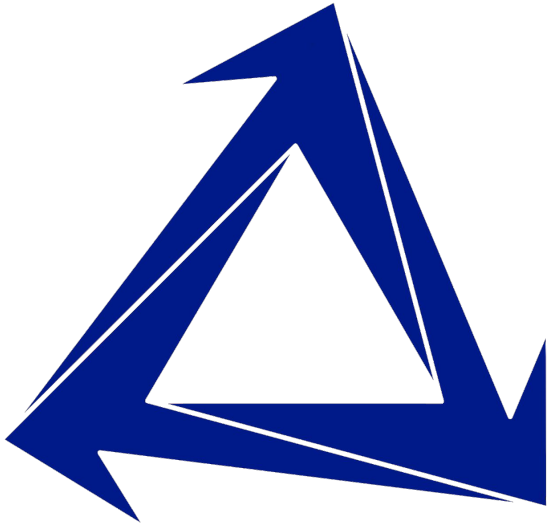
Inspira Crea Transforma

UNIVERSIDAD EAFIT®



APOLO

Centro de computación científica

Universidad EAFIT

Parallel and Concurrent Computing: An introduction

Juan David Pineda-Cárdenas
<jpineda2@eafit.edu.co>

Centro de Computacón Científica APOLO
Universidad EAFIT

Definition

"High performance computing most generally refers to the practice of aggregating computing power in a way that delivers much higher perfomance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering or business".

HPC Inside - www.hpcinside.com

BSC: Supercomputing and Escience

Optimization at many levels

- Hardware Level
  - CPU - Instruction Set - Microinstructions - Pipelining
  - Speed / Amount of Memory
  - Speed / Amount of Storage
  - Speed / Network
- Software Level
  - Algorithm Order Reduction
  - Code Parallelizing
  - Code Distribution
  - Code Optimization

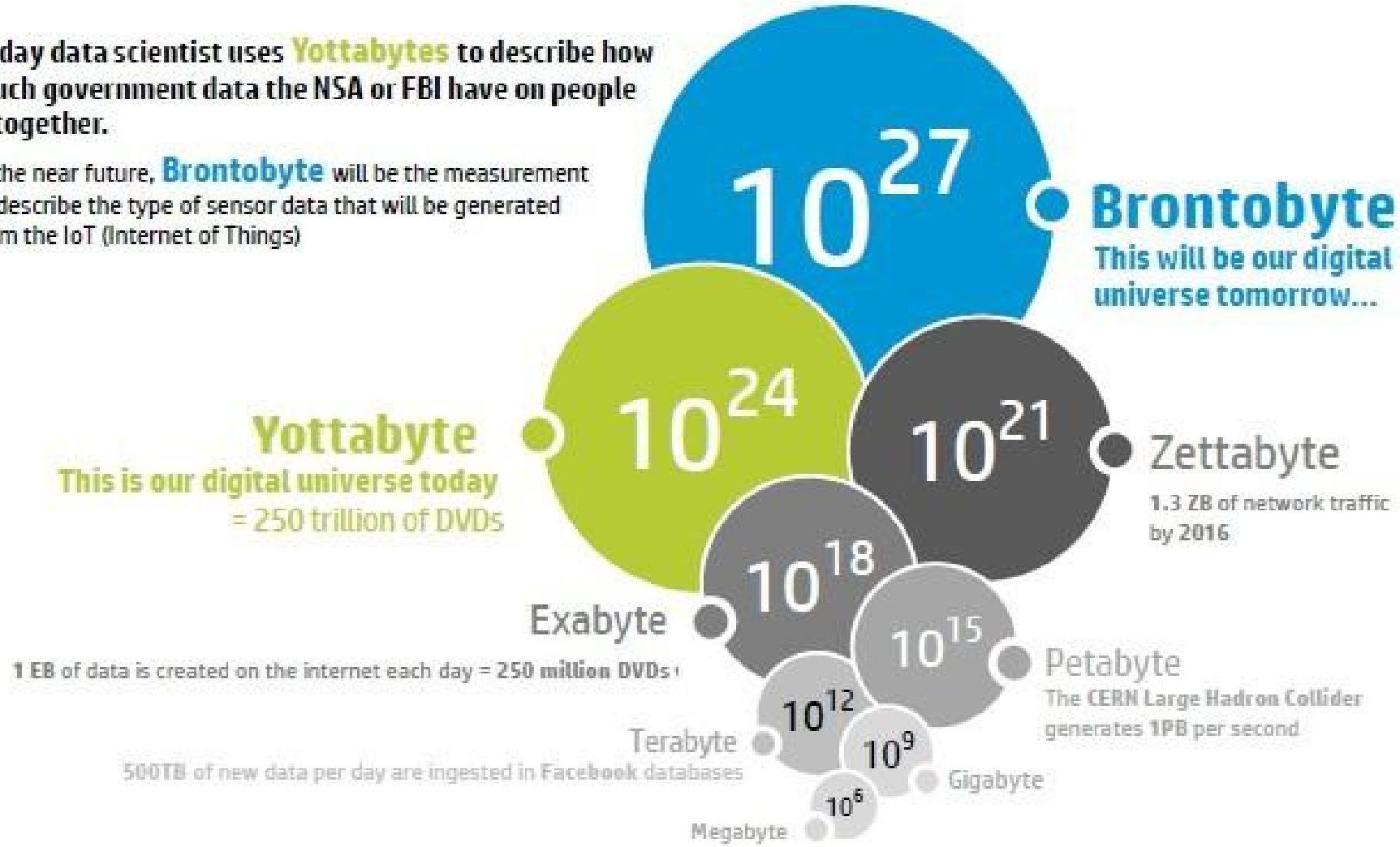
Cheaper
Better
Faster

Linux Prodigy by IBM

https://youtu.be/x7ozaFbqg00

Today data scientist uses Yottabytes to describe how much government data the NSA or FBI have on people altogether.

In the near future, **Brontobyte** will be the measurement to describe the type of sensor data that will be generated from the IoT (Internet of Things)



Taken from: The Register - HP salivates over the future brontobyte digital universe

Intel Slides

The following slides was taken from Intel Academic Courseware: Intro to Parallel Programming. You can find it in:

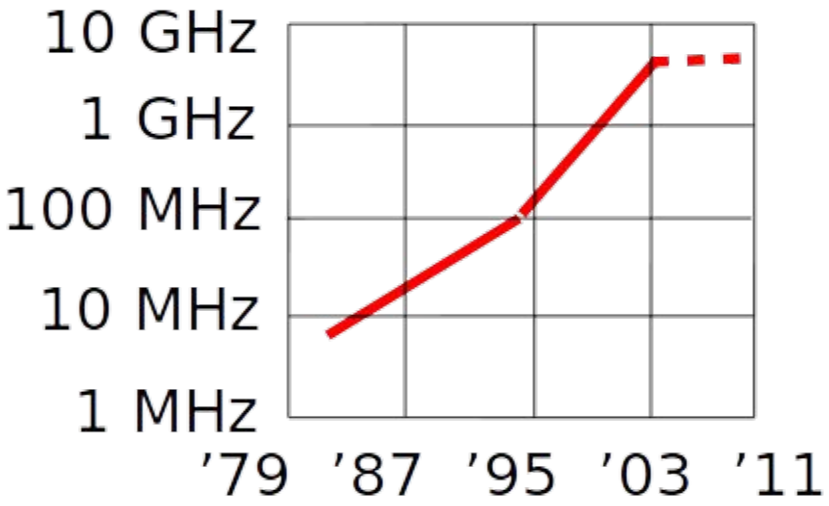
https://software.intel.com/en-us/articles/academic-courseware

Why Parallel Computing?

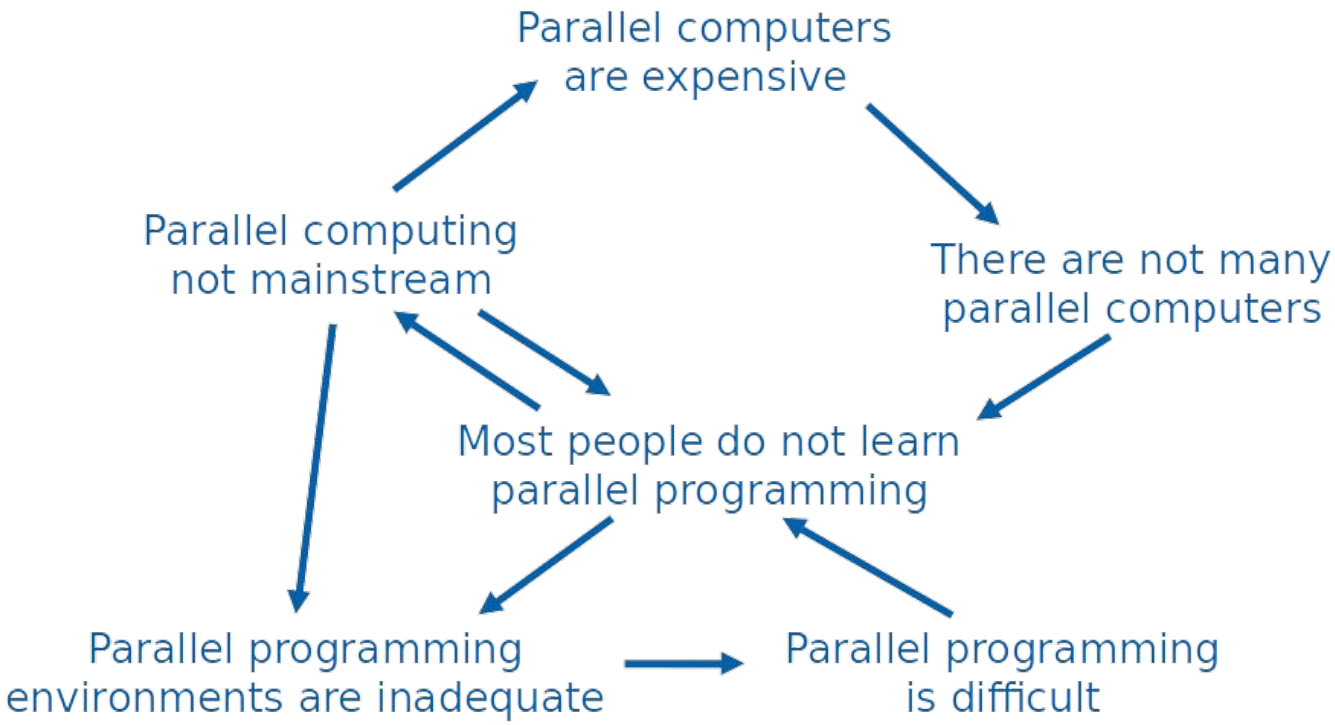
"The free lunch is over." —Herb Sutter

We want applications to execute faster

Clock speeds no longer increasing exponentially



Old Dynamic of Parallel Computing



Sequential Language Approach

- Problem has inherent parallelism
- Programming language cannot express parallelism
- Programmer hides parallelism in sequential constructs
- Compiler and/or hardware must find hidden parallelism
- Sadly, doesn't work

Alternate Approach

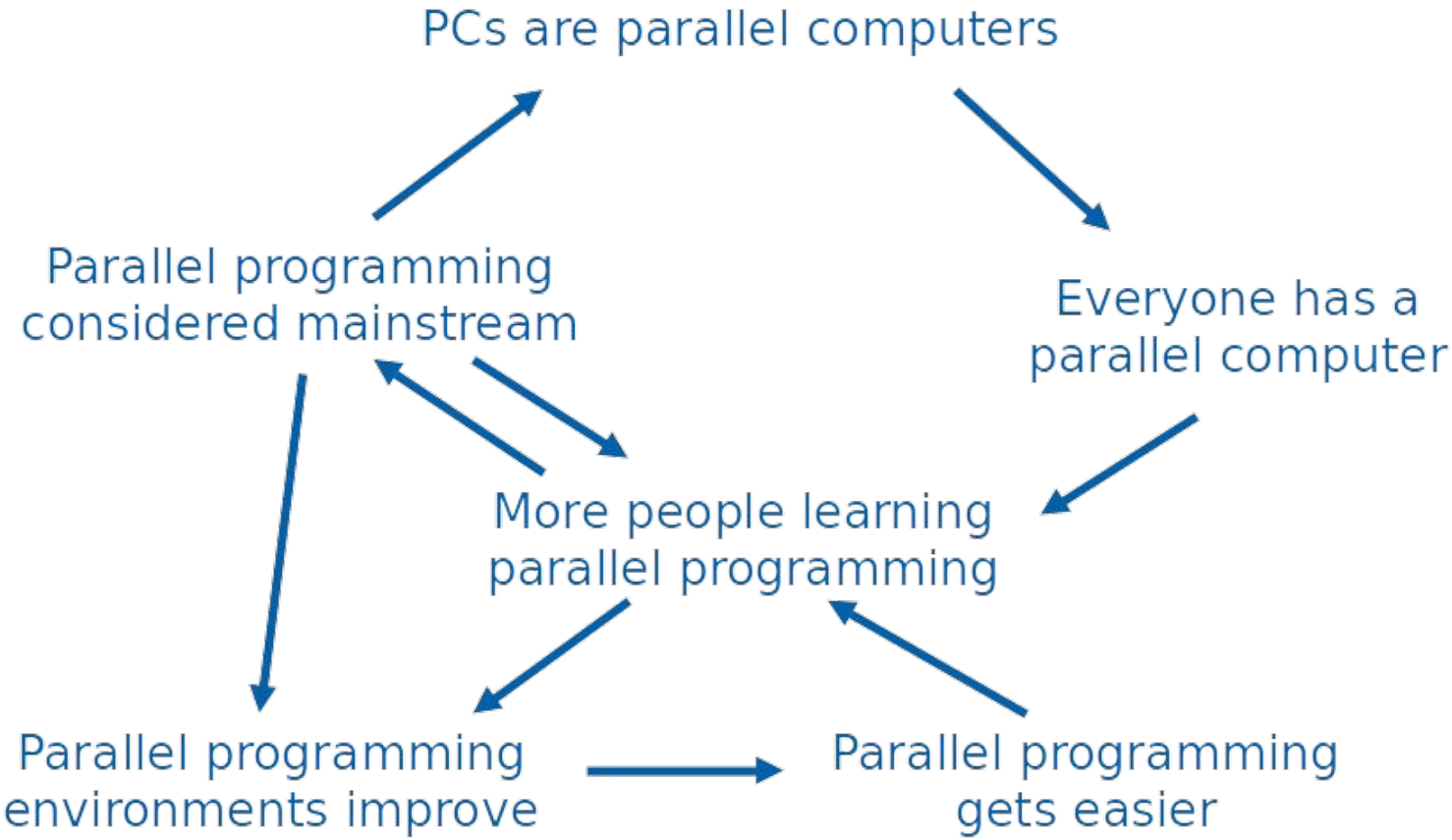
- Programmer and Compiler Work Together
  - Problem has inherent parallelism
  - Programmer has way to express parallelism
  - Compiler translates program for multiple cores

Programmer/Compiler Team

Programmers of modern CPUs must take architecture and compiler into account in order to get peak performance

"…you can actively reorganize data and algorithms to take advantage of architectural capabilities…" Introduction to Microarchitectural Optimization for Itanium® 2 Processors, p. 3

New Dynamic of Parallel Computing



Domain Decomposition

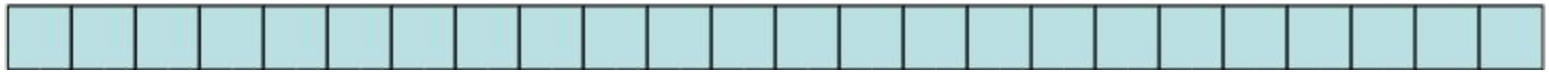
First, decide how data elements should be divided among cores

