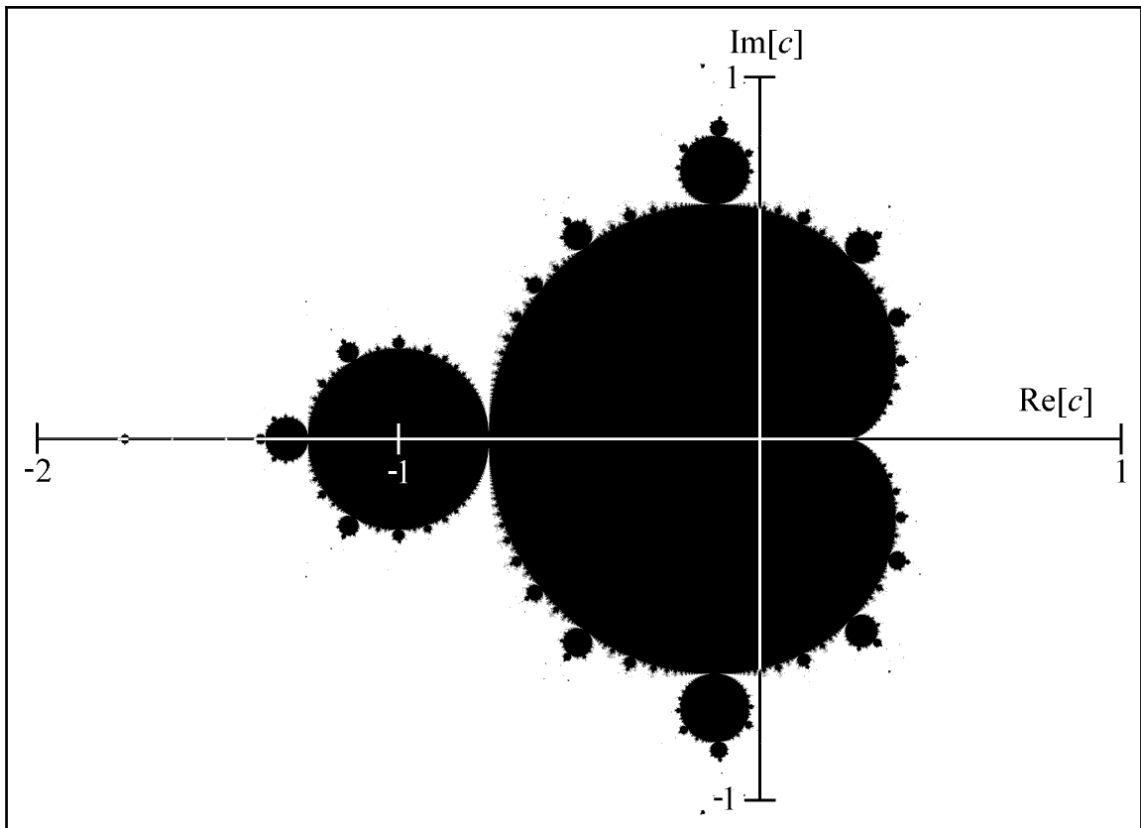
