

Title: Numerical Methods Lecture - 230207

Speakers: Erik Schnetter

Collection: Numerical Methods (2022/2023)

Date: February 07, 2023 - 9:15 AM

URL: <https://pirsa.org/23020001>

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 2.9K/18 0.5K/75 US Tue Feb 7 09:24

symmetry.pi.local JupyterLab Mem:2.11 GB

File Edit View Run Kernel Git Tabs Settings Help

Differential Equations Project X Wave Equation 2023.ipynb X Performance 2023.ipynb X

Code git Julia 1.8.5

Profiling

```
[1]: # Silly, slow example
```

```
[2]: A = randn(10, 10);
```

```
[3]: for i in 1:1000
      A = A * A
      A /= maximum(A)
    end
```

```
[ ]: |
```

Simple 0 s 6 Julia 1.8.5 | Idle Mem: 2.11 GB Saving completed Mode: Edit Ln 1, Col 1 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 4.0K/18.0KB/s Tue Feb 7 09:25

symmetry.pi.local

JupyterLab Mem:2.18 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

File Edit View Run Kernel Git Tabs Settings Help

Code git Julia 1.8.5

Profiling

```
[1]: # Silly, slow example
```

```
[4]: A = randn(30, 30);
```

```
[7]: for i in 1:1000000
      A = A * A
      A /= maximum(A)
    end
```

```
[ ]:
```

Simple 0 s 6 Julia 1.8.5 | Idle Mem: 2.18 GB Saving completed Mode: Command Ln 1, Col 1 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 2.74KB/s 29.4KB/s Tue Feb 7 09:25

symmetry.pi.local JupyterLab Mem:2.24 GB

File Edit View Run Kernel Git Tabs Settings Help

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

Profiling

```
[1]: # Silly, slow example
```

```
[9]: using Profiling
```

```
ArgumentError: Package Profiling not found in current path.
- Run `import Pkg; Pkg.add("Profiling")` to install the Profiling package.
```

```
Stacktrace:
 [1] macro expansion
       @ ./loading.jl:1163 [inlined]
 [2] macro expansion
       @ ./lock.jl:223 [inlined]
 [3] require(into::Module, mod::Symbol)
       @ Base ./loading.jl:1144
```

```
[4]: A = randn(30, 30);
```

```
[8]: @time for i in 1:1000000
      A = A * A
      A /= maximum(A)
end
```

```
4.842915 seconds (3.00 M allocations: 13.724 GiB, 6.88% gc time)
```

```
[ ]:
```

Simple 0 s 6 Julia 1.8.5 | Idle Mem: 2.24 GB Saving completed Mode: Command Ln 1, Col 16 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 2.1K/28.7KB/s Tue Feb 7 09:25

symmetry.pi.local

JupyterLab Mem:2.26 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

File Edit View Run Kernel Git Tabs Settings Help Code git Julia 1.8.5

Profiling

```
[1]: # Silly, slow example
```

```
[10]: using Profile
```

```
[4]: A = randn(30, 30);
```

```
[8]: @time for i in 1:1000000
      A = A * A
      A /= maximum(A)
    end
```

4.842915 seconds (3.00 M allocations: 13.724 GiB, 6.88% gc time)

```
[ ]:
```

Simple 0 s 6 Julia 1.8.5 | Idle Mem: 2.26 GB Saving completed Mode: Edit Ln 1, Col 3 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 0.24KB/s 10.7KB/s Tue Feb 7 09:26

symmetry.pi.local JupyterLab Mem:2.28 GB

File Edit View Run Kernel Git Tabs Settings Help

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

Profiling

```
[1]: # Silly, slow example
```

```
[10]: using Profile
```

```
[4]: A = randn(30, 30);
```

```
[8]: @profile for i in 1:1000000
      A = A * A
      A /= maximum(A)
    end
```

4.842915 seconds (3.00 M allocations: 13.724 GiB, 6.88% gc time)

```
[ ]:
```

Simple 0 s 6 Julia 1.8.5 | Idle Mem: 2.28 GB Saving completed Mode: Edit Ln 1, Col 9 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom 2.7KB/s 14.5KB/s Charged 2.7KB/s 14.5KB/s Tue Feb 7 09:29

symmetry.pi.local

JupyterLab

File Edit View Run Kernel Git Tabs Settings Help Mem:2.28 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb X