Second, decide which tasks each core should be doing

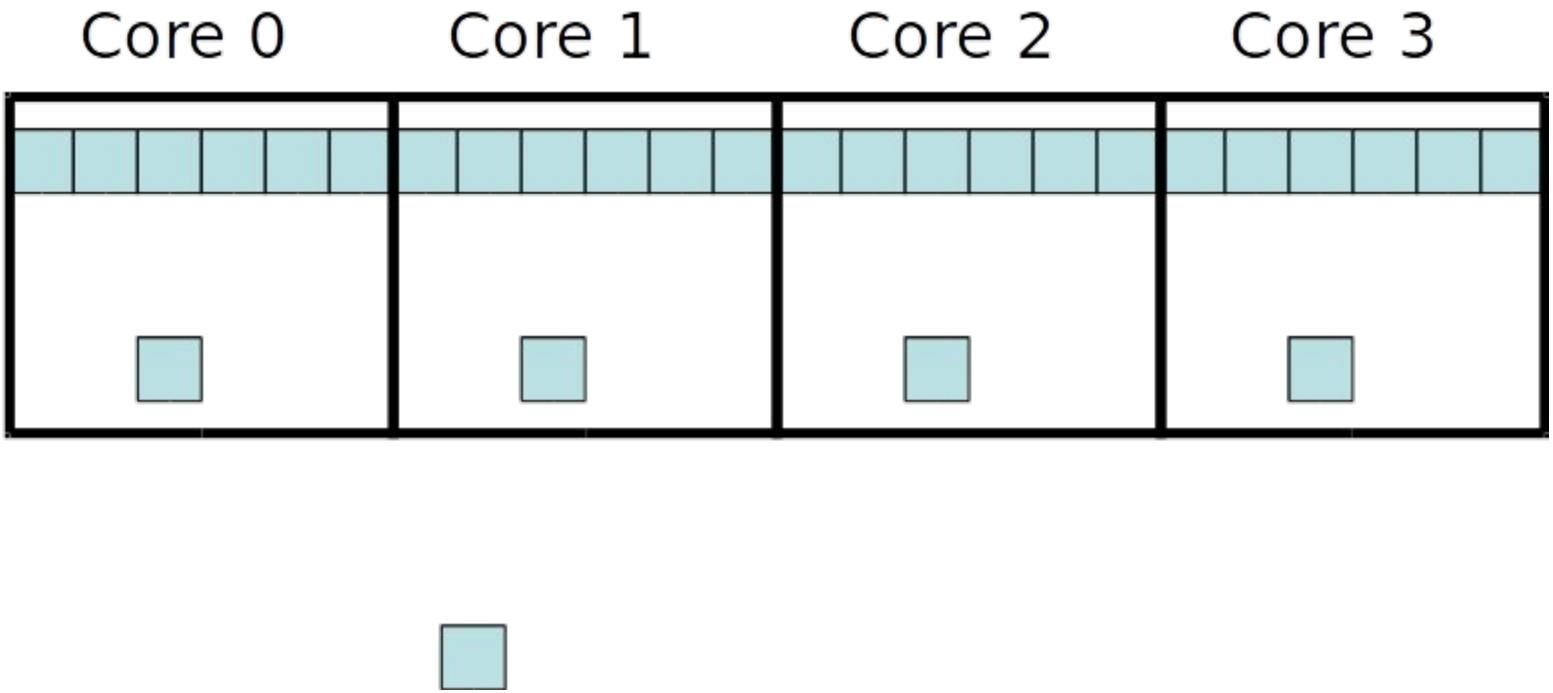
Example: Vector addition

Domain Decomposition

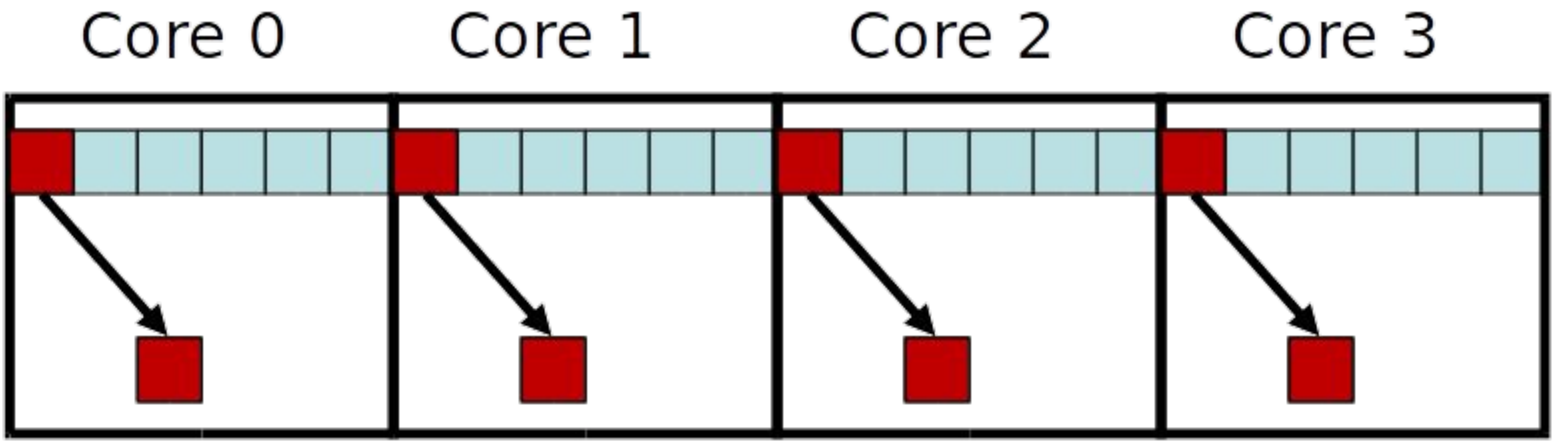
Find the largest element of an array



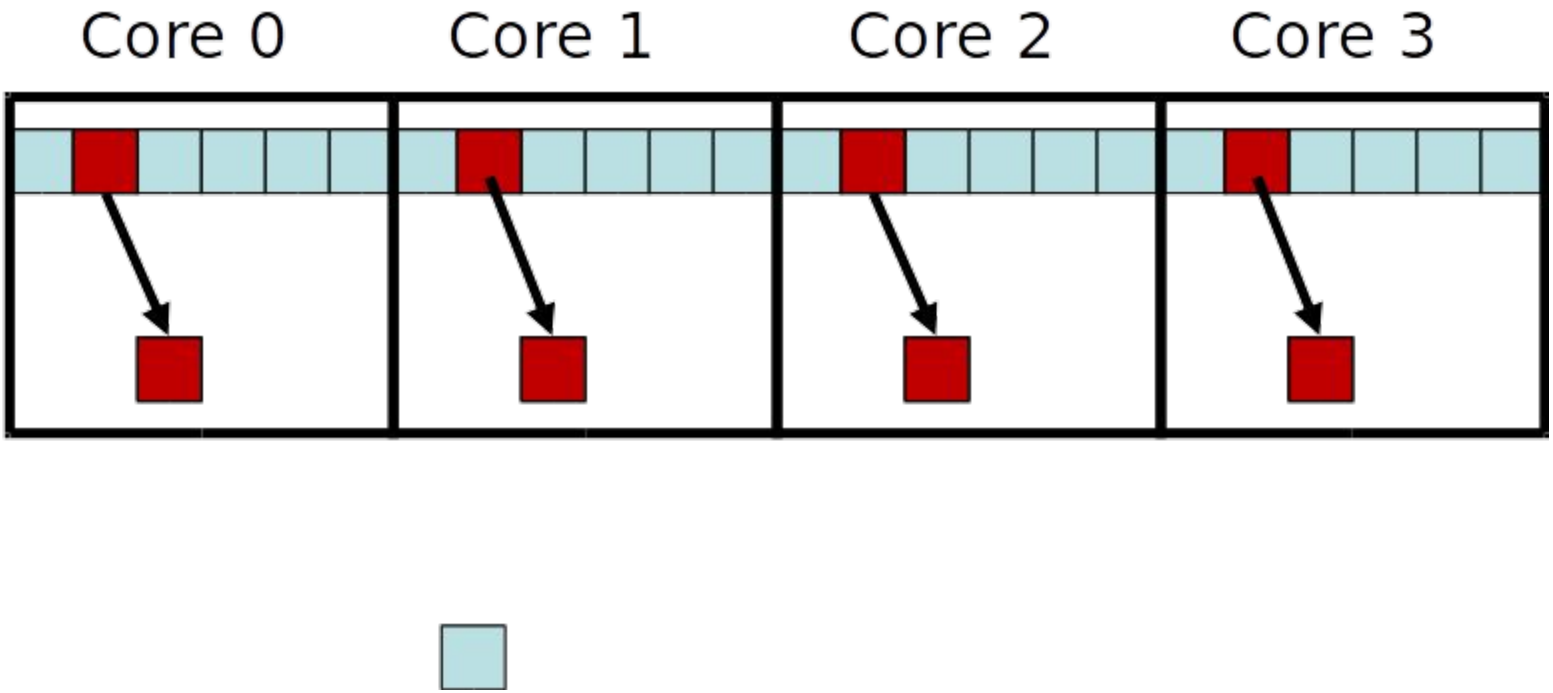
Find the largest element of an array



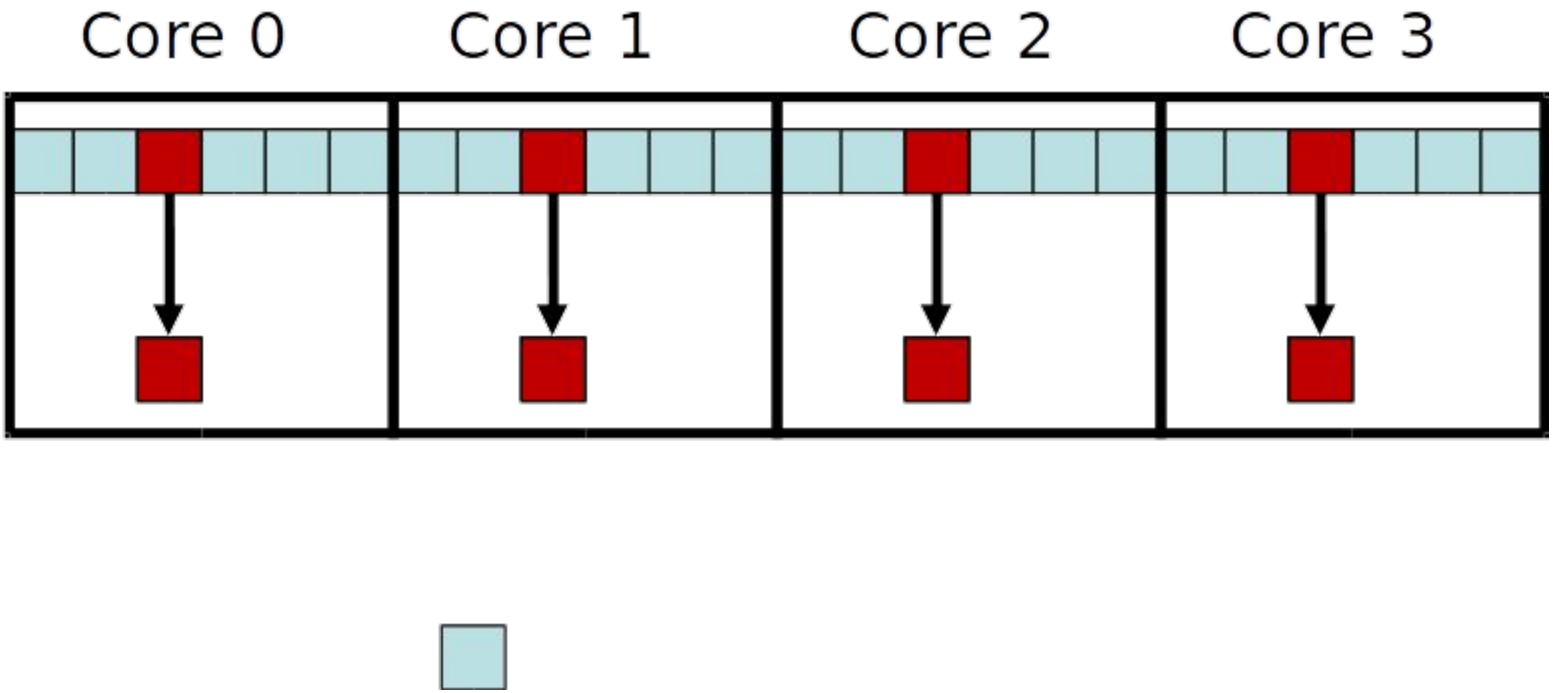
Find the largest element of an array



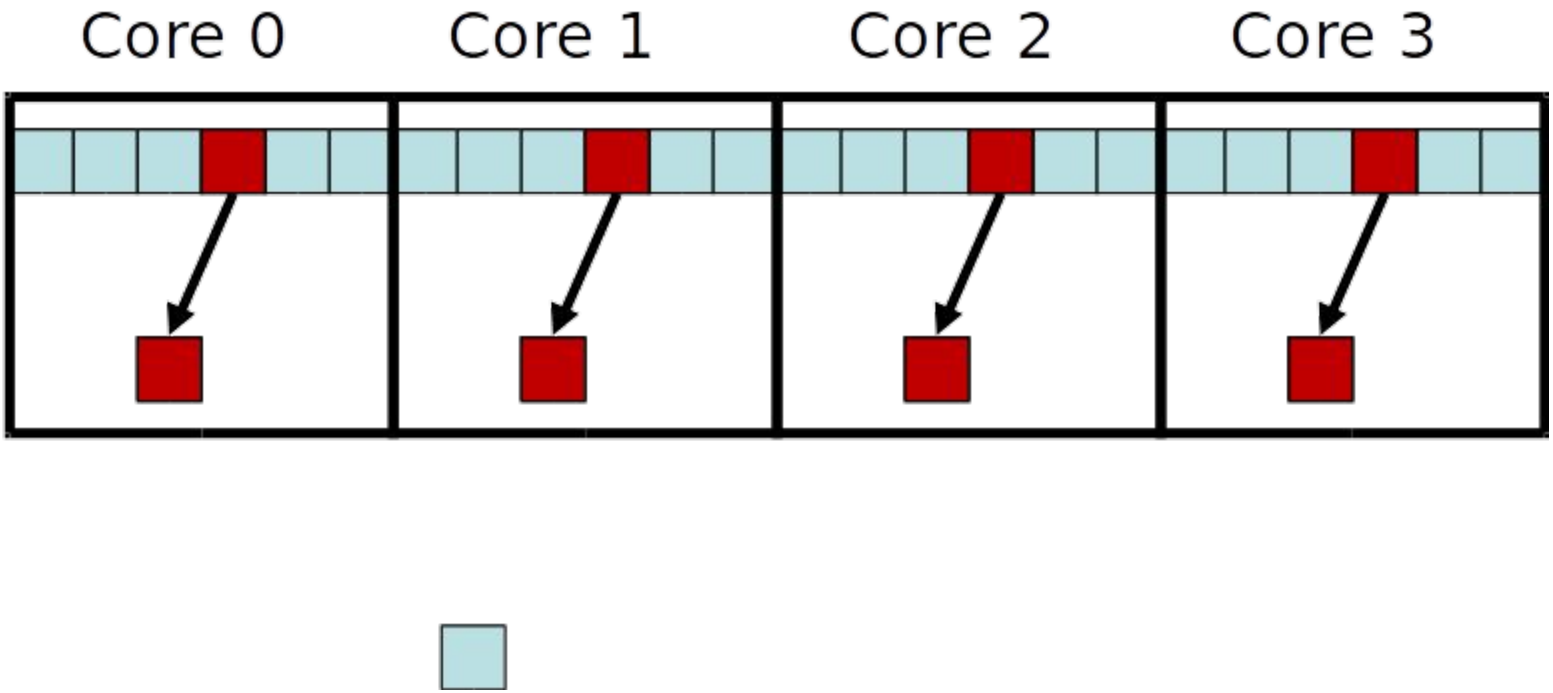
Find the largest element of an array



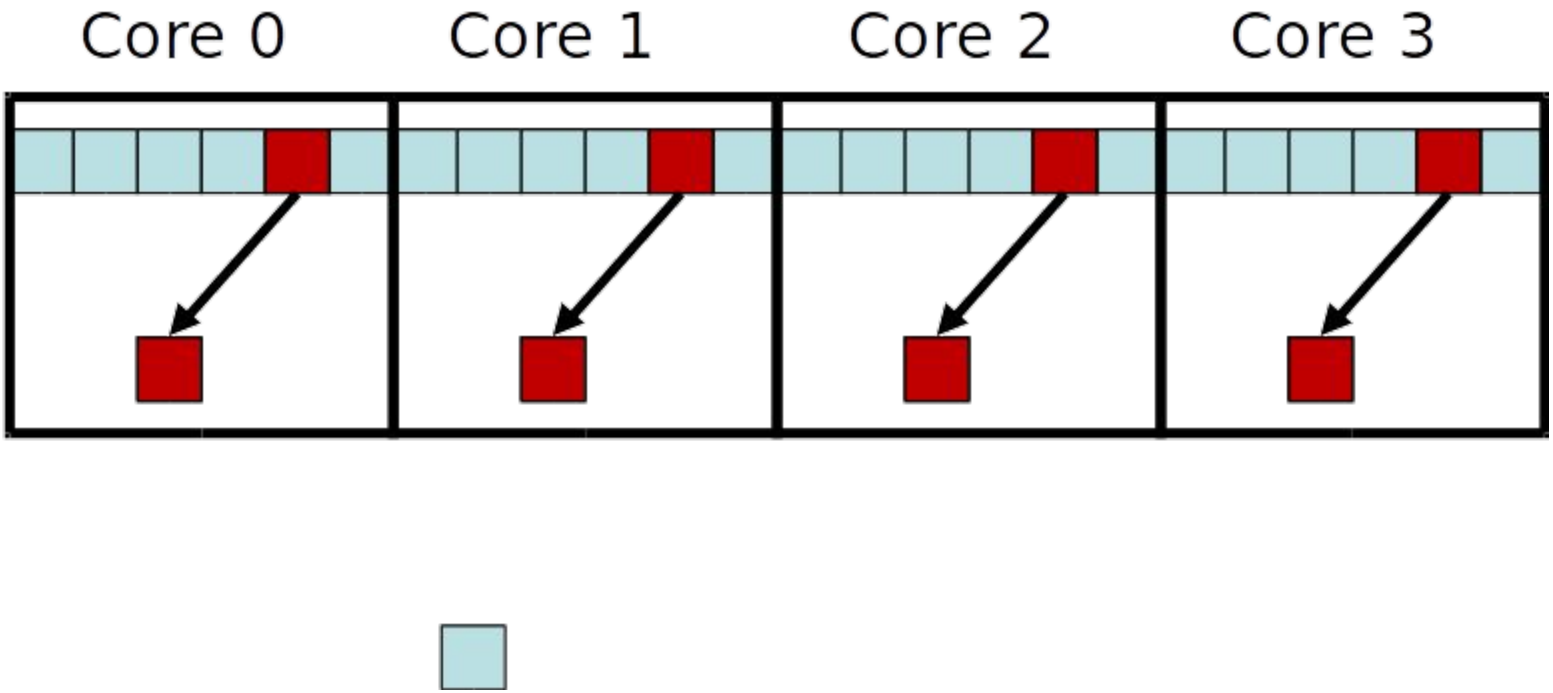
Find the largest element of an array



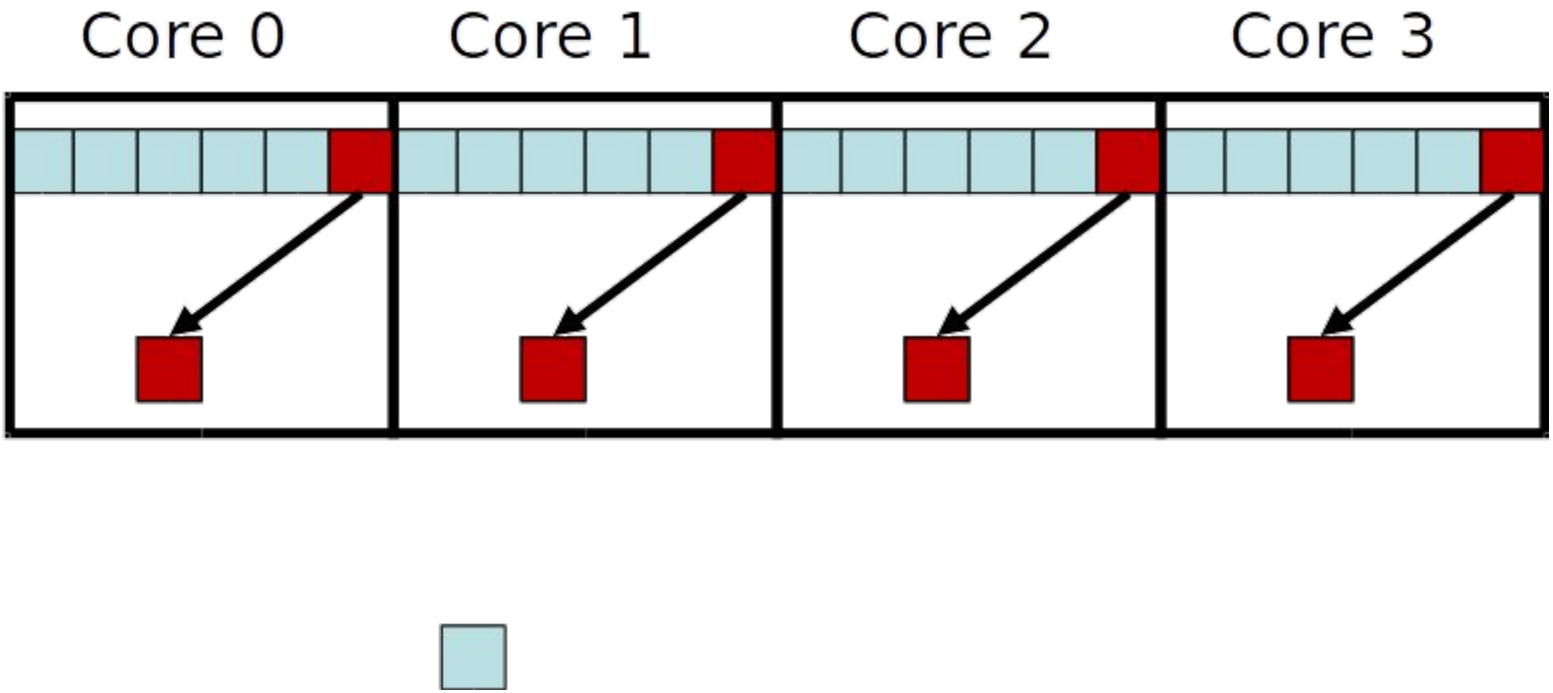
Find the largest element of an array



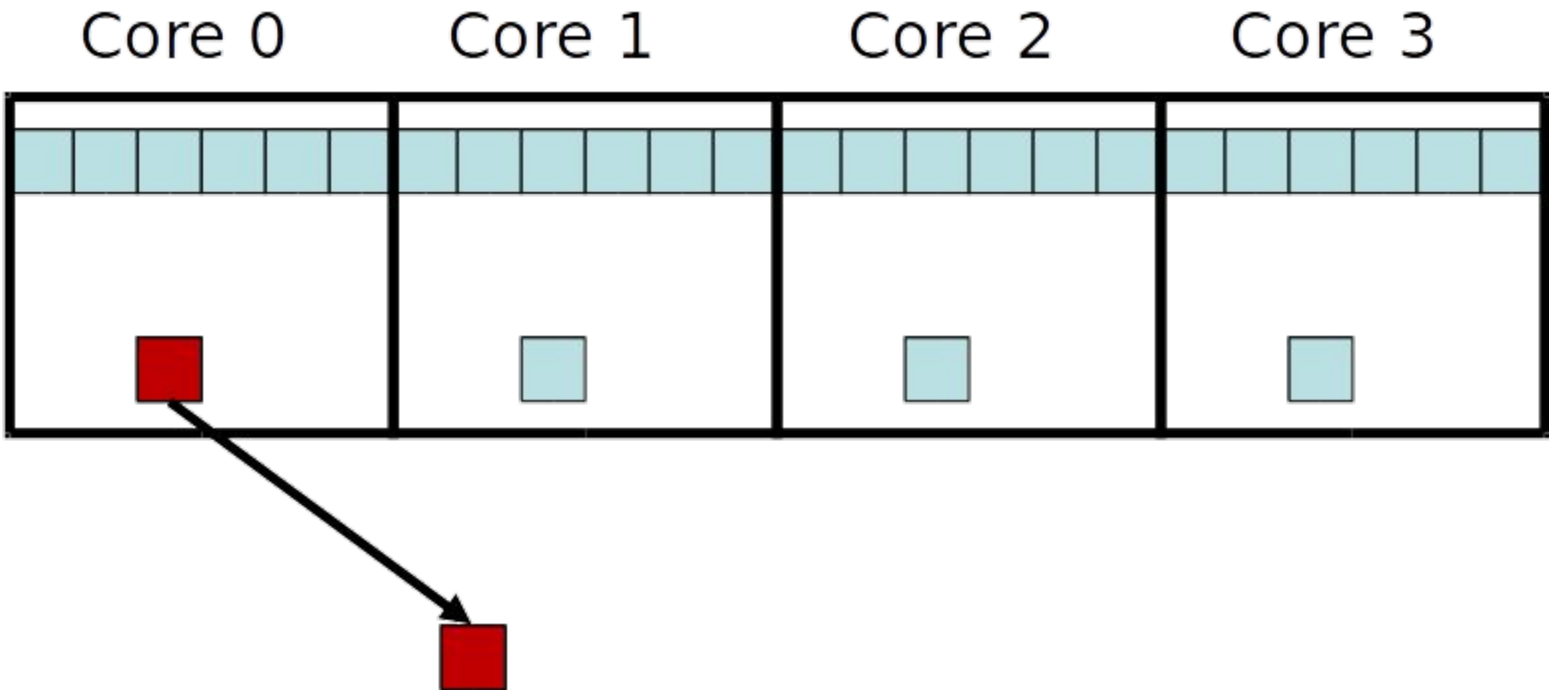