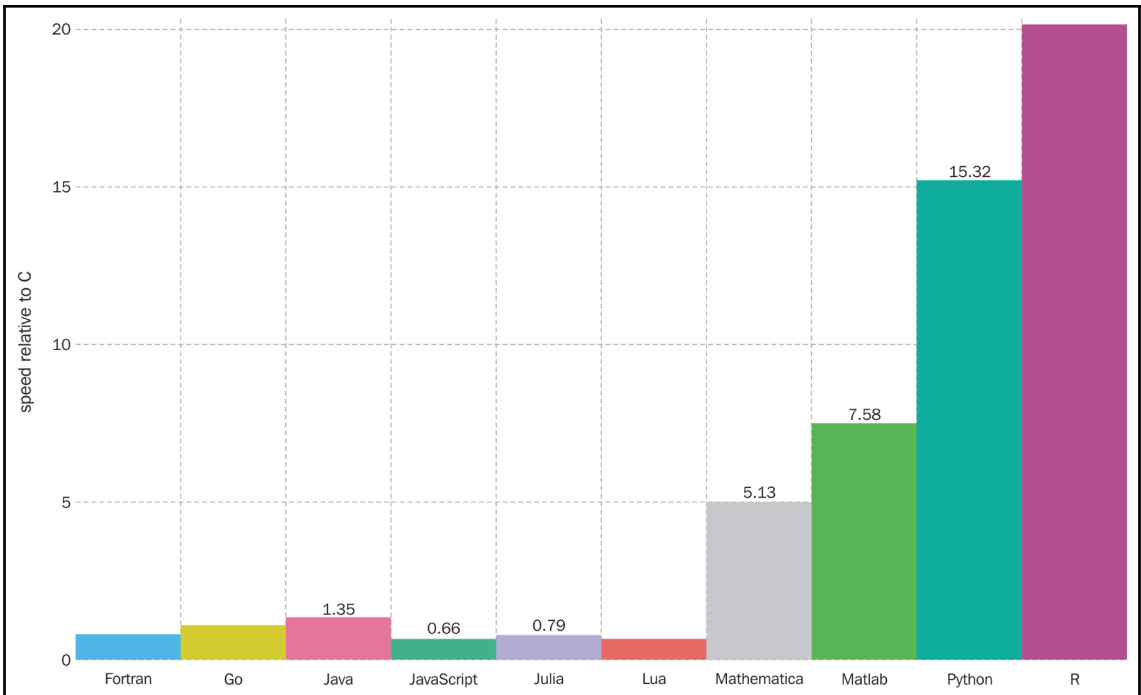