Code git Julia 1.8.5

```
[4]: A = randn(30, 30);
```

```
[11]: @profile for i in 1:1000000
      A = A * A
      A /= maximum(A)
      end
```

```
[13]: Profile.print()
```

```
Overhead | [+additional indent] Count File:Line; Function
=====
```

```
1|1  @Base/arraymath.jl:24; /(A::Matrix{Float64}, B::Float64)
3|3  @Base/reduce.jl:0; mapreduce_impl(f::typeof(identi...
5718 @Base/task.jl:484; (::IJulia.var"#15#18"())
5718 ...lia/src/eventloop.jl:8; eventloop(socket::ZMQ.Socket)
5718 @Base/essentials.jl:726; invokelatest
5718 @Base/essentials.jl:729; #invokelatest#2
5718 ...execute_request.jl:67; execute_request(socket::ZMQ.S...
5718 ...SoftGlobalScope.jl:65; softscope_include_string(m:...
5718 @Base/loading.jl:1428; include_string(mapexpr::typ...
2| 5718 @Base/boot.jl:368; eval
1| 1 ...le/src/Profile.jl:0; top-level scope
1| 5715 ...le/src/Profile.jl:27; top-level scope
44| 2351 In[11]:2; macro expansion
1| 2307 ...a/src/matmul.jl:148; *(A::Matrix{Float64}, B:...
738| 738 @Base/array.jl:378; similar
738| 738 @Base/boot.jl:469; Array
738| 738 @Base/boot.jl:461; Array
1568 ...a/src/matmul.jl:276; mul!
1568 ...a/src/matmul.jl:161; mul!
```

Simple 0 6 Julia 1.8.5 | Idle Mem: 2.28 GB Saving completed Mode: Command Ln 1, Col 14 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 2.5KB/s 11.5KB/s Tue Feb 7 09:30

symmetry.pi.local

JupyterLab

Mem:2.20 GB

File Edit View Run Kernel Git Tabs Settings Help

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb X

Code git Julia 1.8.5

```

1      ...e/range.jl:400; unirange_last
384    ...e/reduce.jl:658; mapreduce_impl(f::t...
384    @Base/math.jl:760; max
80     @Base/bool.jl:154; <
40     @Base/bool.jl:35; !
40     @Base/bool.jl:38; &
260    ...entials.jl:489; ifelse
44     ...oatfuncs.jl:15; signbit
99     ...e/reduce.jl:659; mapreduce_impl(f::t...
99     @Base/range.jl:883; iterate
99     ...romotion.jl:477; ==
1      ...e/reduce.jl:664; mapreduce_impl(f::t...
1      ...e/reduce.jl:630; isbadzero
1      @Base/float.jl:410; ==
1      @Base/range.jl:883; iterate
1      @Base/promotion.jl:477; ==

```

Total snapshots: 5751. Utilization: 100% across all threads and tasks. Use the `groupby` kwarg to break down by thread and/or task

[]:

Simple 0 s 6 Julia 1.8.5 | Idle Mem: 2.20 GB Saving completed Mode: Command Ln 1, Col 14 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 2.64KB/s 17.2KB/s Tue Feb 7 09:34

symmetry.pi.local

JupyterLab Mem:2.20 GB

File Edit View Run Kernel Git Tabs Settings Help

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```
[1]: # Silly, slow example
```

```
[10]: using Profile
```

```
[4]: A = randn(30, 30);
```

```
[*]: @time for i in 1:1000000
      A = A * A
      A /= maximum(A)
      end
```

```
[11]: @profile for i in 1:1000000
      A = A * A
      A /= maximum(A)
      end
```

