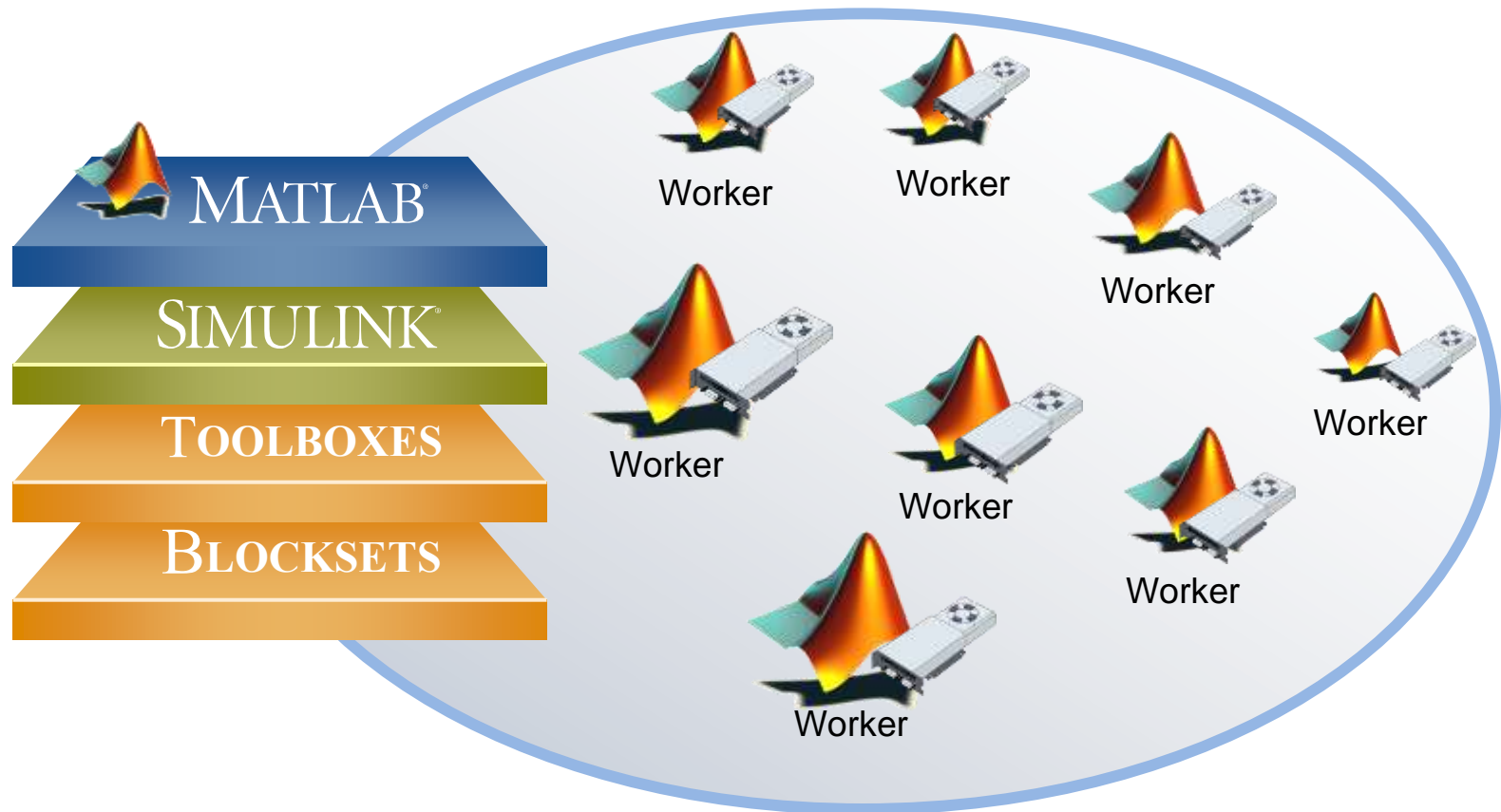
CPU против GPU



Grid Size	CPU (s)	GPU (s)	Ускорение
64 x 64	0.1004	0.3553	0.28
128 x 128	0.1931	0.3368	0.57
256 x 256	0.5888	0.4217	1.4
512 x 512	2.8163	0.8243	3.4
1024 x 1024	13.4797	2.4979	5.4
2048 x 2048	74.9904	9.9567	7.5

Intel Xeon Processor X5650, NVIDIA Tesla C2050 GPU

Масштабирование на несколько GPU



Работа с несколькими GPU

Single GPU

```
N = 1000; % Number of iterations

A = gpuArray(A); % transfer data to GPU

for ix = 1:M
    % Do the GPU-based calculation
    X = myGPUFunction(ix,A);
    % Gather data
    Xtotal(ix,:) = gather(X);
end
```

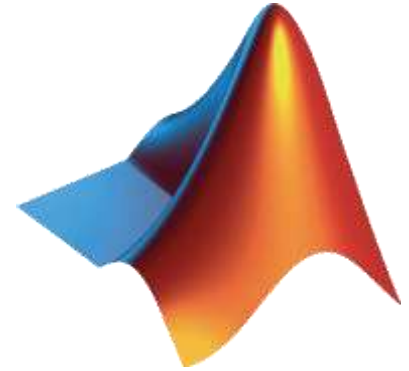
Multiple GPUs

```
N = 1000; % Number of iterations

spmd
    % Assign each worker a different GPU
    gpuDevice(labindex);
    A = gpuArray(A); % transfer data
end

parfor ix = 1:M
    % Do the GPU-based calculation
    X = myGPUFunction(ix,A);
    % Gather data
    Xtotal(ix,:) = gather(X);
end
```


Контактная информация



W: www.matlab.ru

E: matlab@sl-matlab.ru

T: +7 (495) 232 00 23 доб. 0609

Дополнительная информация: www.mathworks.com