Graphic Bundle

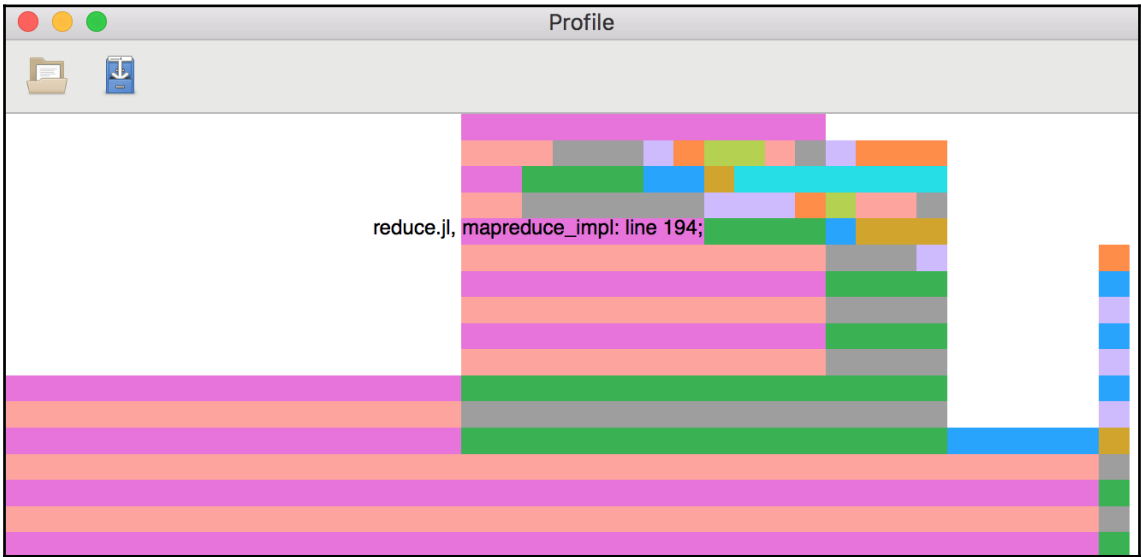
Chapter 1: Julia is Fast

$$f_c(z) = z^2 + c$$





Chapter 2: Analyzing Performance



```
In [9]: using ProfileView

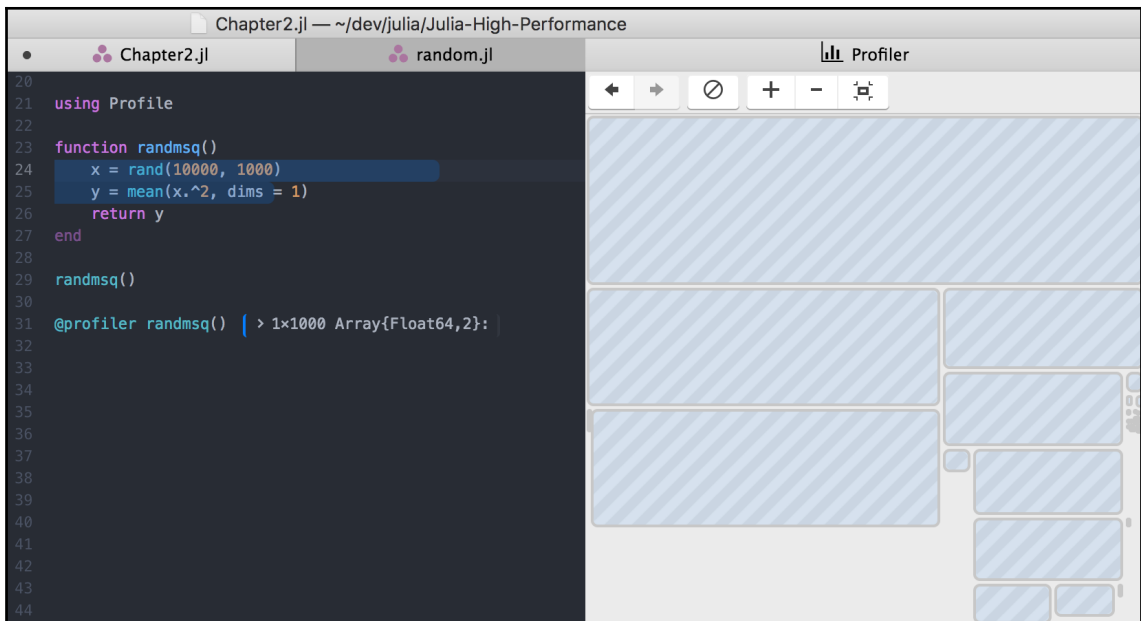
In [10]: @profile testfunc()
Out[10]: 1x1000 Array{Float64,2}:
 0.289969 0.290125 0.288434 0.288483 ... 0.28958 0.286604 0.288659

In [11]: ProfileView.view()
Out[11]:
```