Find the largest element of an array



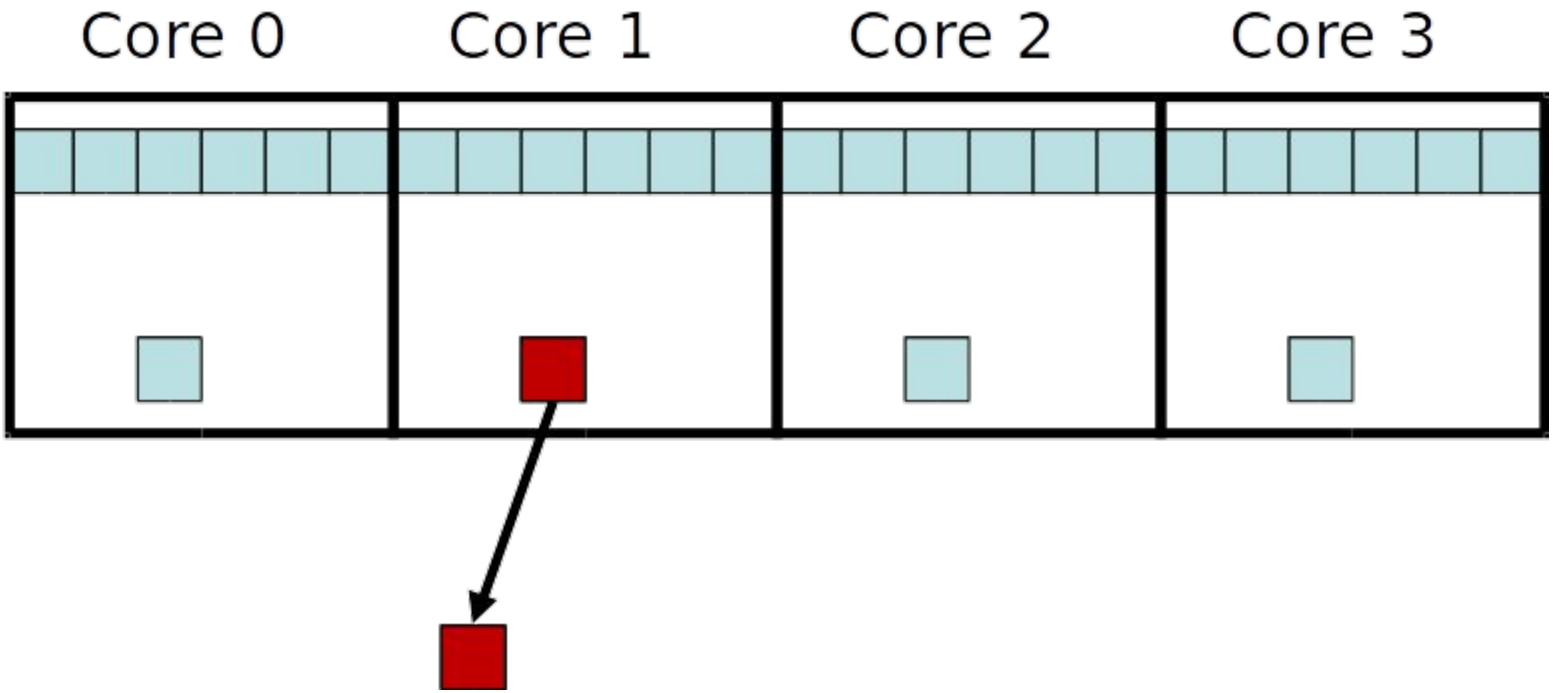
Find the largest element of an array



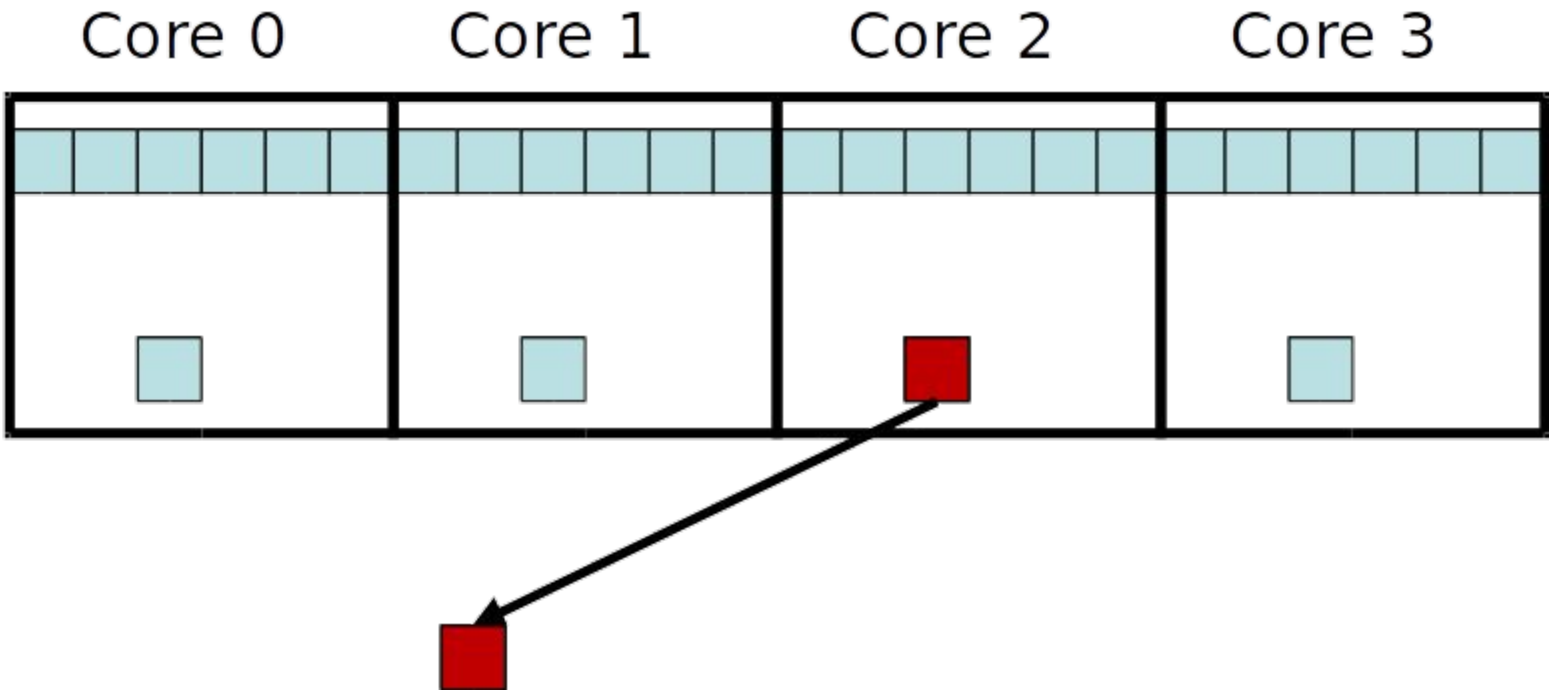
Find the largest element of an array



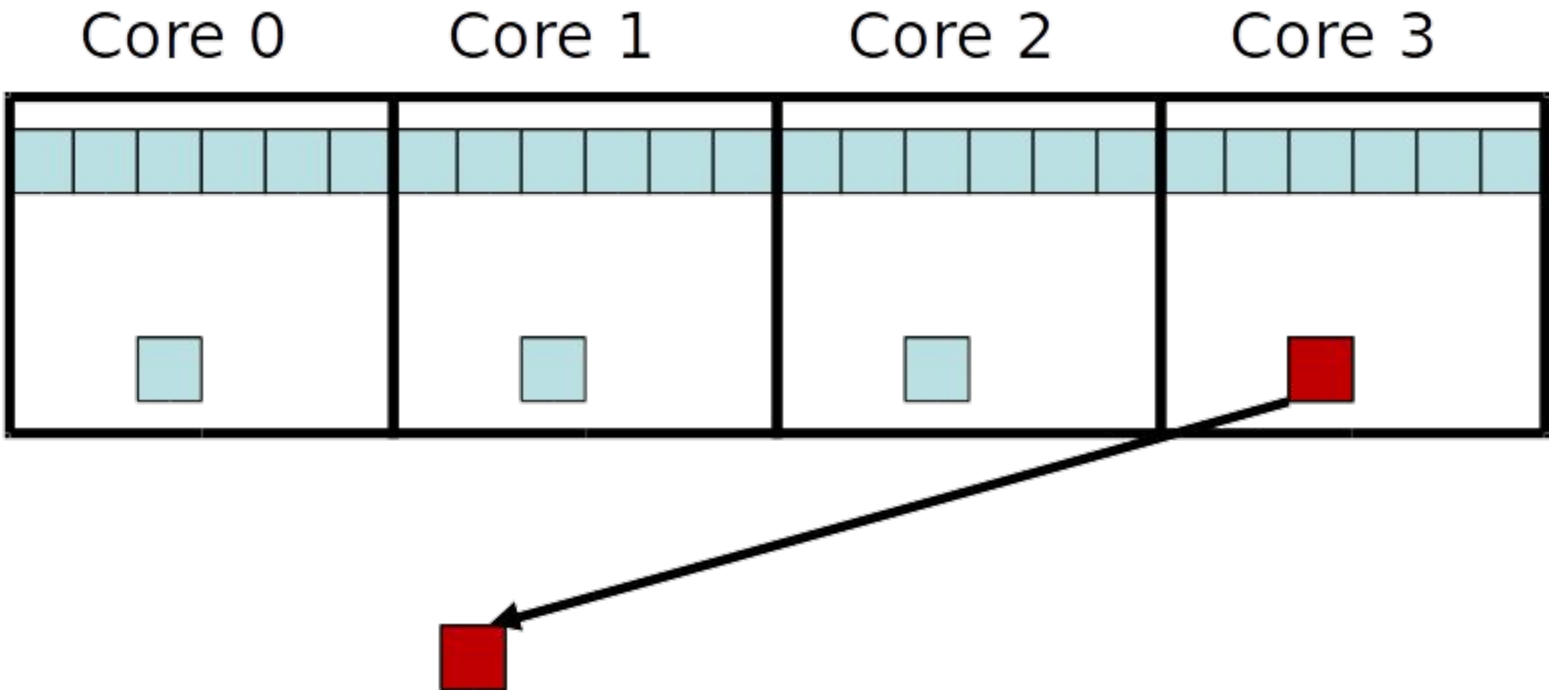
Find the largest element of an array



Find the largest element of an array



Find the largest element of an array

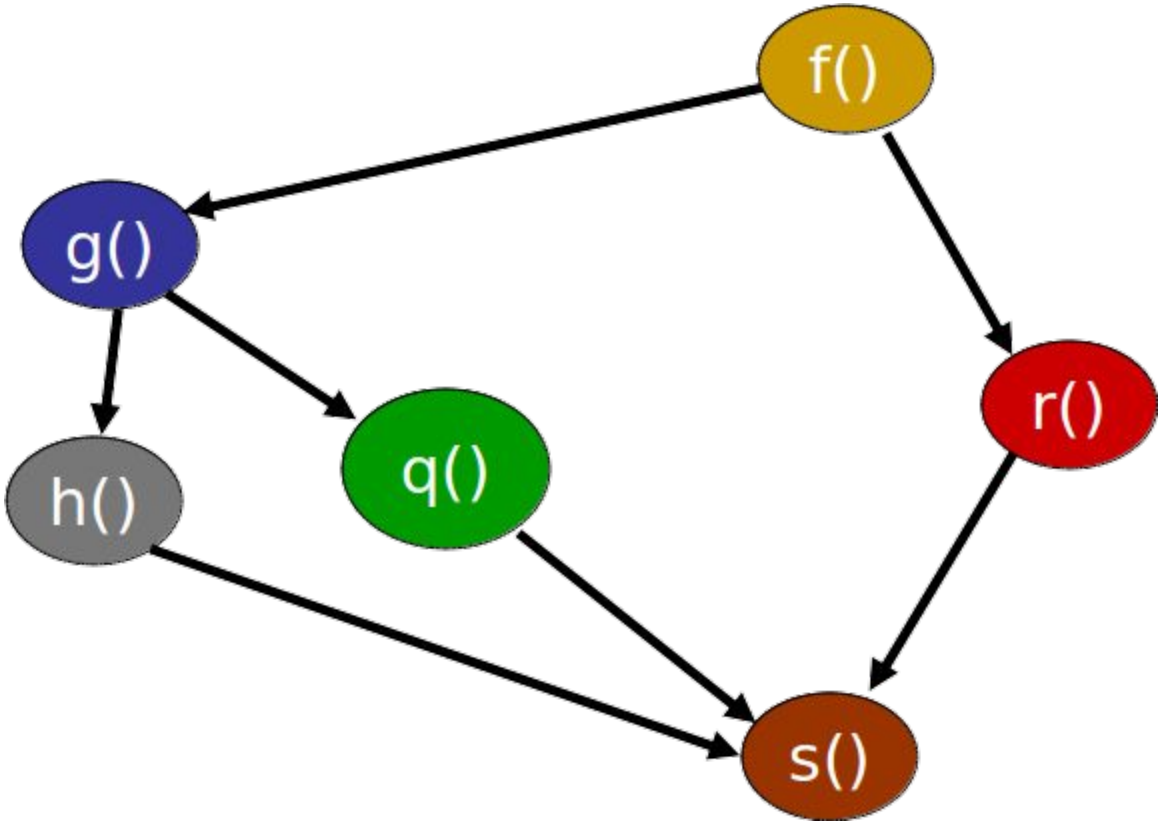


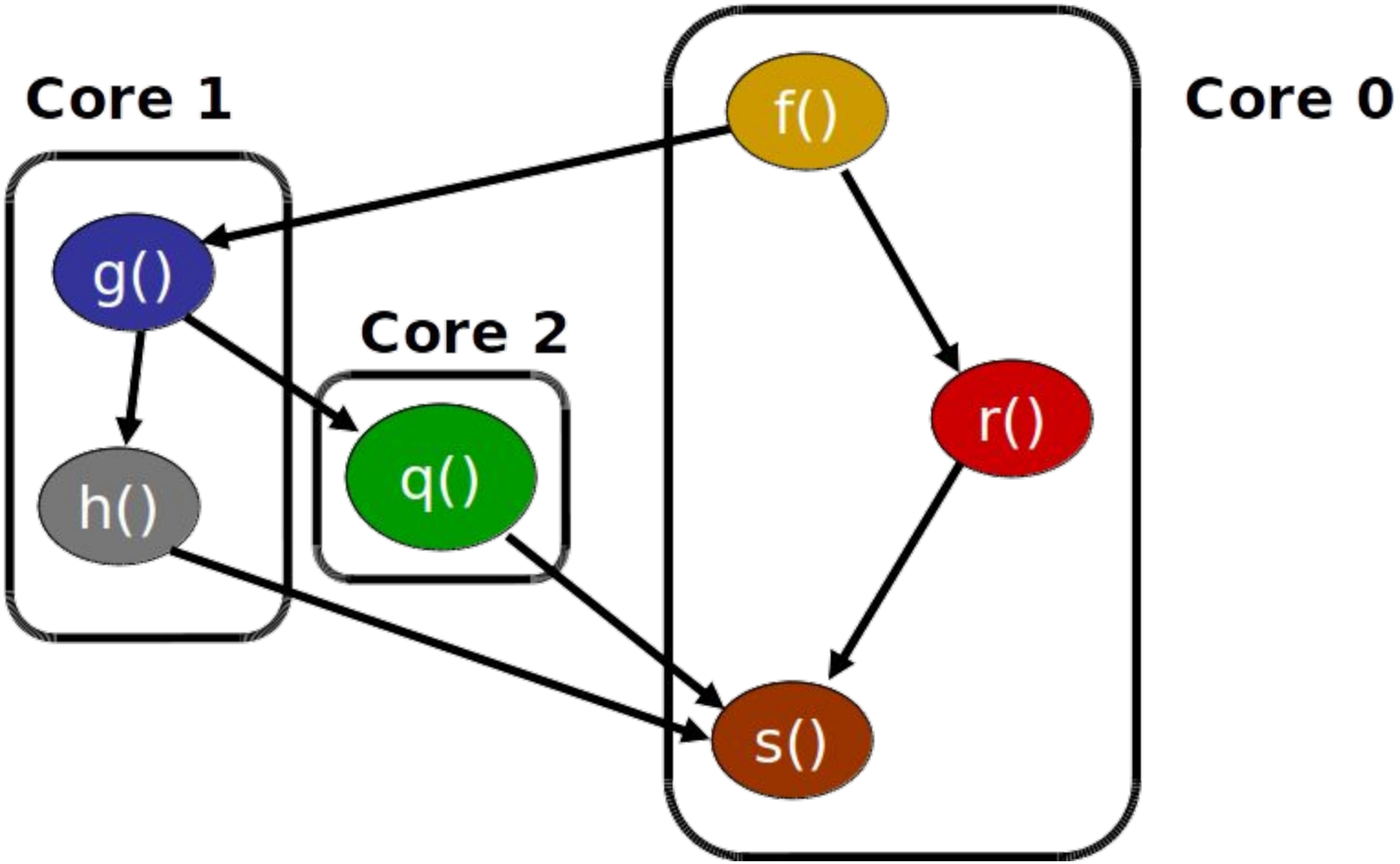
Task (Functional) Decomposition

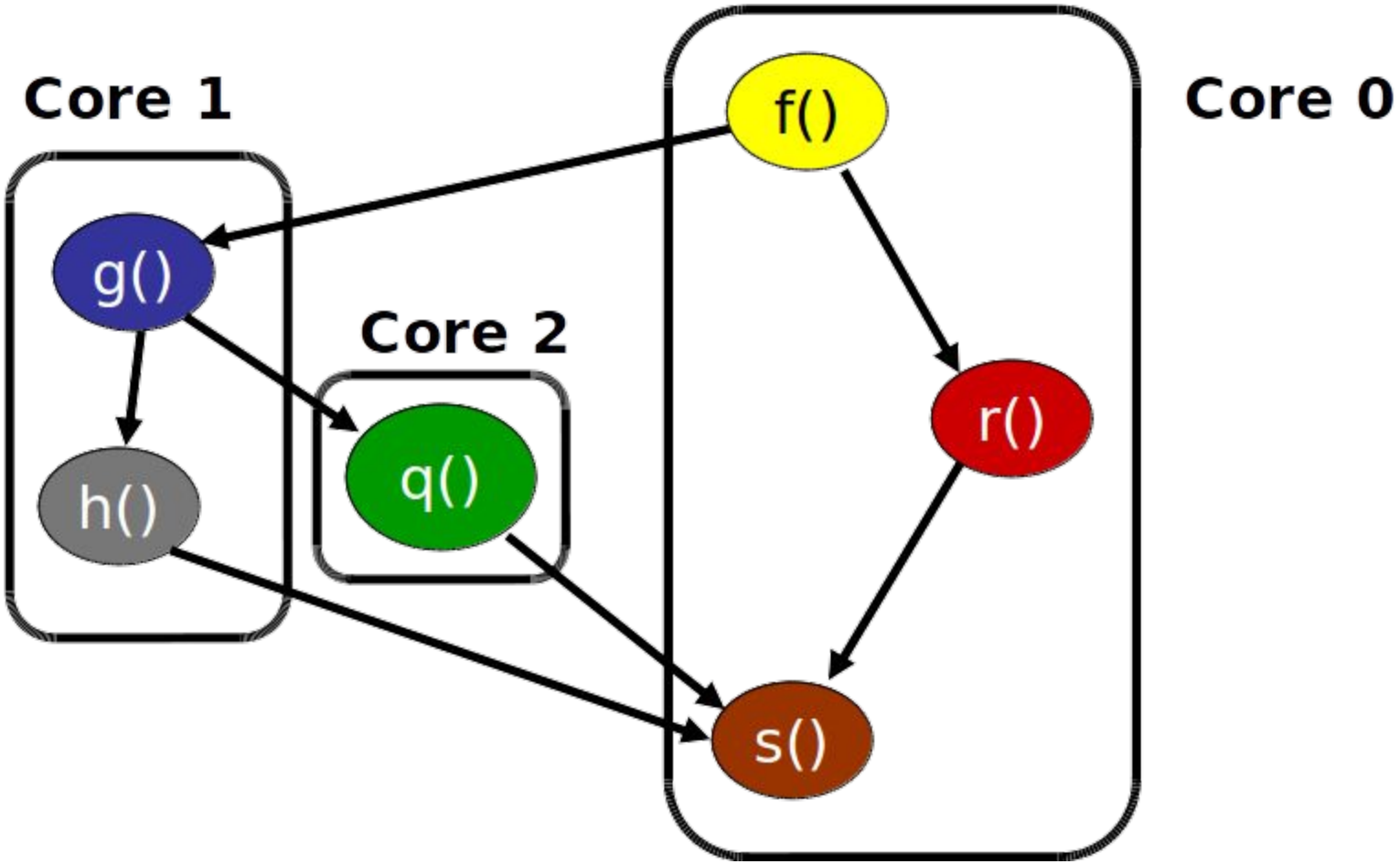
First, divide problem into independent tasks

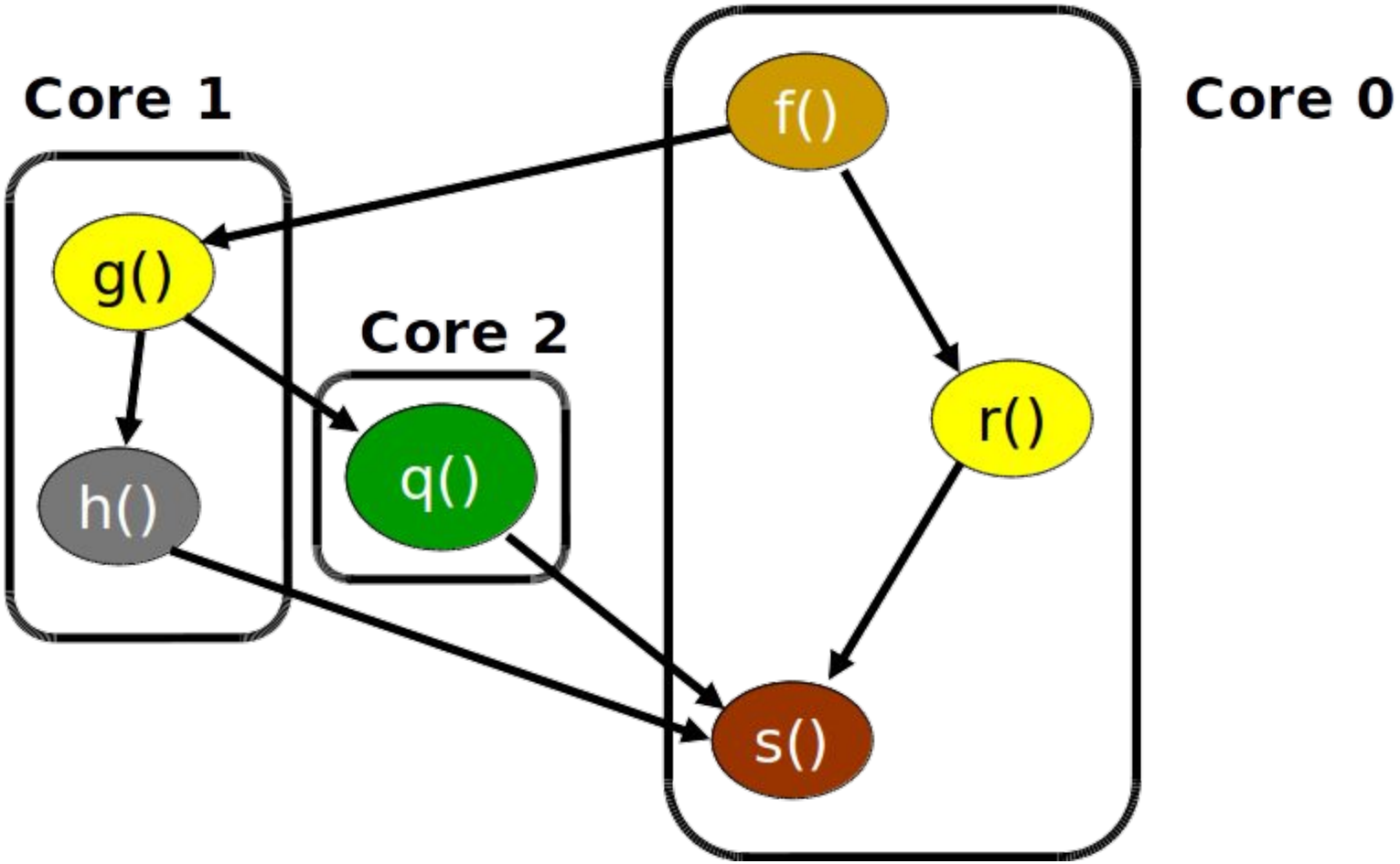
Second, decide which data elements are going to be accessed (read and/or written) by which tasks