```
[13]: Profile.print()
```

```
Overhead | [+additional indent] Count File:Line; Function
-----|-----
```

```
1|1  @Base/arraymath.jl:24; /(A::Matrix{Float64}, B::Float64)
3|3  @Base/reduce.jl:0; mapreduce_impl(f::typeof(identi...
5718| @Base/task.jl:484; (::IJulia.var"#15#18")()
5718| ...lia/src/eventloop.jl:8; eventloop(socket::ZMQ.Socket)
5718| @Base/essentials.jl:726; invokelatest
5718| @Base/essentials.jl:729; #invokelatest#2
5718| ...execute_request.jl:67; execute_request(socket::ZMQ.S...
5718| ...SoftGlobalScope.jl:65; softscope_include_string(m:...
5718| @Base/loading.jl:1428; include_string(mapexpr::typ
```

Simple 0 s 6 Julia 1.8.5 | Busy Mem: 2.20 GB Saving completed Mode: Command Ln 1, Col 1 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 6.0KB/s 4.0KB/s Tue Feb 7 09:35

symmetry.pi.local

JupyterLab Mem:2.24 GB

File Edit View Run Kernel Git Tabs Settings Help

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb X

Code git Julia 1.8.5

```
[1]: # Silly, slow example
```

```
[10]: using Profile
```

```
[4]: A = randn(30, 30);
```

```
[14]: @time for i in 1:1000000
        A = A * A
        A /= maximum(A)
    end
4.383367 seconds (3.00 M allocations: 13.724 GiB, 6.05% gc time)
```

```
[11]: @profile for i in 1:1000000
        A = A * A
        A /= maximum(A)
    end
```

```
[13]: Profile.print()
```

```
Overhead | [+additional indent] Count File:Line; Function
=====
```

```
1|1  @Base/arraymath.jl:24; /(A::Matrix{Float64}, B::Float64)
3|3  @Base/reduce.jl:0; mapreduce_impl(f::typeof(identi...
5718 @Base/task.jl:484; (::IJulia.var"#15#18")()
5718 ...lia/src/eventloop.jl:8; eventloop(socket::ZMQ.Socket)
5718 @Base/essentials.jl:726; invokelatest
5718 @Base/essentials.jl:729; #invokelatest#2
5718 ...execute_request.jl:67; execute_request(socket::ZMQ.S...
```

Simple 0 6 Julia 1.8.5 | Idle Mem: 2.24 GB Saving completed Mode: Command Ln 1, Col 6 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom 0.5KB/s 4.7KB/s Charged Tue Feb 7 09:36

symmetry.pi.local JupyterLab Mem:2.35 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```
1 | 1 @Base/promotion.jl:477; ==
Total snapshots: 5751. Utilization: 100% across all threads and tasks. Use the `groupby` kwarg to break down by thread and/or task
```

```
[*]: # Manual memory management
@time for i in 1:1000000
    A = A * A
    A ./= maximum(A)
end
```

```
[ ]:
```

Simple 0 s 6 Julia 1.8.5 | Busy Mem: 2.35 GB Saving completed Mode: Edit Ln 1, Col 1 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom 2.2K/16.7KB/s Charged Tue Feb 7 09:37
symmetry.pi.local
JupyterLab Mem:2.39 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```
1 | 1 @Base/promotion.jl:477; ==  
Total snapshots: 5751. Utilization: 100% across all threads and tasks. Use the `groupby` kwarg to break down by thread and/or task
```

```
[16]: # Manual memory management  
@time for i in 1:1000000  
    A = A * A  
    A |= maximum(A)  
end  
4.881326 seconds (3.00 M allocations: 13.724 GiB, 5.95% gc time)
```

```
[ ]:
```

Simple 0 s 6 Julia 1.8.5 | Idle Mem: 2.39 GB Saving completed Mode: Edit Ln 4, Col 7 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 6.74KB/s 30.5KB/s Tue Feb 7 09:37

symmetry.pi.local JupyterLab Mem:2.45 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```
1 | 1 @Base/promotion.jl:477; ==
Total snapshots: 5751. Utilization: 100% across all threads and tasks. Use the `groupby` kwarg to break down by thread and/or task

[*]: # Manual memory management
@time for i in 1:1000000
    A = A * A
    m = maximum(A)
    A /= m
end

[]:
```