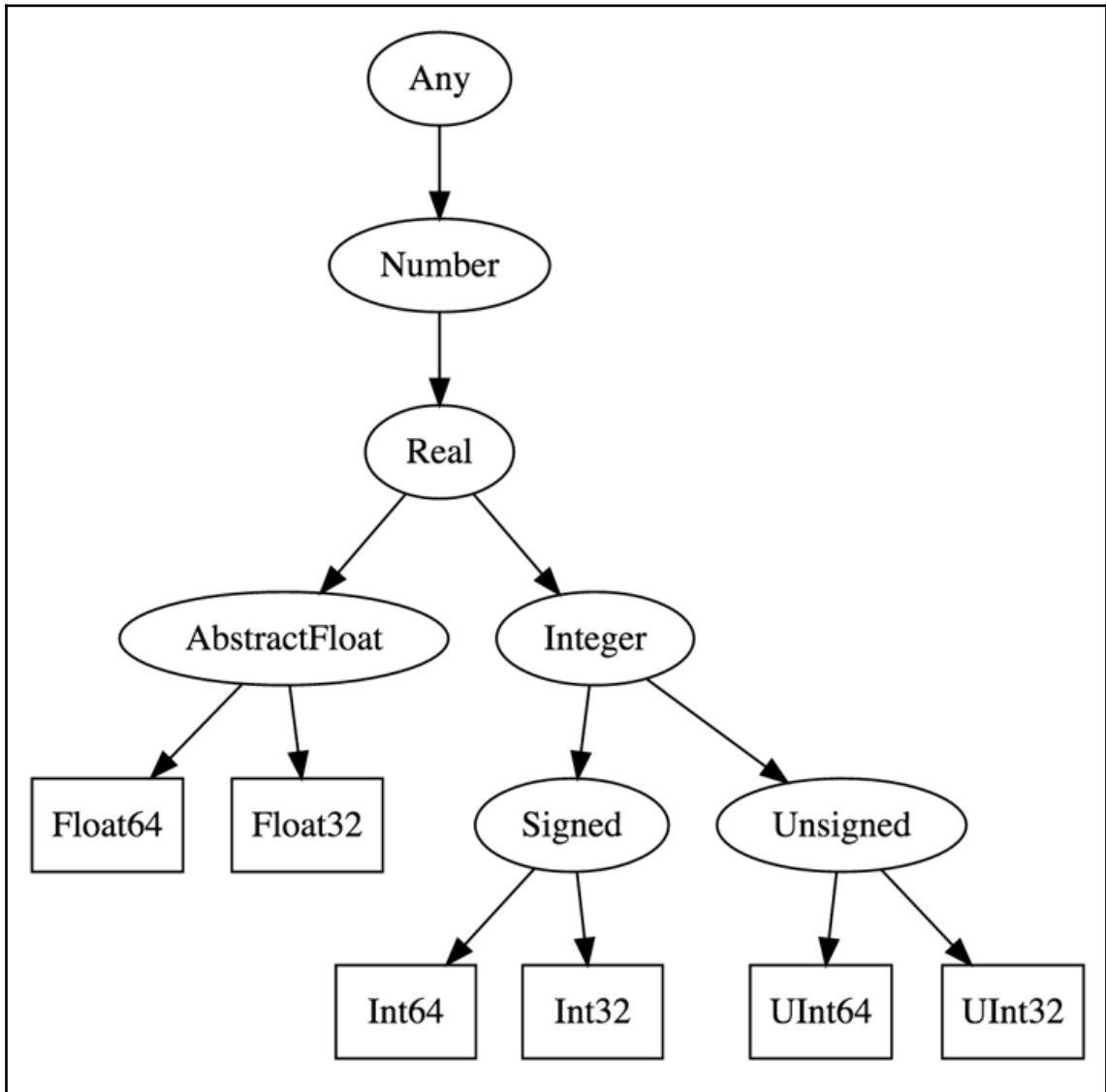
Profile results

dsfmt_fill_array_close_open! in dSFMT.jl:84	#var#534 in #var#534 in sta..
rand! in random.jl:433	#var in <missing>;:0
testfunc in In[1]:2	testfunc in In[1]:3
macro expansion in profile.jl:23; anonymous in <missing>;:-1	rand in random.jl:371; rand
include_string in loading.jl:515	
include_string in Compat.jl:407	
execute_request in execute_request.jl:154	
eventloop in eventloop.jl:8	
#14 in task.jl:335	