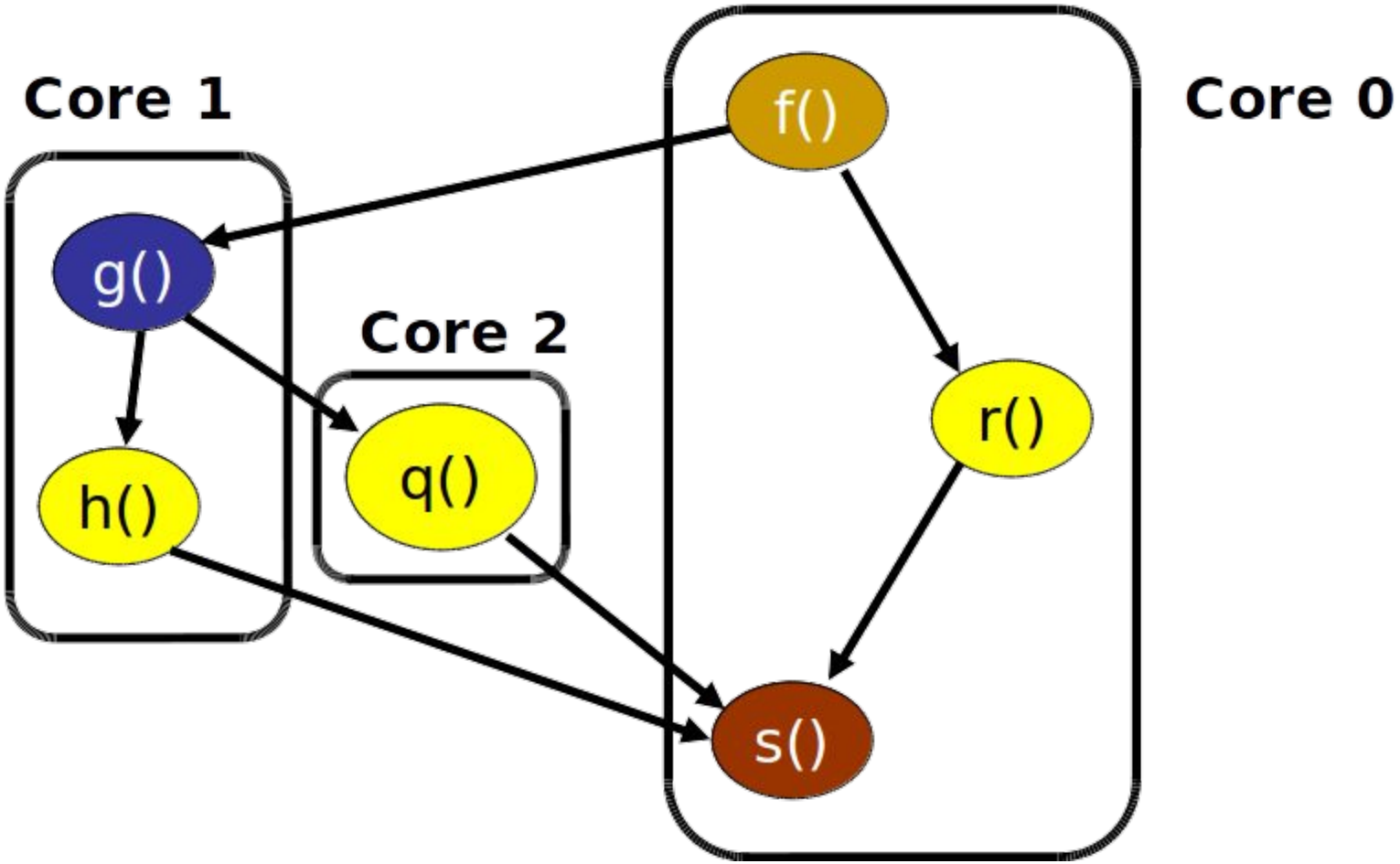
Example: Event-handler for GUI

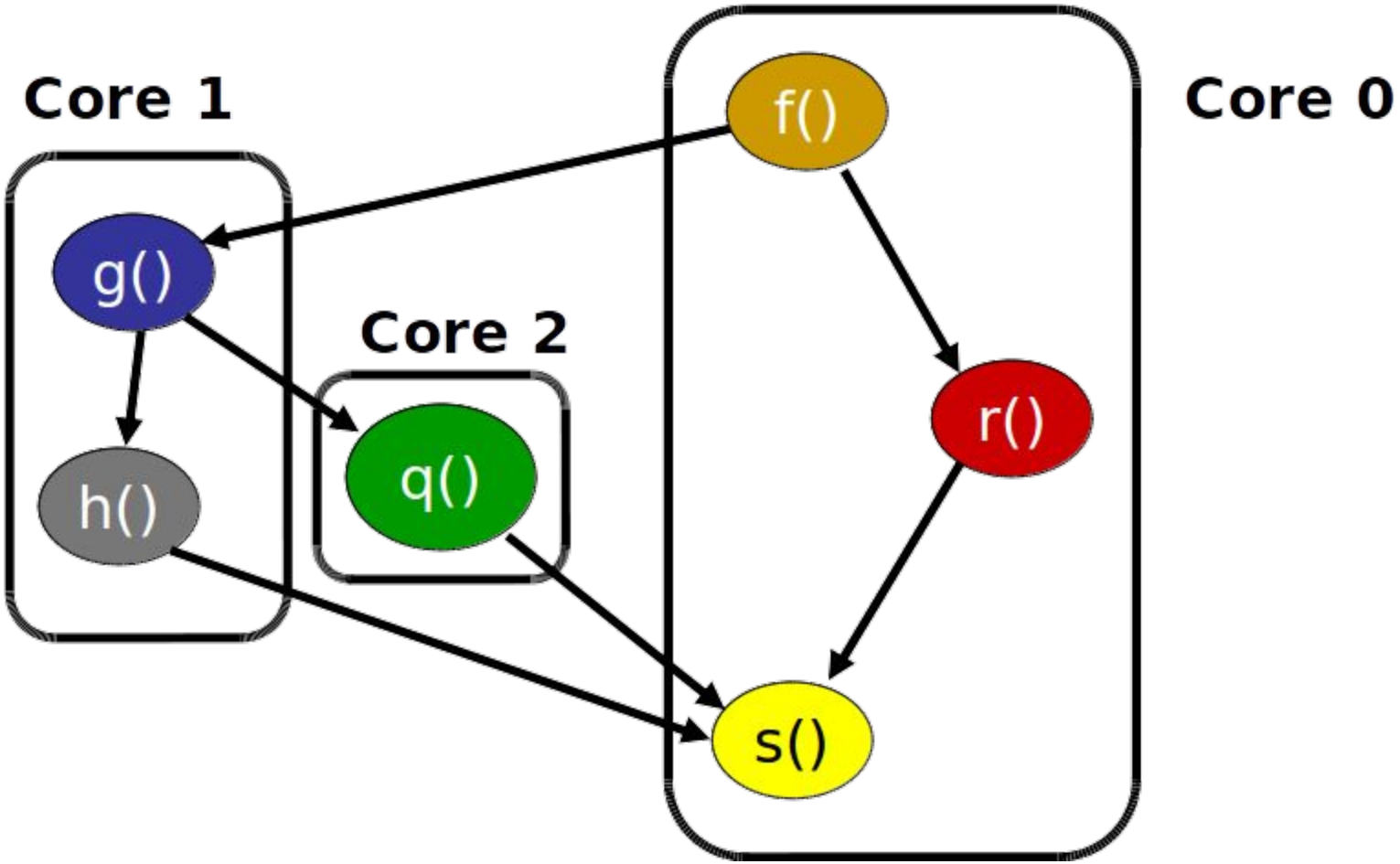

f()
g()
h()
q()
r()
s()


Core 1
Core 0
Core 2
f()
g()
h()
q()
r()
s()


Core 1
Core 0
Core 2
f()
g()
h()
q()
r()
s()


Core 1
Core 0
Core 2
f()
g()
h()
q()
r()
s()


Core 1
Core 0
Core 2
f()
g()
h()
q()
r()
s()


Core 1
Core 0
Core 2
f()
g()
h()
q()
r()
s()

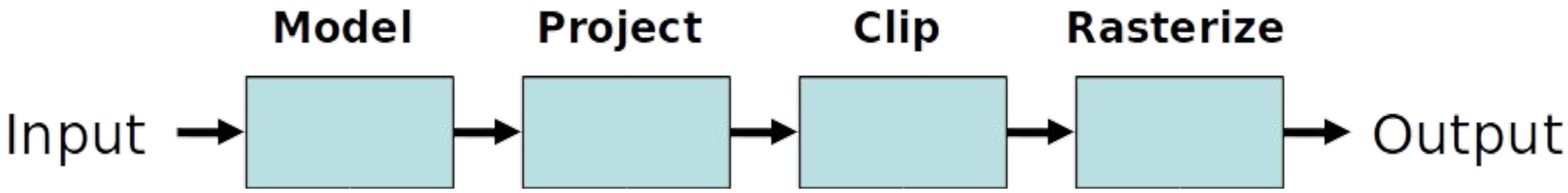
Pipelining

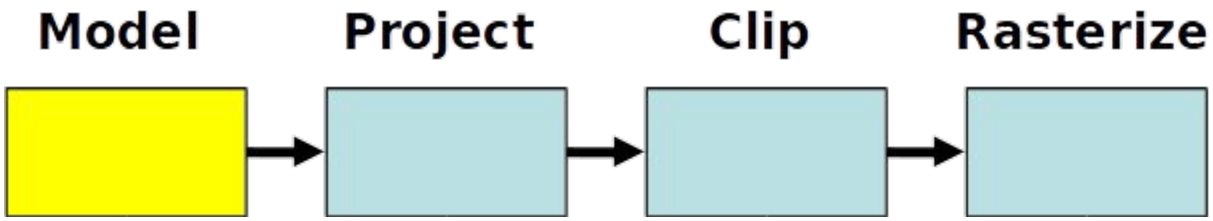
Special kind of task decomposition

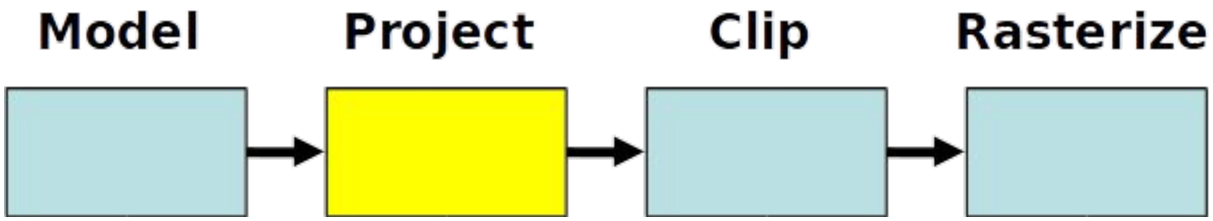
“Assembly line” parallelism

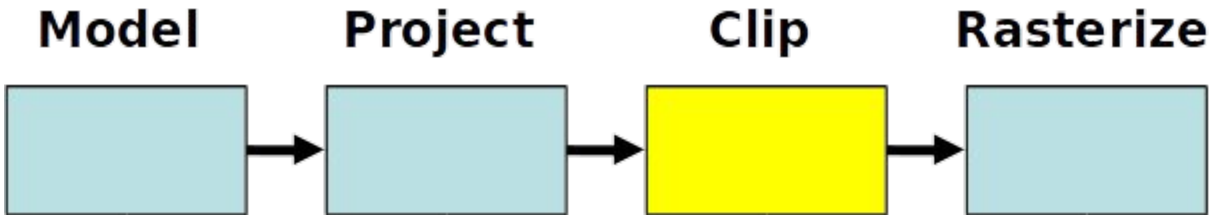
Example: 3D rendering in computer graphics

