

# Parallel and Distributed Computing

Alberto Paoluzzi – Lecture 10 Julia High Performance: Timing Julia  
& Using Types

Monday 28-03-2022

Chapters 2 & 3 from [Julia High Performance 2nd Edition](#)

## 1 Julia High Performance – 2nd Edition

# Section 1

## Julia High Performance – 2nd Edition

# Julia High Performance – 2nd Edition

## Chapter 2: Timing Julia

### Analyzing Performance

Shows you how to **measure the speed** of **Julia programs** and understand where the **bottlenecks** are. It also shows you **how to measure** the **memory usage** of Julia programs and the **amount of time** spent on **garbage collection**

- 1 The @time macro

## Chapter 2: Timing Julia

### Analyzing Performance

Shows you how to **measure the speed** of **Julia programs** and understand where the **bottlenecks** are. It also shows you **how to measure** the **memory usage** of Julia programs and the **amount of time** spent on **garbage collection**

- 1 The @time macro
- 2 Other time macros

## Chapter 2: Timing Julia

### Analyzing Performance

Shows you how to **measure the speed** of **Julia programs** and understand where the **bottlenecks** are. It also shows you **how to measure** the **memory usage** of Julia programs and the **amount of time** spent on **garbage collection**

- 1 The @time macro
- 2 Other time macros
- 3 The Julia profiler

## Chapter 2: Timing Julia

### Analyzing Performance

Shows you how to **measure the speed** of **Julia programs** and understand where the **bottlenecks** are. It also shows you **how to measure** the **memory usage** of Julia programs and the **amount of time** spent on **garbage collection**

- 1 The @time macro
- 2 Other time macros
- 3 The Julia profiler
- 4 Using the profiler



## Chapter 2: Timing Julia

### Analyzing Performance

Shows you how to **measure the speed** of **Julia programs** and understand where the **bottlenecks** are. It also shows you **how to measure** the **memory usage** of Julia programs and the **amount of time** spent on **garbage collection**

- 1 The @time macro
- 2 Other time macros
- 3 The Julia profiler
- 4 Using the profiler
- 5 Profile View

## Chapter 2: Timing Julia

### Analyzing Performance

Shows you how to **measure the speed** of **Julia programs** and understand where the **bottlenecks** are. It also shows you **how to measure** the **memory usage** of Julia programs and the **amount of time** spent on **garbage collection**

- 1 The @time macro
- 2 Other time macros
- 3 The Julia profiler
- 4 Using the profiler
- 5 Profile View
- 6 Using Juno for profiling

## Chapter 2: Timing Julia

### Analyzing Performance

Shows you how to **measure the speed** of **Julia programs** and understand where the **bottlenecks** are. It also shows you **how to measure** the **memory usage** of Julia programs and the **amount of time** spent on **garbage collection**

- 1 The @time macro
- 2 Other time macros
- 3 The Julia profiler
- 4 Using the profiler
- 5 Profile View
- 6 Using Juno for profiling
- 7 Using TimerOutputs

# Chapter 3: Using Types

## Types, Type Inference, and Stability

One of the principal ways in which Julia achieves its performance goals is by using type information. This chapter describes how the Julia compiler uses type information to create fast machine code. It describes ways of writing Julia code to provide effective type information to the Julia compiler.

- 1 The Julia type system

# Chapter 3: Using Types

## Types, Type Inference, and Stability

One of the principal ways in which Julia achieves its performance goals is by using type information. This chapter describes how the Julia compiler uses type information to create fast machine code. It describes ways of writing Julia code to provide effective type information to the Julia compiler.

- 1 The Julia type system
- 2 Using types

# Chapter 3: Using Types

## Types, Type Inference, and Stability

One of the principal ways in which Julia achieves its performance goals is by using type information. This chapter describes how the Julia compiler uses type information to create fast machine code. It describes ways of writing Julia code to provide effective type information to the Julia compiler.

- 1 The Julia type system
- 2 Using types
- 3 Multiple dispatch

# Chapter 3: Using Types

## Types, Type Inference, and Stability

One of the principal ways in which Julia achieves its performance goals is by using type information. This chapter describes how the Julia compiler uses type information to create fast machine code. It describes ways of writing Julia code to provide effective type information to the Julia compiler.

- 1 The Julia type system
- 2 Using types
- 3 Multiple dispatch
- 4 Abstract types

# Chapter 3: Using Types

## Types, Type Inference, and Stability

One of the principal ways in which Julia achieves its performance goals is by using type information. This chapter describes how the Julia compiler uses type information to create fast machine code. It describes ways of writing Julia code to provide effective type information to the Julia compiler.

- 1 The Julia type system
- 2 Using types
- 3 Multiple dispatch
- 4 Abstract types
- 5 Julia's type hierarchy



# Chapter 3: Using Types

## Types, Type Inference, and Stability

One of the principal ways in which Julia achieves its performance goals is by using type information. This chapter describes how the Julia compiler uses type information to create fast machine code. It describes ways of writing Julia code to provide effective type information to the Julia compiler.