Function: dsfmt_fill_array_close_open! in ./dSFMT.jl:84



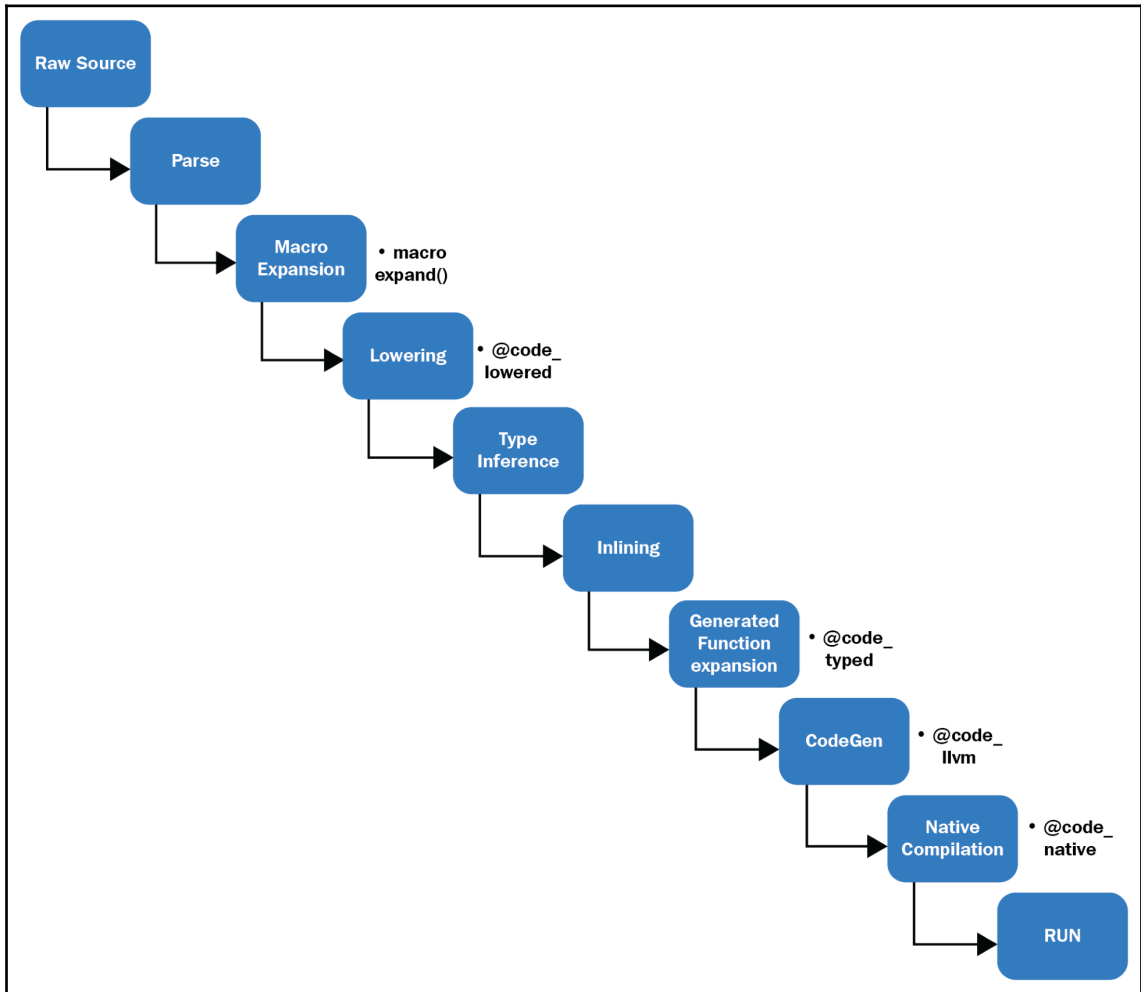
Chapter 3: Types, Type Inference, and Stability



```
julia> @code_warntype pos(2.5)
Body::Union{Float64, Int64}
1 - %1 = (Base.lt_float)(x, 0.0)::Bool
   |   %2 = (Base.eq_float)(x, 0.0)::Bool
   |   %3 = (Base.and_int)(%2, true)::Bool
   |   %4 = (Base.and_int)(%3, false)::Bool
   |   %5 = (Base.or_int)(%1, %4)::Bool
   |   goto #3 if not %5
2 -     return 0
3 -     return x
```

```
[julia> @code_warntype pos_fixed(2.5)
Body::Float64
1 - %1 = (Base.lt_float)(x, 0.0)::Bool
   |   %2 = (Base.eq_float)(x, 0.0)::Bool
   |   %3 = (Base.and_int)(%2, true)::Bool
   |   %4 = (Base.and_int)(%3, false)::Bool
   |   %5 = (Base.or_int)(%1, %4)::Bool
   |   goto #3 if not %5
2 -     return 0.0
3 -     return x
```