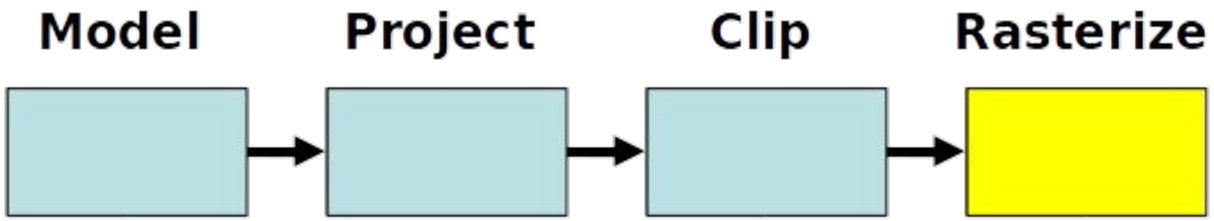
Pipelining



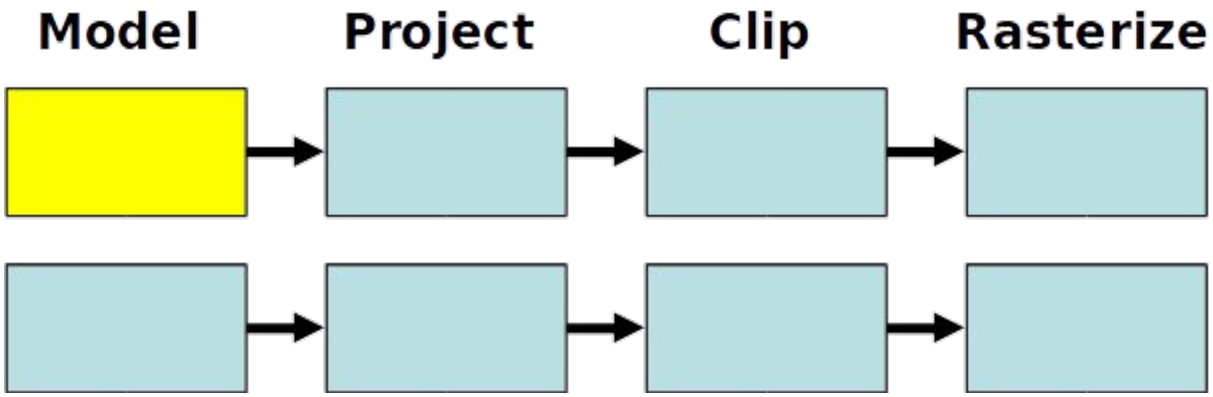

Model
Project
Clip
Rasterize

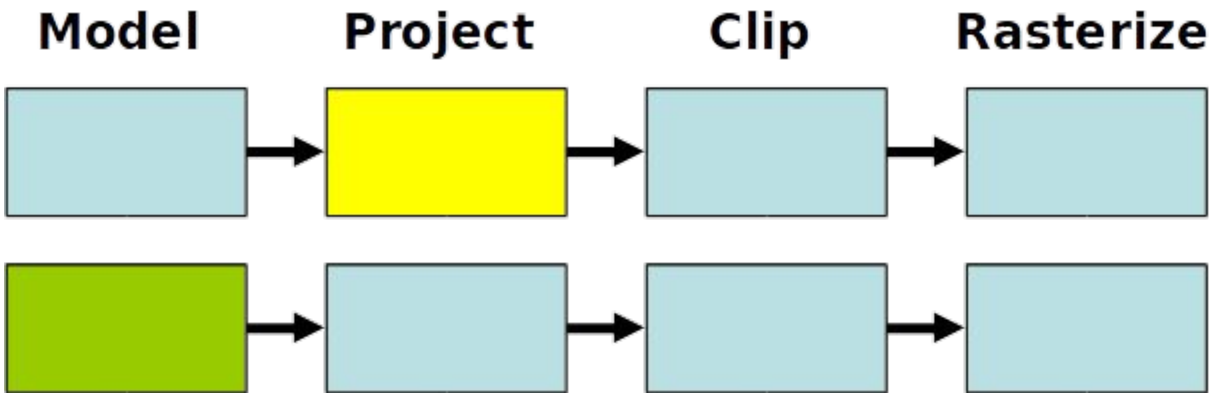

Model
Project
Clip
Rasterize

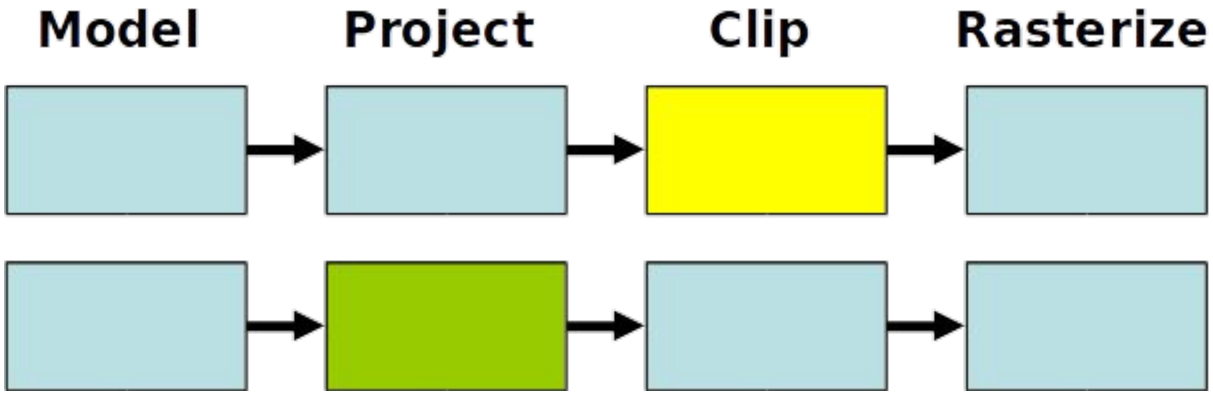

Model
Project
Clip
Rasterize

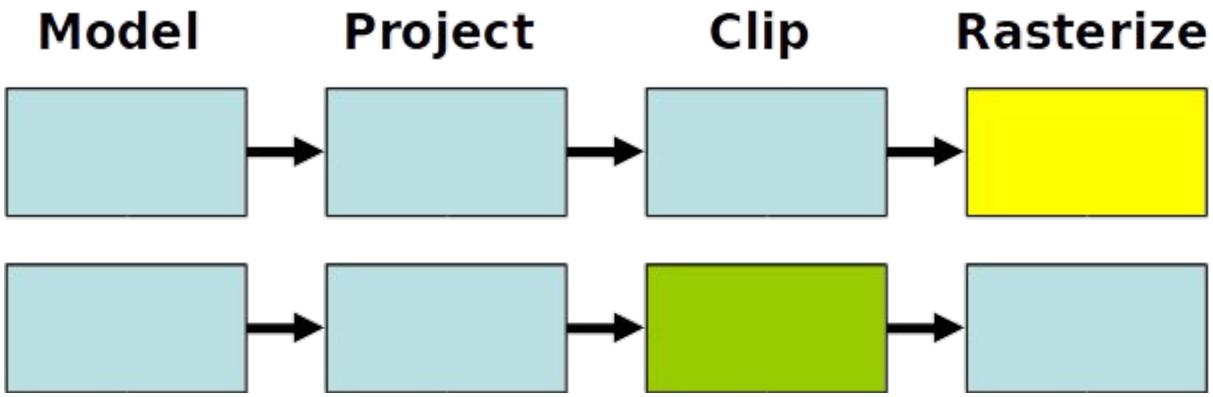


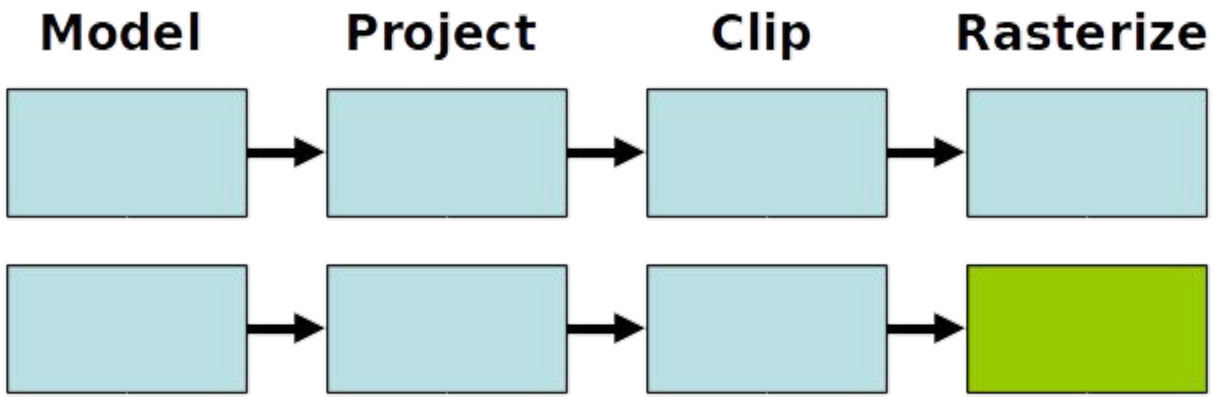
The pipeline processes 1 data set in 4 steps