- 1 The Julia type system
- 2 Using types
- 3 Multiple dispatch
- 4 Abstract types
- 5 Julia's type hierarchy
- 6 Composite and immutable types

# Chapter 3: Using Types

## Types, Type Inference, and Stability

One of the principal ways in which Julia achieves its performance goals is by using type information. This chapter describes how the Julia compiler uses type information to create fast machine code. It describes ways of writing Julia code to provide effective type information to the Julia compiler.

- 1 The Julia type system
- 2 Using types
- 3 Multiple dispatch
- 4 Abstract types
- 5 Julia's type hierarchy
- 6 Composite and immutable types
- 7 Type parameters

# Chapter 3: Using Types

## Types, Type Inference, and Stability

One of the principal ways in which Julia achieves its performance goals is by using type information. This chapter describes how the Julia compiler uses type information to create fast machine code. It describes ways of writing Julia code to provide effective type information to the Julia compiler.

- 1 The Julia type system
- 2 Using types
- 3 Multiple dispatch
- 4 Abstract types
- 5 Julia's type hierarchy
- 6 Composite and immutable types
- 7 Type parameters
- 8 Type inference

# Chapter 3: Using Types

## Types, Type Inference, and Stability

One of the principal ways in which Julia achieves its performance goals is by using type information. This chapter describes how the Julia compiler uses type information to create fast machine code. It describes ways of writing Julia code to provide effective type information to the Julia compiler.

- 1 The Julia type system
- 2 Using types
- 3 Multiple dispatch
- 4 Abstract types
- 5 Julia's type hierarchy
- 6 Composite and immutable types
- 7 Type parameters
- 8 Type inference
- 9 Type-stability Definitions

# Chapter 3: Using Types

## Types, Type Inference, and Stability

One of the principal ways in which Julia achieves its performance goals is by using type information. This chapter describes how the Julia compiler uses type information to create fast machine code. It describes ways of writing Julia code to provide effective type information to the Julia compiler.

- 1 The Julia type system
- 2 Using types
- 3 Multiple dispatch
- 4 Abstract types
- 5 Julia's type hierarchy
- 6 Composite and immutable types
- 7 Type parameters
- 8 Type inference
- 9 Type-stability Definitions
- 10 Fixing type instability

# Chapter 3: Using Types

## Types, Type Inference, and Stability

One of the principal ways in which Julia achieves its performance goals is by using type information. This chapter describes how the Julia compiler uses type information to create fast machine code. It describes ways of writing Julia code to provide effective type information to the Julia compiler.

- 1 The Julia type system
- 2 Using types
- 3 Multiple dispatch
- 4 Abstract types
- 5 Julia's type hierarchy
- 6 Composite and immutable types
- 7 Type parameters
- 8 Type inference
- 9 Type-stability Definitions
- 10 Fixing type instability
- 11 The performance pitfalls

# Chapter 3: Using Types

## Types, Type Inference, and Stability

One of the principal ways in which Julia achieves its performance goals is by using type information. This chapter describes how the Julia compiler uses type information to create fast machine code. It describes ways of writing Julia code to provide effective type information to the Julia compiler.

- 1 The Julia type system
- 2 Using types
- 3 Multiple dispatch
- 4 Abstract types
- 5 Julia's type hierarchy
- 6 Composite and immutable types
- 7 Type parameters
- 8 Type inference
- 9 Type-stability Definitions
- 10 Fixing type instability
- 11 The performance pitfalls
- 12 Identifying type stability

# Chapter 3: Using Types

## Types, Type Inference, and Stability

One of the principal ways in which Julia achieves its performance goals is by using type information. This chapter describes how the Julia compiler uses type information to create fast machine code. It describes ways of writing Julia code to provide effective type information to the Julia compiler.

- 1 The Julia type system
- 2 Using types
- 3 Multiple dispatch
- 4 Abstract types
- 5 Julia's type hierarchy
- 6 Composite and immutable types
- 7 Type parameters
- 8 Type inference
- 9 Type-stability Definitions
- 10 Fixing type instability
- 11 The performance pitfalls
- 12 Identifying type stability
- 13 Loop variables



# Chapter 3: Using Types

## Types, Type Inference, and Stability

One of the principal ways in which Julia achieves its performance goals is by using type information. This chapter describes how the Julia compiler uses type information to create fast machine code. It describes ways of writing Julia code to provide effective type information to the Julia compiler.

- 1 The Julia type system
- 2 Using types
- 3 Multiple dispatch
- 4 Abstract types
- 5 Julia's type hierarchy
- 6 Composite and immutable types
- 7 Type parameters
- 8 Type inference
- 9 Type-stability Definitions
- 10 Fixing type instability
- 11 The performance pitfalls
- 12 Identifying type stability
- 13 Loop variables
- 14 Kernel methods and function barriers