Chapter 4: Making Fast Function Calls



$$p(x) = \sum_{i=0}^n a_i x^i = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \dots + a_n x^n$$

$$f(x) = 1 + 2x + 3x^2 + 4x^3 + 5x^4 + 6x^5 + 7x^6 + 8x^7 + 9x^8$$

$$b_n = a_n$$

$$b_{n-1} = a_{n-1} + b_n x$$

$$b_{n-2} = a_{n-2} + b_{n-1} x$$

⋮

$$b_0 = a_0 + b_1 x$$

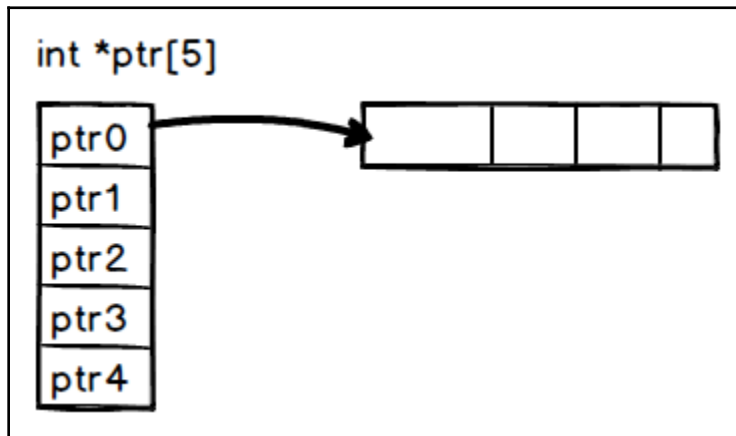
Chapter 5: Fast Numbers

$$O(\sqrt{n})$$

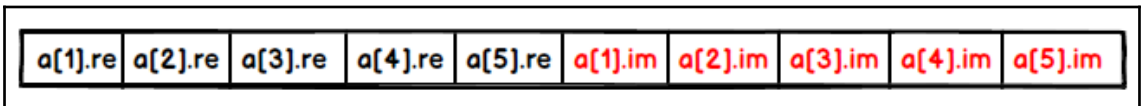
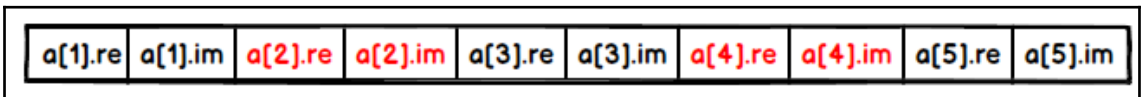
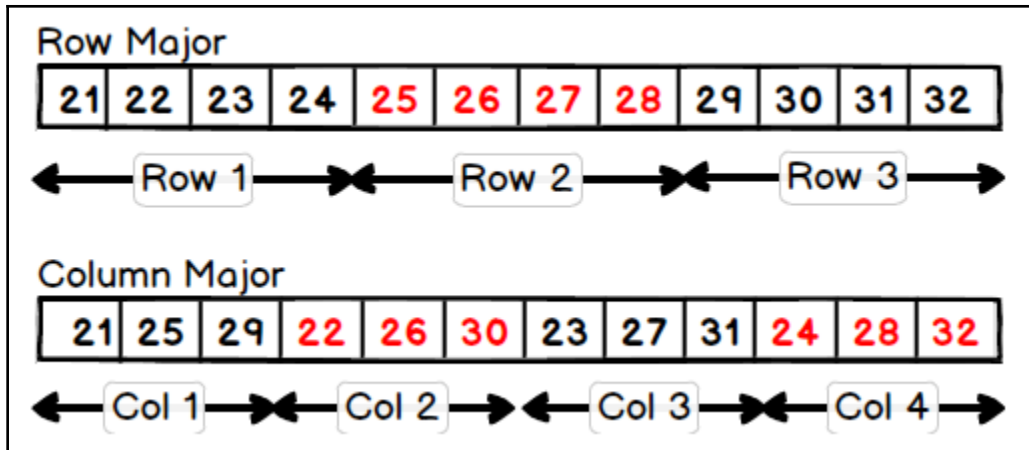
$$O\left(\sqrt{\log(n)}\right)$$