Model
Project
Clip
Rasterize

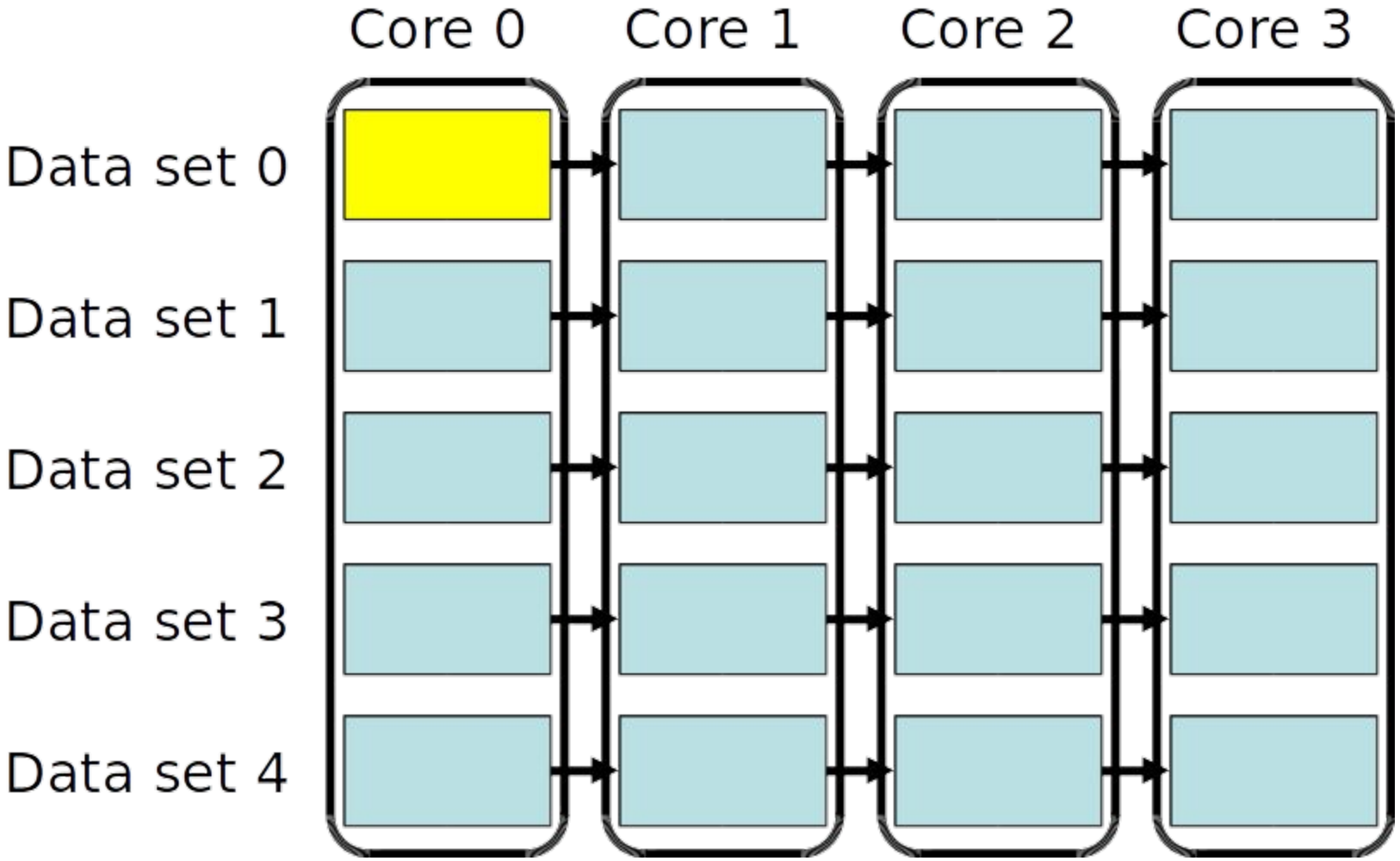

Model
Project
Clip
Rasterize

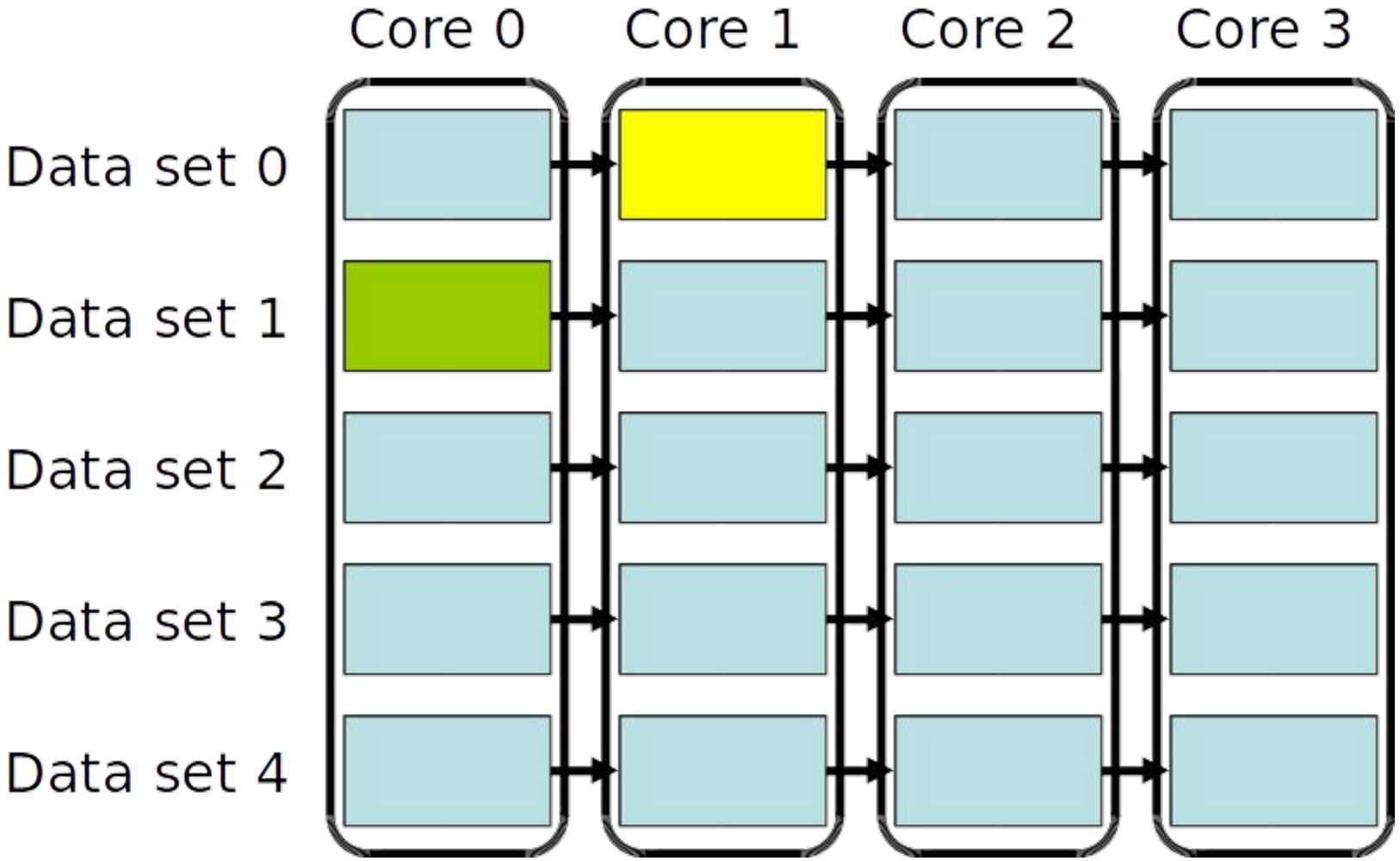

Model
Project
Clip
Rasterize

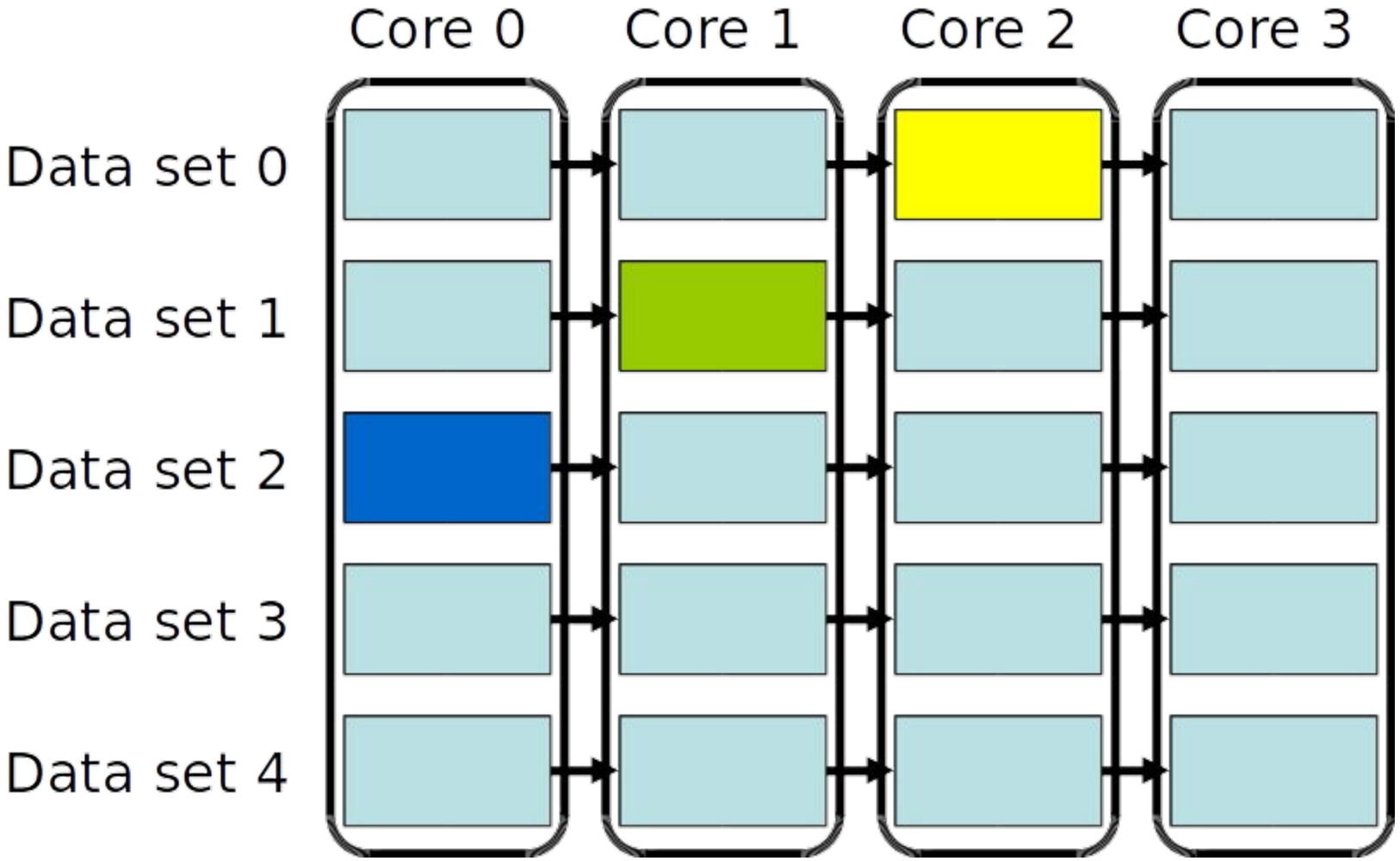

Model
Project
Clip
Rasterize

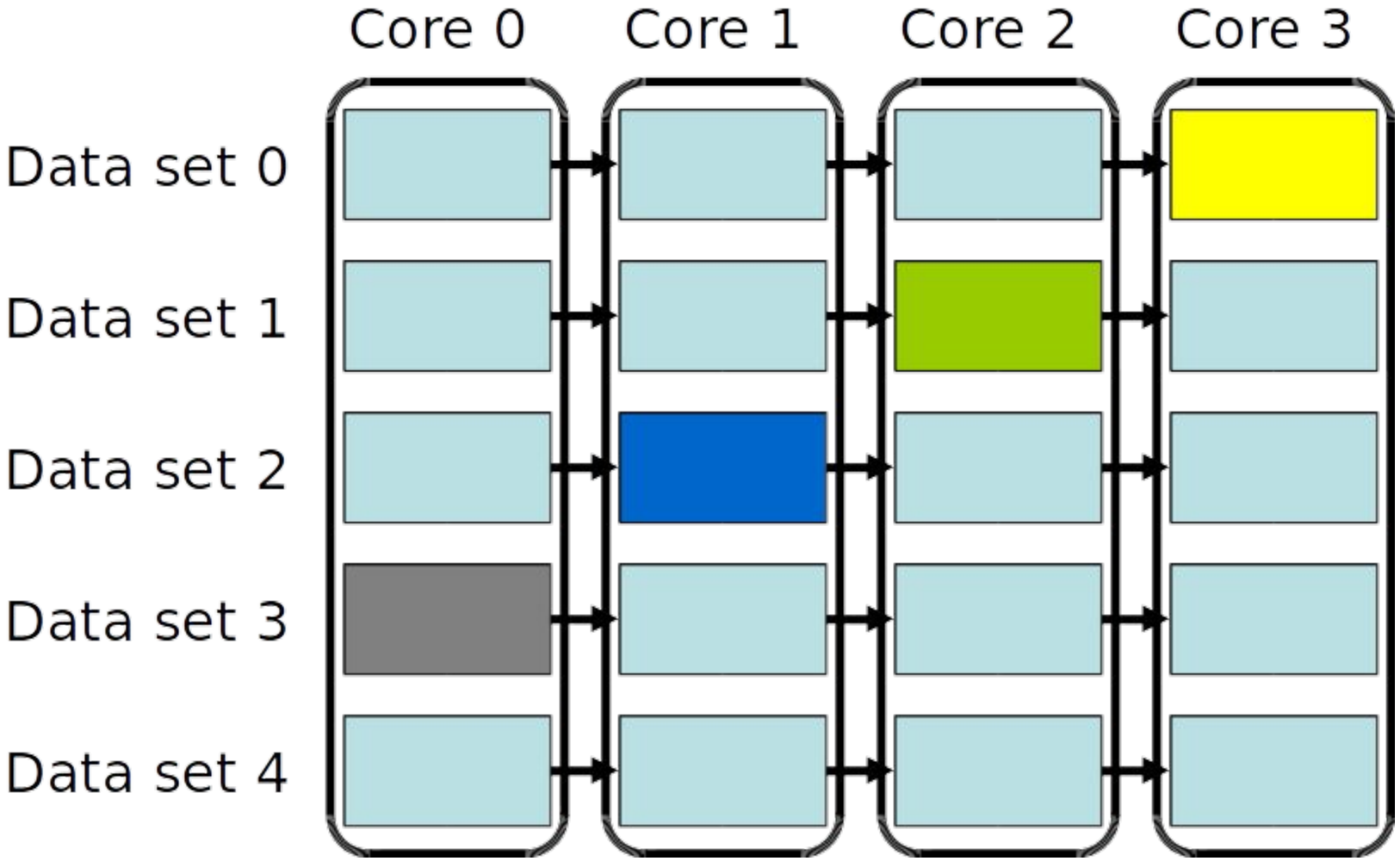


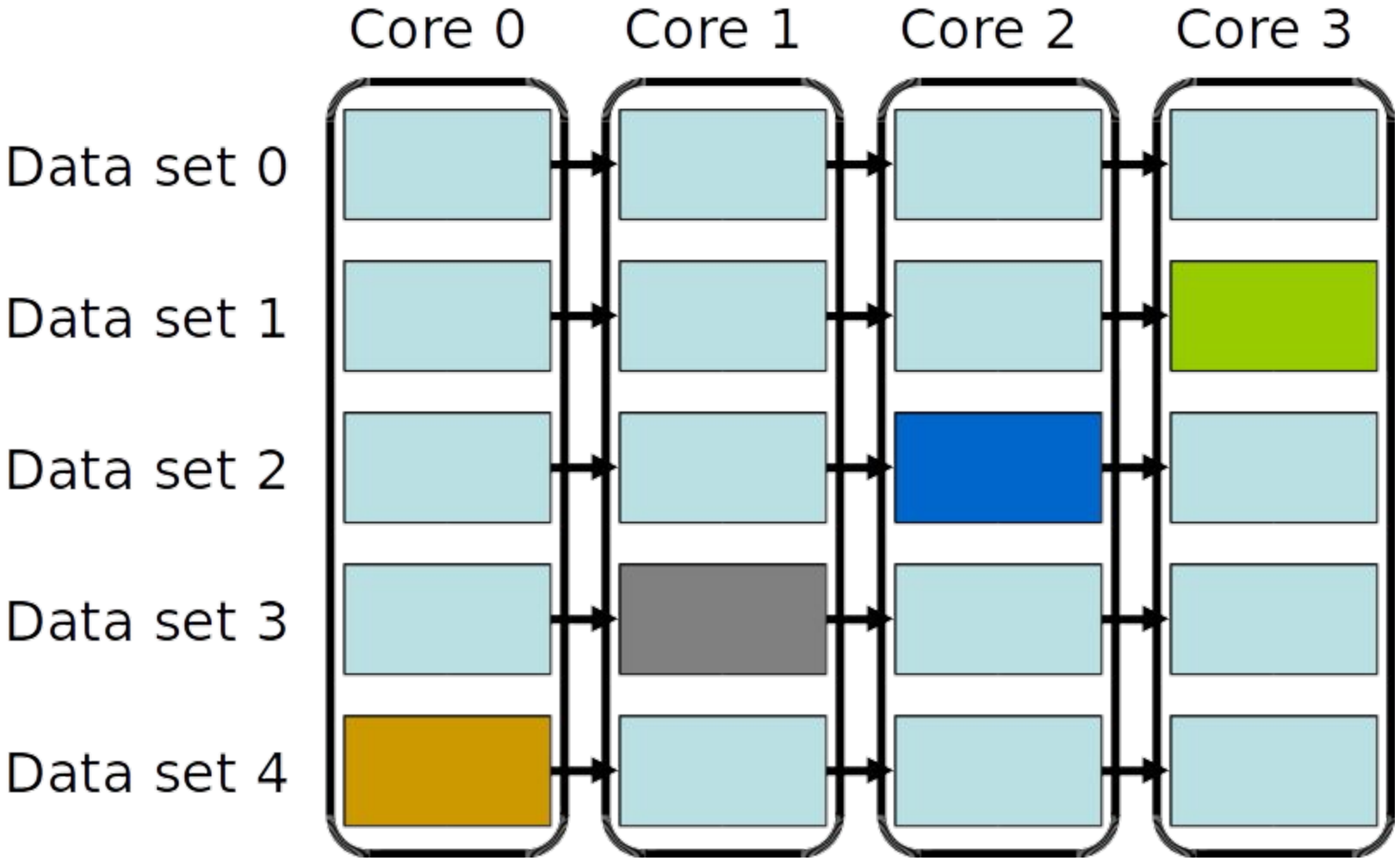
The pipeline processes 2 data sets in 5 steps

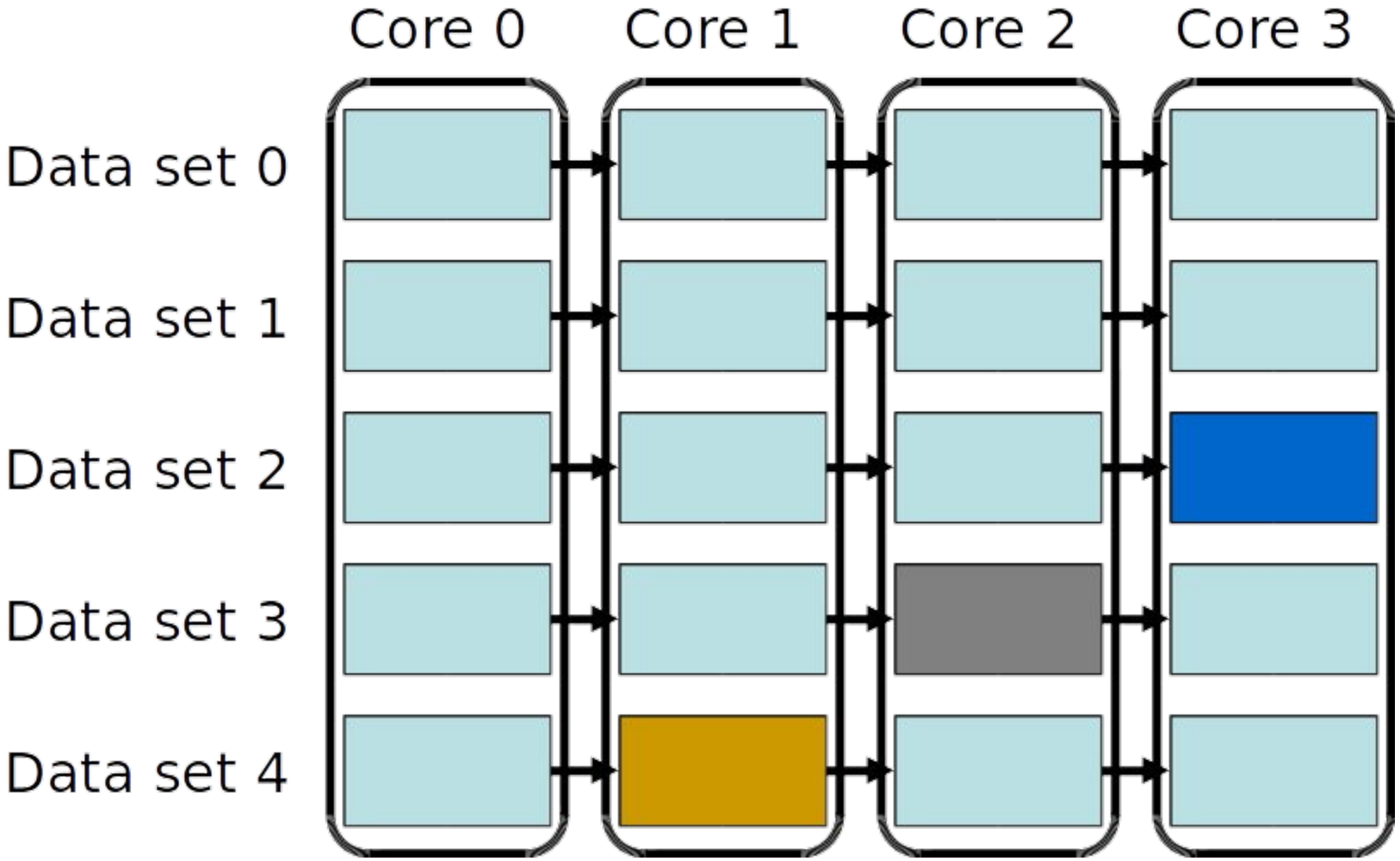

Core 0
Core 1
Core 2
Core 3
Data set 0
Data set 1
Data set 2
Data set 3
Data set 4

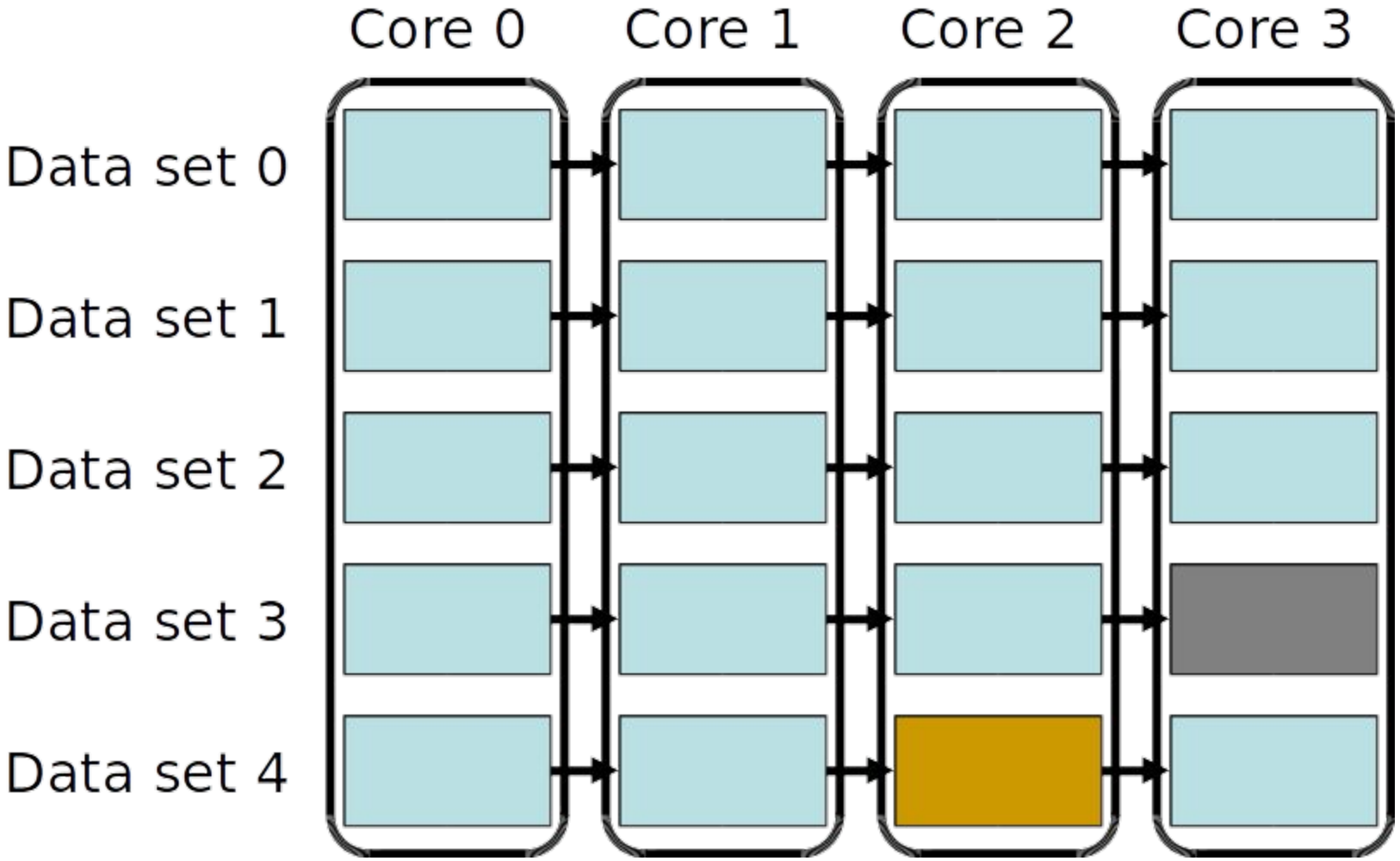

Core 0
Core 1
Core 2
Core 3
Data set 0
Data set 1
Data set 2
Data set 3
Data set 4

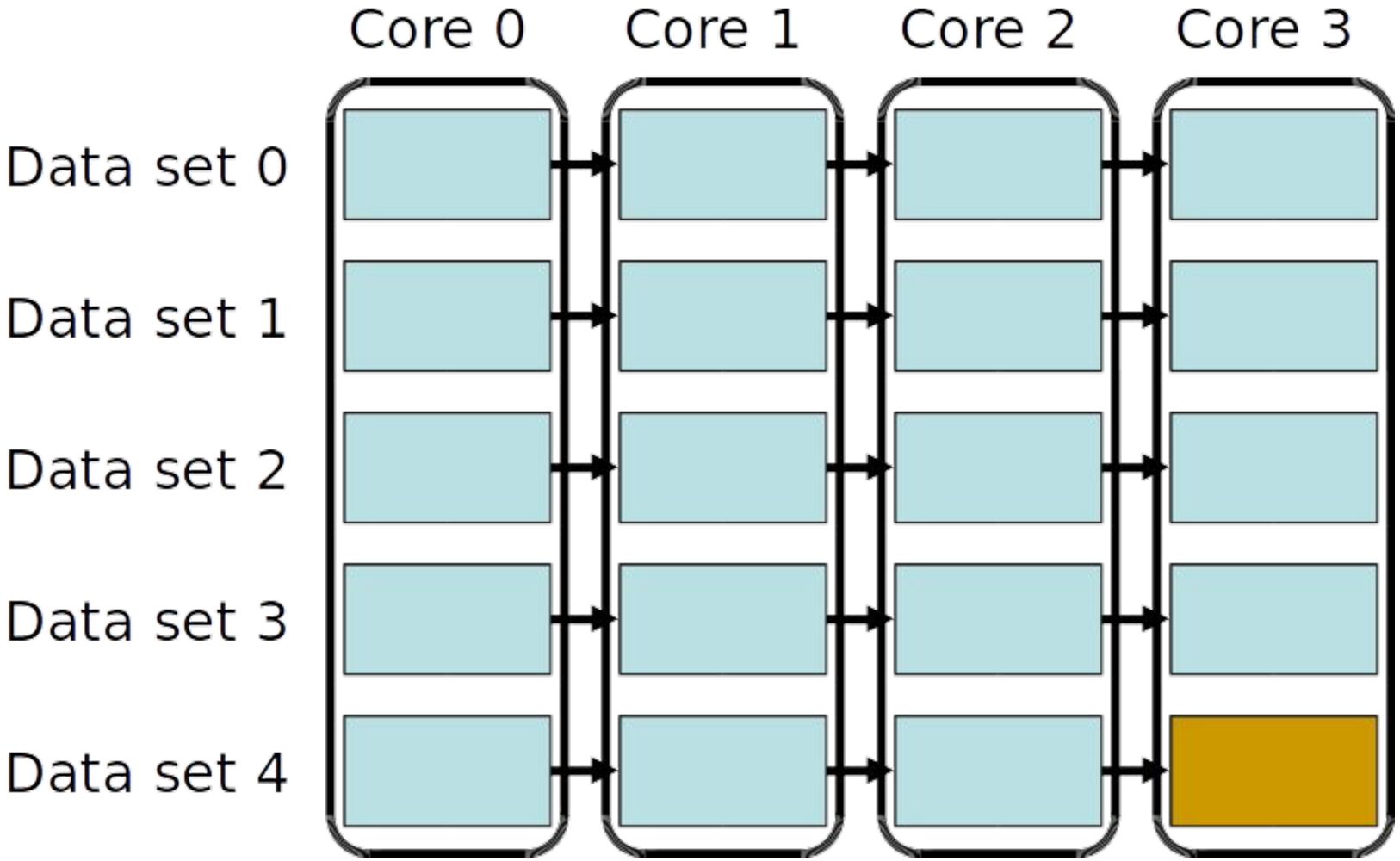

Core 0
Core 1
Core 2
Core 3
Data set 0
Data set 1
Data set 2
Data set 3
Data set 4


Core 0
Core 1
Core 2
Core 3
Data set 0
Data set 1
Data set 2
Data set 3
Data set 4


Core 0
Core 1
Core 2
Core 3
Data set 0
Data set 1
Data set 2
Data set 3
Data set 4


Core 0
Core 1
Core 2
Core 3
Data set 0
Data set 1
Data set 2
Data set 3
Data set 4


Core 0
Core 1
Core 2
Core 3
Data set 0
Data set 1
Data set 2
Data set 3
Data set 4


Core 0
Core 1
Core 2
Core 3
Data set 0
Data set 1
Data set 2
Data set 3
Data set 4

Where can you find Independent Tasks

If you are familiar with the application, you should have an idea about where independent computations can be found

- Is there a large data set involved?
- Are there portions of code that can execute in a different order?
- Is the computation a set of stages that don't interact except for using the output of one as the input for the next?

A more formal method of discovering parallelism uses dependence graphs

Dependence Graph

Graph = (nodes, arrows)

Node for each

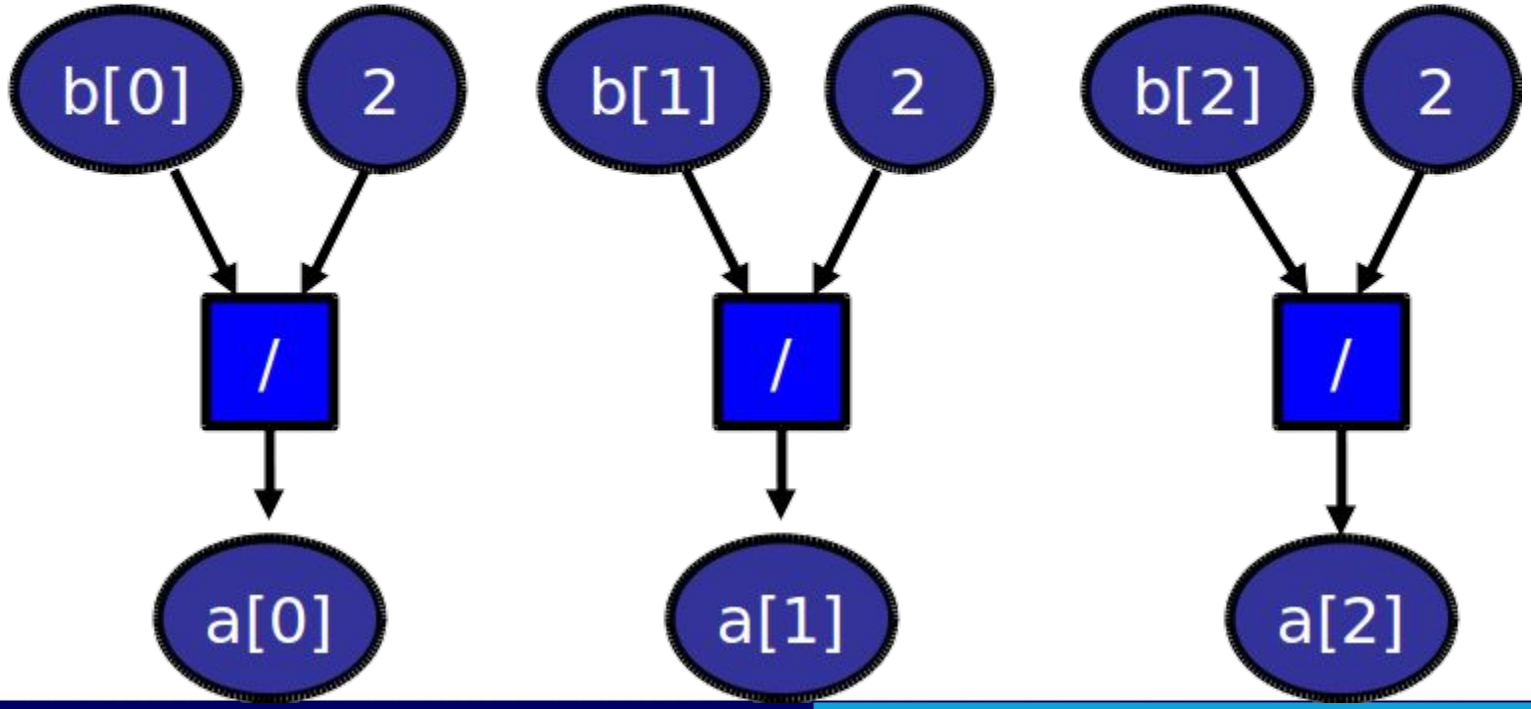
- Variable assignment (except index variables)
- Constant
- Operator or function call

Arrows indicate use of variables and constants

- Data flow
- Control flow

Dependence Graph Example #1

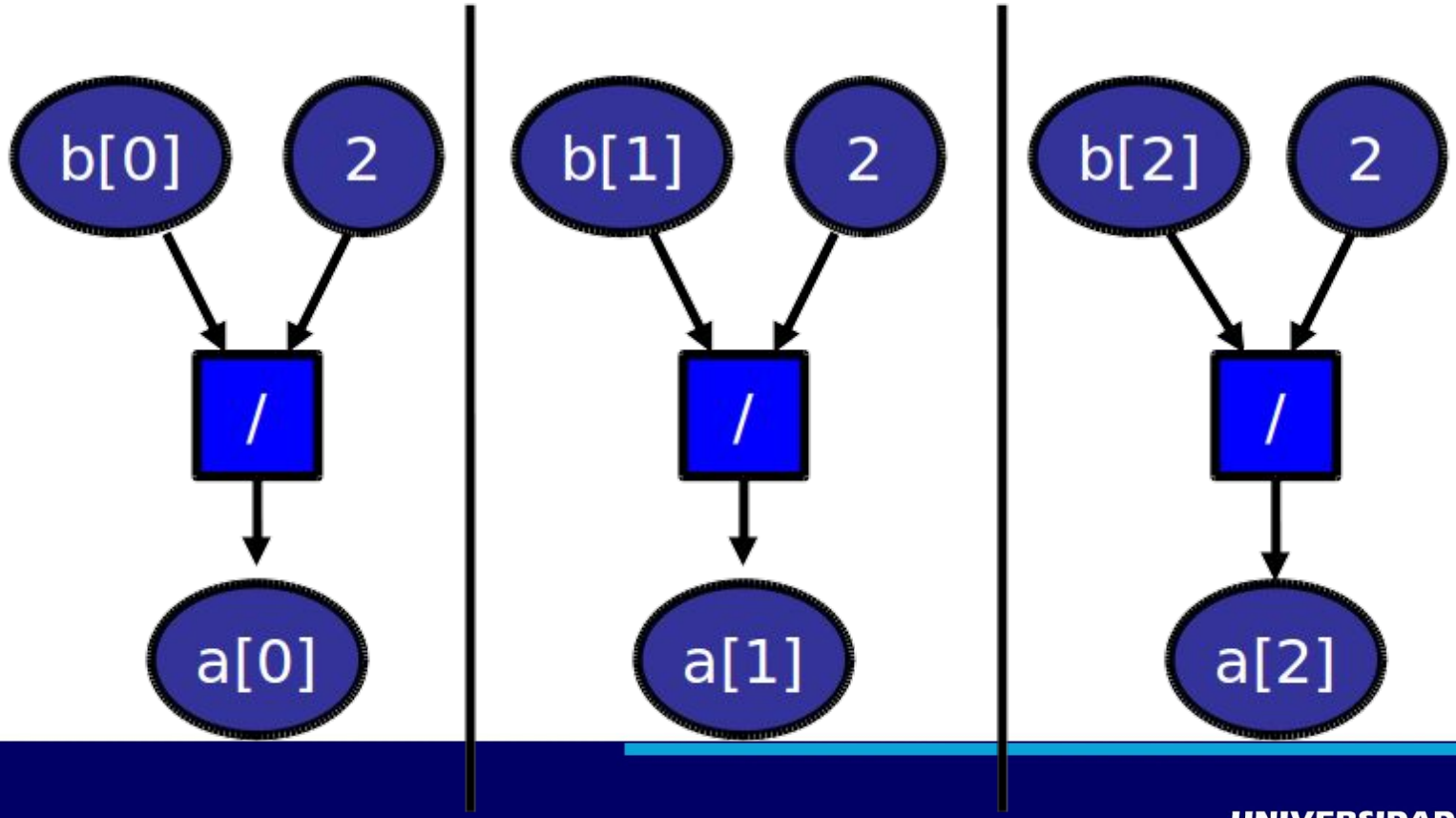
```
for (i = 0; i < 3; i++)
   a[i] = b[i] / 2.0;
```



Dependence Graph Example #1

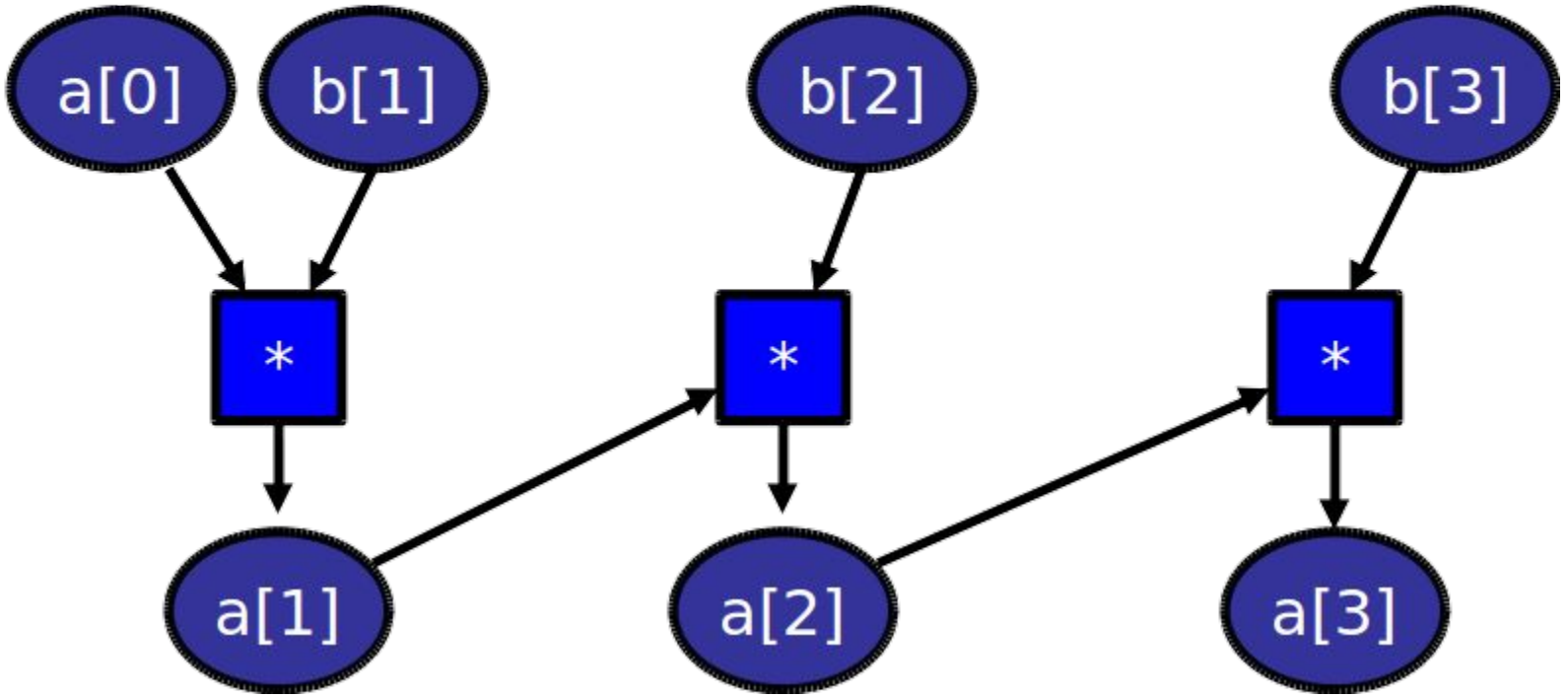
```
for (i = 0; i < 3; i++)
  a[i] = b[i] / 2.0;
```

Domain decomposition possible



Dependence Graph Example #2

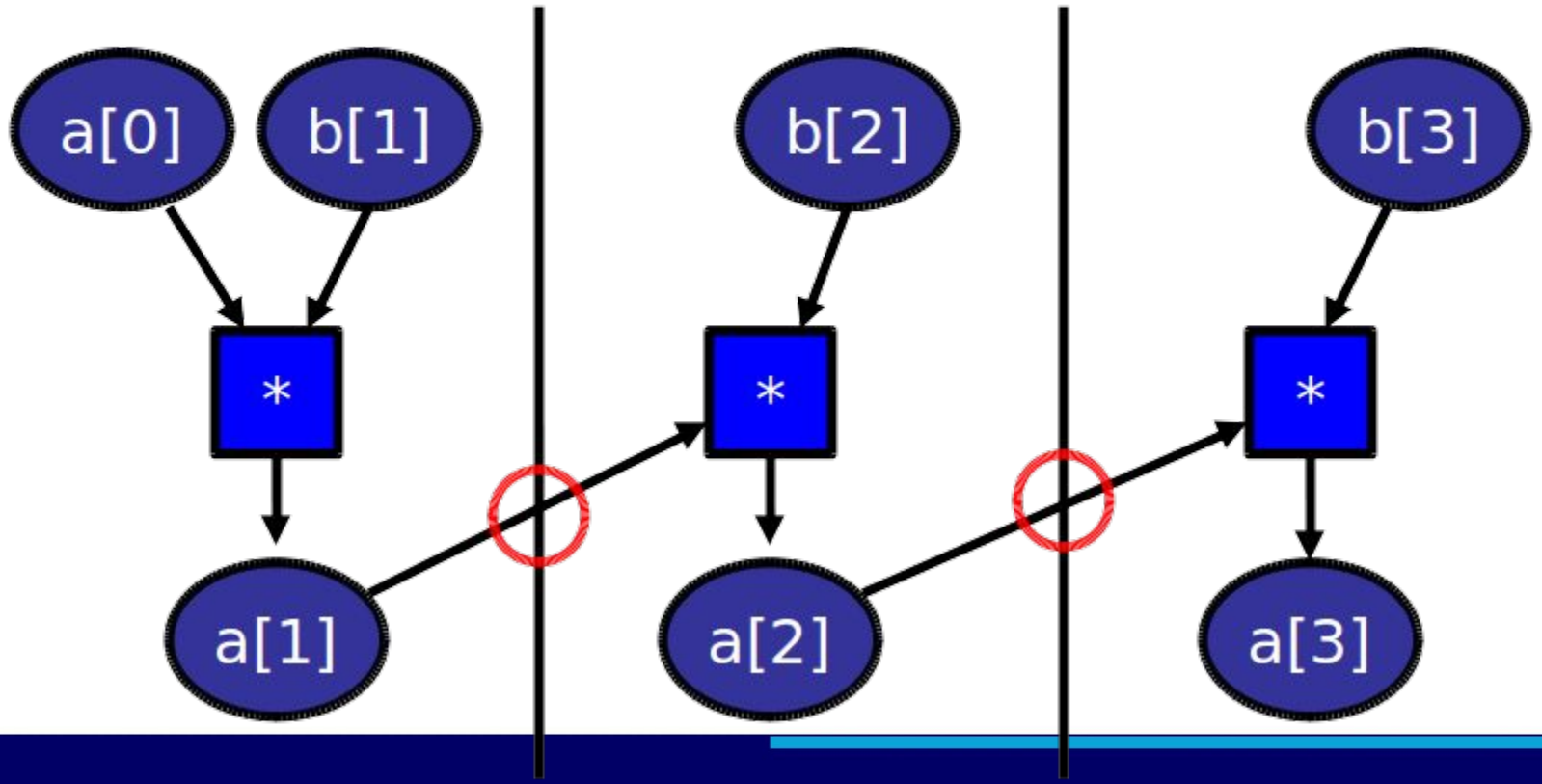
```
for (i = 1; i < 4; i++)
   a[i] = a[i-1] * b[i];
```



Dependence Graph Example #2

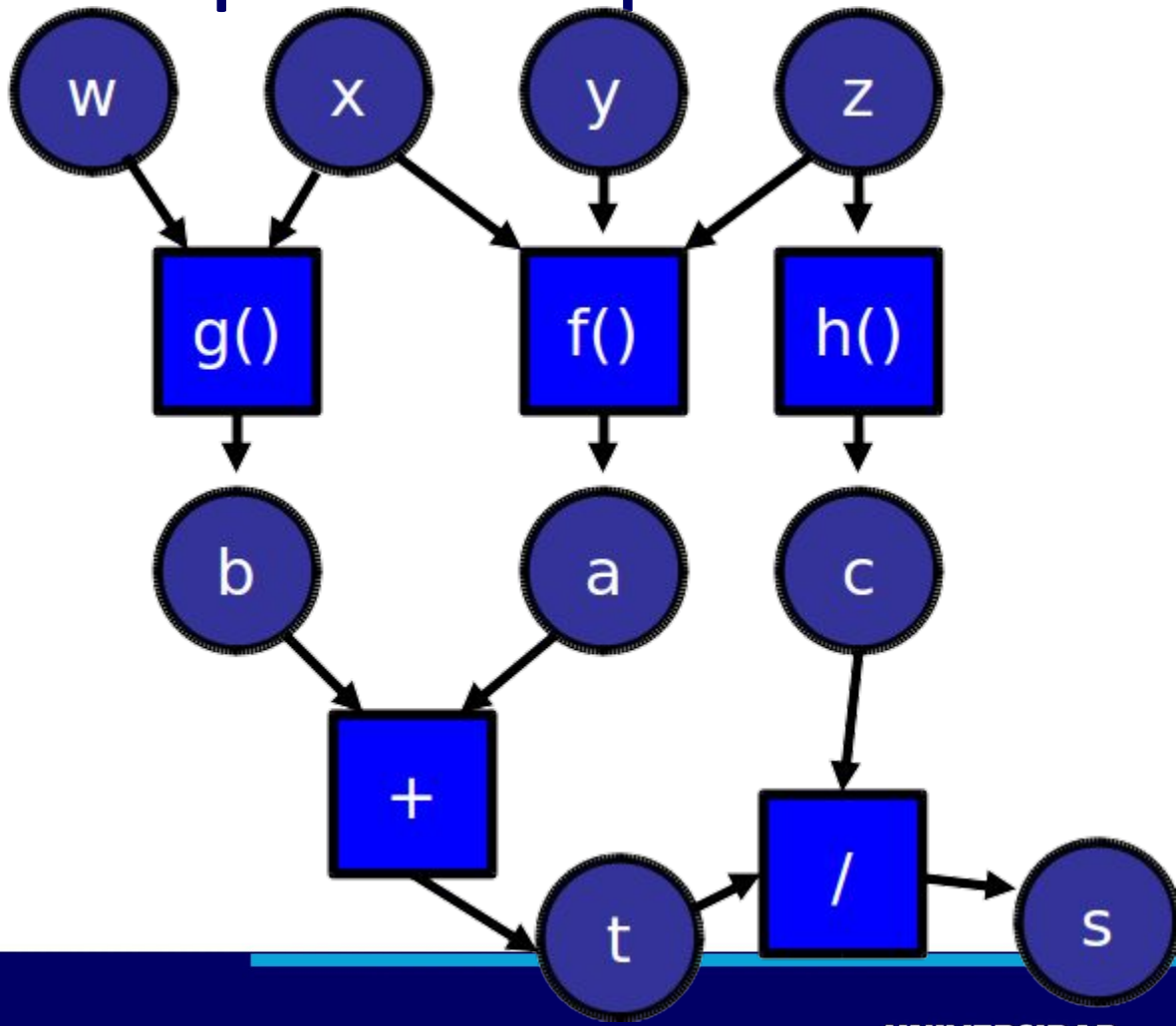
```
for (i = 1; i < 4; i++)
  a[i] = a[i-1] * b[i];
```