Chapter 6: Using Arrays

Index	1	2	3	4	5	6
Value	34	55	63	23	45	11
Address	1000	1004	1008	1012	1016	1020



		Columns			
		1	2	3	4
Rows	1	21	22	23	24
	2	25	26	27	28
	3	29	30	31	32



Chapter 7: Accelerating Code with the GPU

```
jrun@notebook-jg7xj:/home/jrun$ nvidia-smi
Mon May 20 22:16:40 2019
```

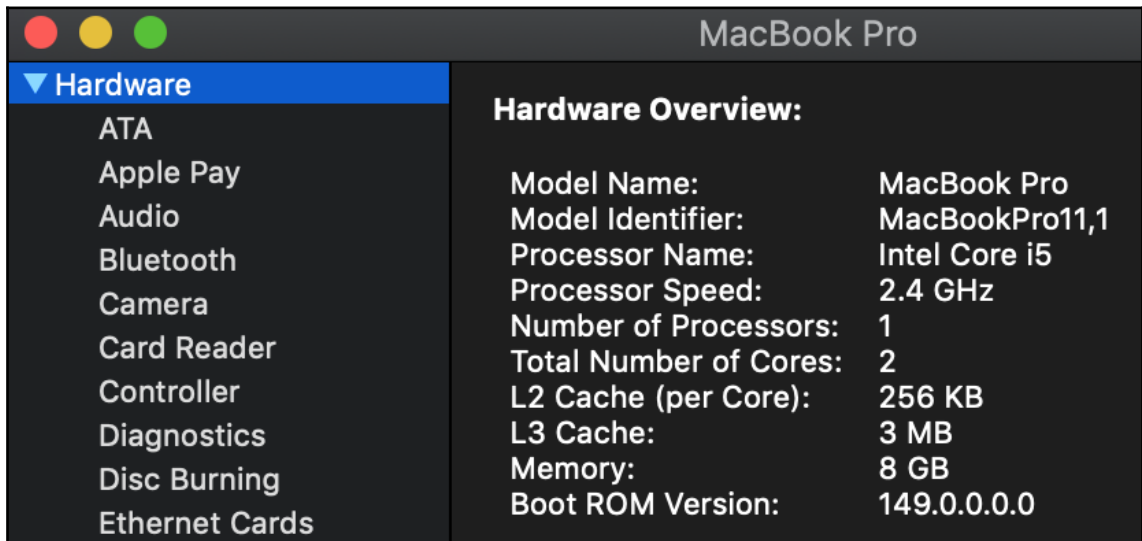
NVIDIA-SMI 396.26		Driver Version: 396.26				
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr. ECC
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute M.
0	Tesla K80	Off	0000D718:00:00.0	Off		0
N/A	51C	P0	87W / 149W	9222MiB / 11441MiB	93%	Default

```
Processes:
```

GPU	PID	Type	Process name	GPU Memory Usage
-----	-----	------	--------------	------------------

```
jrun@notebook-jg7xj:/home/jrun$
```

Chapter 9: Threads



Task Manager

File Options View

Processes Performance App history Start-up Users Details Services

CPU
5% 2.72 GHz

Memory
3.3/15.9 GB (21%)

Disk 0 (C:)
0%

WiFi
S: 0 R: 0 Kbps

Bluetooth PAN
Not connected

GPU 0
Intel(R) HD Graphics 620
1%

CPU Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz

% Utilisation 100%

60 seconds 0

Utilisation	Speed	Base speed:	2.90 GHz
5%	2.72 GHz	Sockets:	1
Processes	Threads	Cores:	2
164	2090	Logical processors:	4
Handles		Virtualisation:	Enabled
66915		L1 cache:	128 KB
Up time		L2 cache:	512 KB
2:22:53:12		L3 cache:	4.0 MB

^ Fewer details | 🛑 Open Resource Monitor

```

1 [|||||] Tasks: 393, 1442 thr; 3 running
2 [|||||] Load average: 2.17 1.79 1.77
3 [|||||] Uptime: 48 days, 04:39:01
4 [|||||]
Mem [|||||]
Swp [|||||]

```

Chapter 10: Distributed Computing with Julia