No domain decomposition



Dependence Graph Example #3

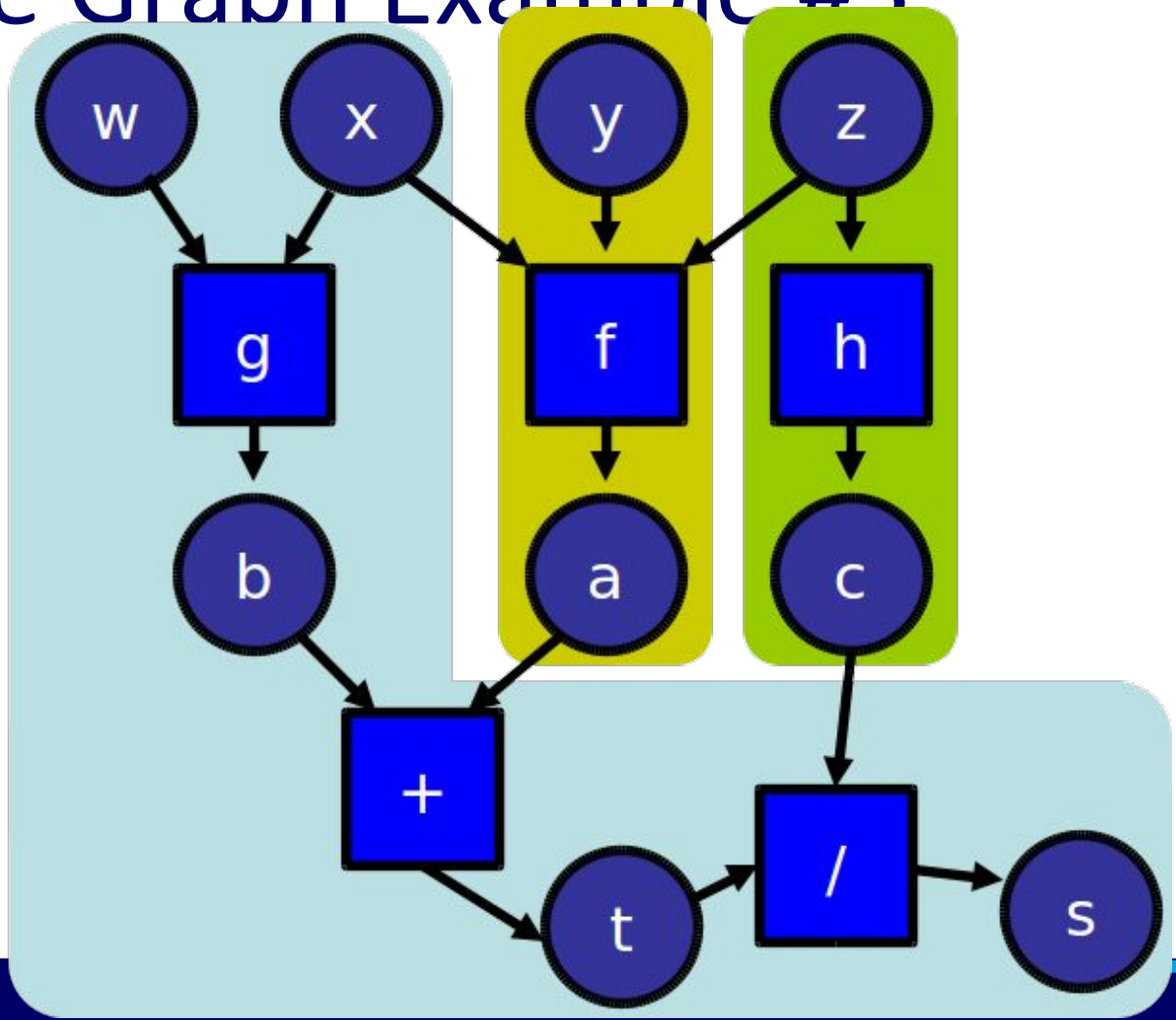
```
a = f(x, y, z);
b = g(w, x);
t = a + b;
c = h(z);
s = t / c;
```



Dependence Graph Example #3

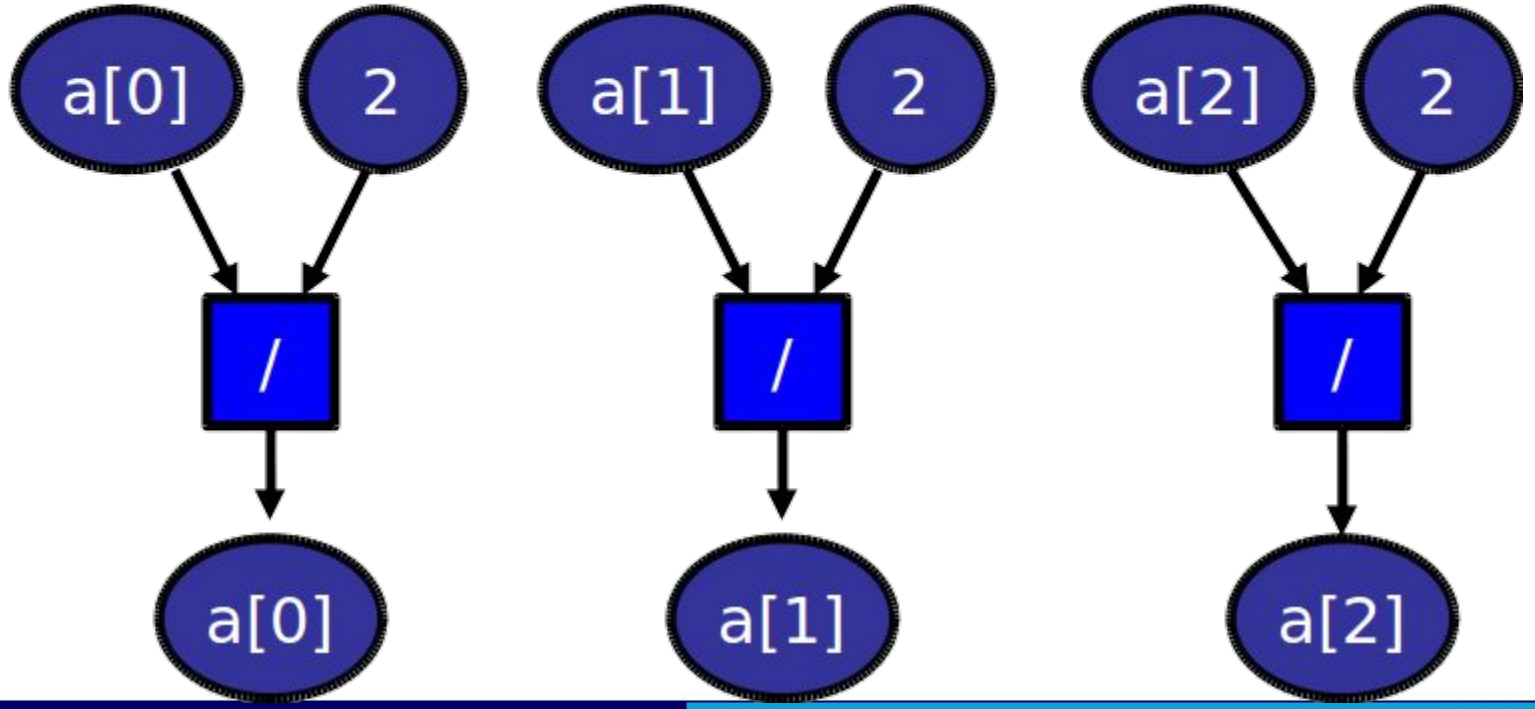
```
a = f(x, y, z);
b = g(w, x);
t = a + b;
c = h(z);
s = t / c;
```

Task decomposition with 3 cores.



Dependence Graph Example #4

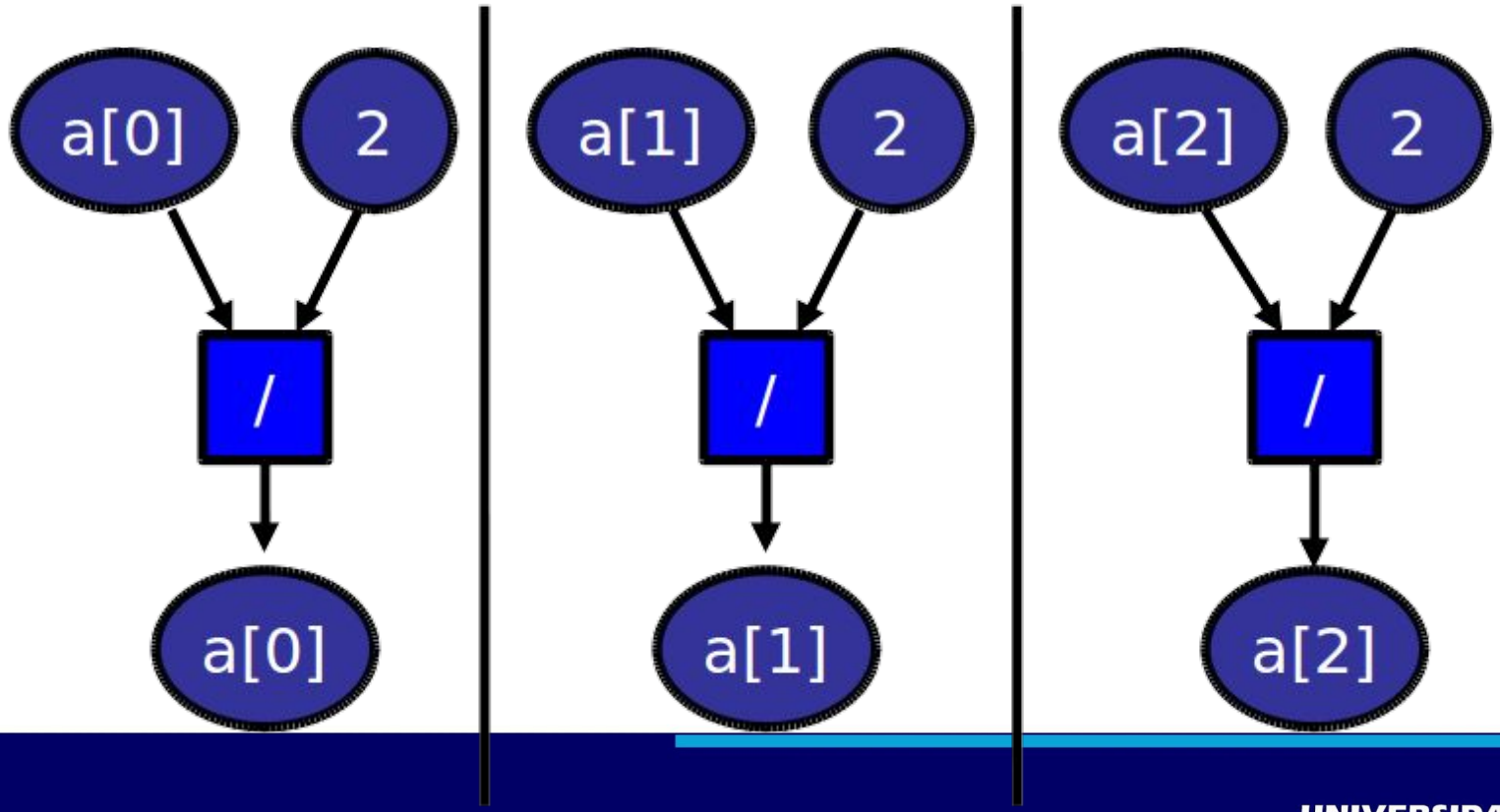
```
for (i = 0; i < 3; i++)
  a[i] = a[i] / 2.0;
```



Dependence Graph Example #4

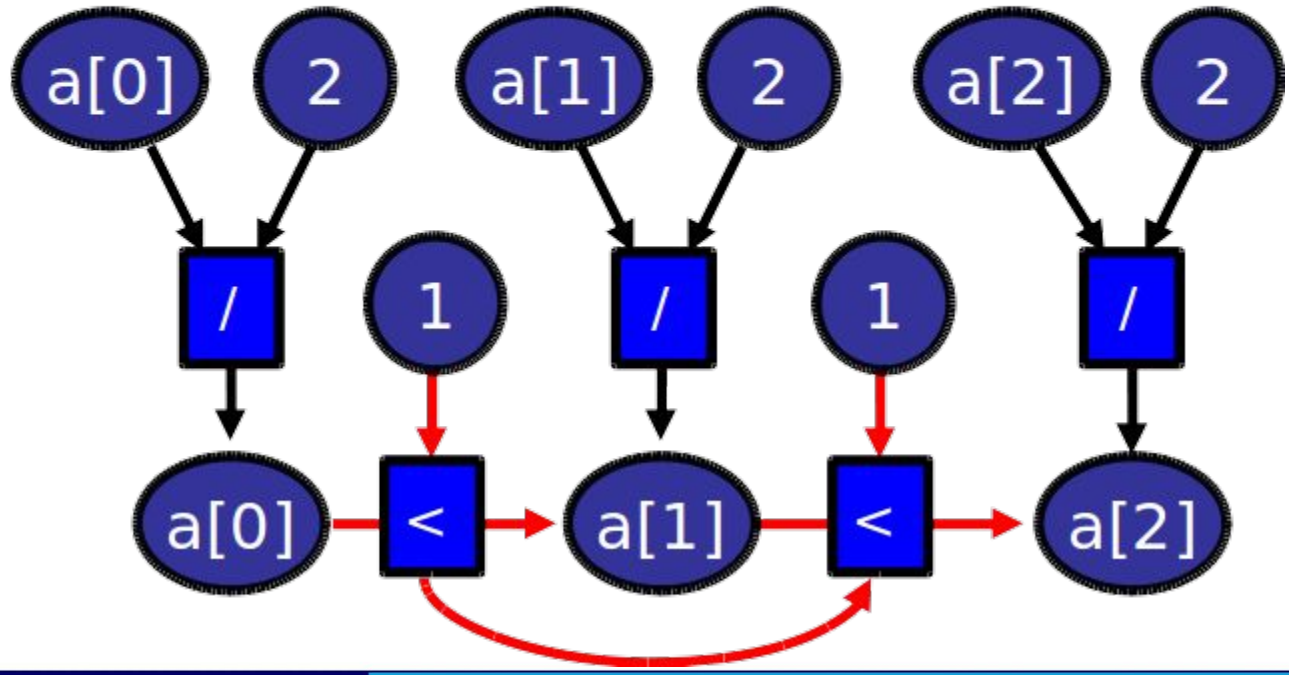
```
for (i = 0; i < 3; i++)
  a[i] = a[i] / 2.0;
```

Domain decomposition



Dependence Graph Example #5

```
for (i = 0; i < 3; i++) {
   a[i] = a[i] / 2.0;
   if (a[i] < 1.0) break;
}
```



Can you find the Parallelism

- Resizing a photo
- Searching a document for all instances of a word
- Updating a spreadsheet
- Compiling a program
- Prefetching pages in a Web browser
- Using a word processor to type a report

Opportunities for a Parallel Solution?

Parallel Solution Easier	Parallel Solution More Difficult or Even Impossible
Larger data sets	Smaller data sets
Dense matrices	Sparse matrices
Dividing space among cores	Dividing time among cores

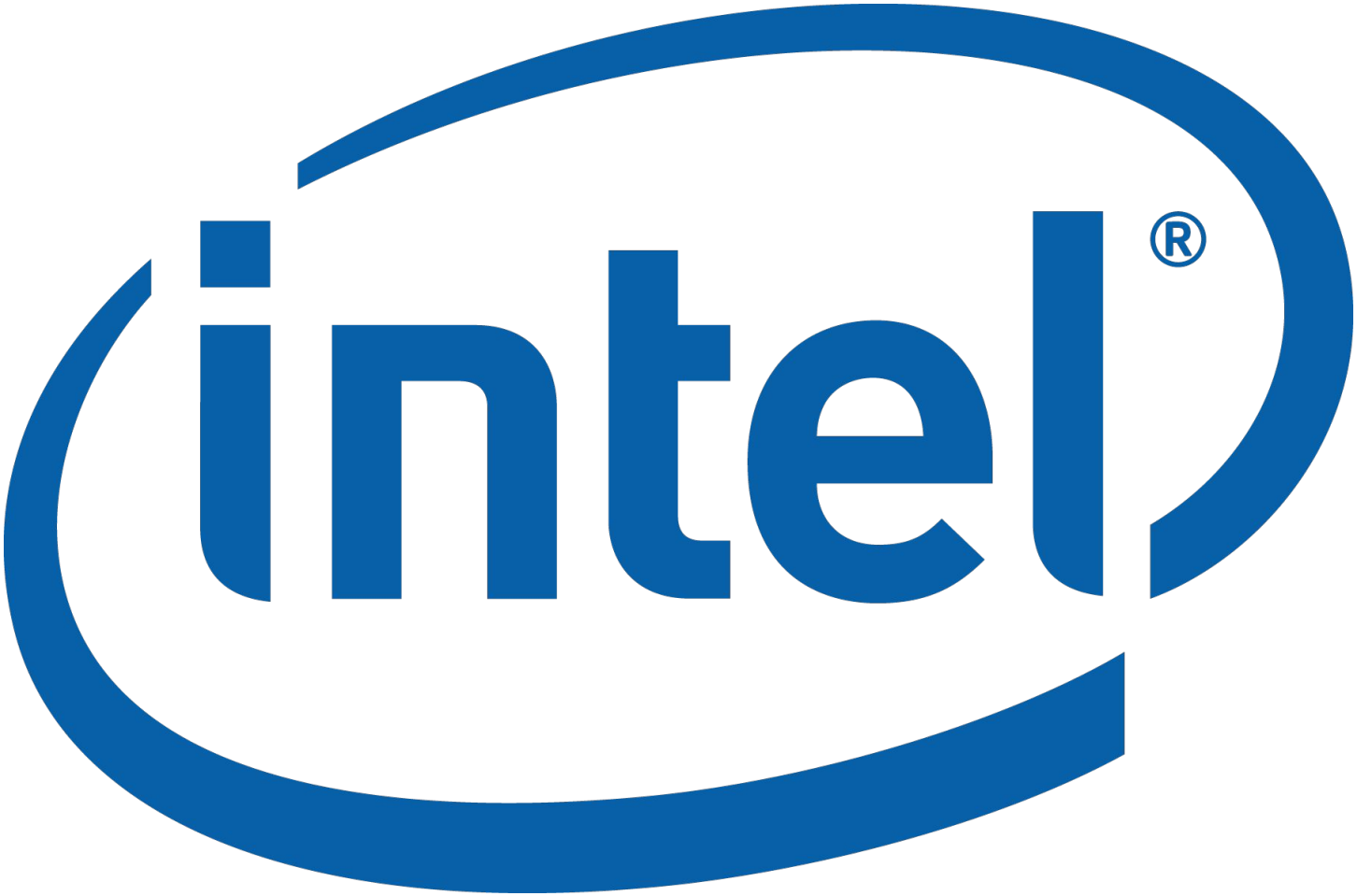
References

Richard H. Carver and Kuo-Chung Tai, *Modern Multithreading: Implementing, Testing, and Debugging Java and C++/Pthreads/ Win32 Programs*, Wiley-Interscience (2006).

Robert L. Mitchell, "Decline of the Desktop," *Computerworld* (September 26, 2005).

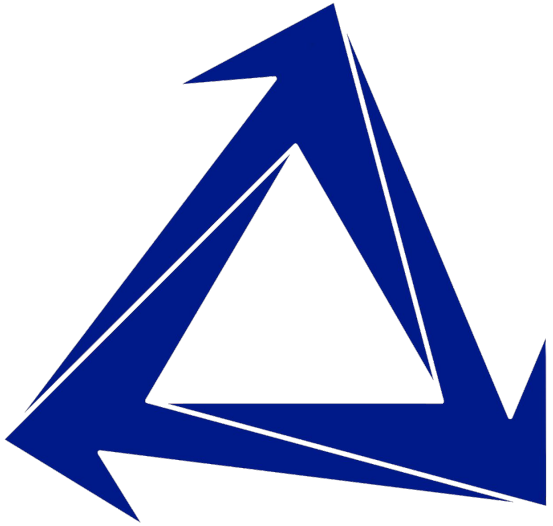
Michael J. Quinn, *Parallel Programming in C with MPI and OpenMP*, McGraw-Hill (2004).

Herb Sutter, "The Free Lunch is Over: A Fundamental Turn Toward Concurrency in Software," *Dr. Dobb's Journal* 30, 3 (March 2005).



Inspira Crea Transforma

UNIVERSIDAD EAFIT



APOLO

Centro de computación científica

Universidad EAFIT