Simple 0 s 6 Julia 1.8.5 | Busy Mem: 2.45 GB Saving completed Mode: Command Ln 1, Col 1 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom 5.5KB/s 16.0KB/s Charged 5.5KB/s 16.0KB/s Tue Feb 7 09:37

symmetry.pi.local

JupyterLab Mem:2.46 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```
1 | 1 @Base/promotion.jl:477; ==
Total snapshots: 5751. Utilization: 100% across all threads and tasks. Use the `groupby` kwarg to break down by thread and/or task

[18]: # Manual memory management
@time for i in 1:1000000
    A = A * A
    m = maximum(A)
    A /= m
end

4.716019 seconds (3.00 M allocations: 13.724 GiB, 5.79% gc time)

[ ]:
```

Simple 0 s 6 Julia 1.8.5 | Idle Mem: 2.46 GB Saving completed Mode: Command Ln 1, Col 1 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom 6.5K/26.0KB/s Charged Tue Feb 7 09:40

symmetry.pi.local Linear Algebra - The Julia Language Mem:2.36 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```

200 ...e/reduce.jl:659; mapreduce_impl(f::t...
44 ...oatfuncs.jl:15; signbit
99 @Base/range.jl:883; iterate
99 ...romotion.jl:477; ==
1 ...e/reduce.jl:664; mapreduce_impl(f::t...
1 ...e/reduce.jl:630; isbadzero
1 @Base/float.jl:410; ==
1 @Base/range.jl:883; iterate
1 @Base/promotion.jl:477; ==

```

Total snapshots: 5751. Utilization: 100% across all threads and tasks. Use the `groupby` kwarg to break down by thread and/or task

[21]: `# Manual memory management`
`using LinearAlgebra`
`B = similar(A)`
`@time for i in 1:1000000`
 `mul!(B, A, A)`
 `m = maximum(B)`
 `A[i] = B / m`
`end`

5.177417 seconds (3.00 M allocations: 13.724 GiB, 7.17% gc time)

[22]: `?mul!`

search: `mul! rmul! lmul! accumulate! muladd widemul accumulate module Module`

[22]: `mul!(Y, A, B) -> Y`
Calculates the matrix-matrix or matrix-vector product `AB` and stores the result in `Y`, overwriting the existing value of `Y`. Note that `Y` must not be aliased with either `A` or `B`.

Simple 0 6 Julia 1.8.5 | Idle Mem: 2.36 GB Saving completed Mode: Edit Ln 7, Col 15 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 5.2K/18.28.0KB/s Tue Feb 7 09:40

symmetry.pi.local

JupyterLab Linear Algebra - The Julia Language Mem:2.36 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```

200 ...e/reduce.jl:659; mapreduce_impl(f::t...
44 ...oatfuncs.jl:15; signbit
99 @Base/range.jl:883; iterate
99 ...romotion.jl:477; ==
1 ...e/reduce.jl:664; mapreduce_impl(f::t...
1 ...e/reduce.jl:630; isbadzero
1 @Base/float.jl:410; ==
1 @Base/range.jl:883; iterate
1 @Base/promotion.jl:477; ==

```

Total snapshots: 5751. Utilization: 100% across all threads and tasks. Use the `groupby` kwarg to break down by thread and/or task

```

[*]: # Manual memory management
using LinearAlgebra
B = similar(A)
@time for i in 1:1000000
    mul!(B, A, A)
    m = maximum(B)
    A .= B / m
end

```

[22]: ?mul!

search: mul! rmul! lmul! accumulate! muladd widemul accumulate module Module

[22]: mul!(Y, A, B) -> Y
Calculates the matrix-matrix or matrix-vector product AB and stores the result in Y , overwriting the existing value of Y . Note that Y must not be aliased with either A or B .

Examples

Simple 0 6 Julia 1.8.5 | Busy Mem: 2.36 GB Saving completed Mode: Command Ln 1, Col 2 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 0.0KB/s 0.5KB/s Tue Feb 7 09:41

symmetry.pi.local Linear Algebra - The Julia Language Mem:2.46 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```

1 | ...e/reduce.jl:664; mapreduce_impl(f::t...
1 | ...e/reduce.jl:630; isbadzero
1 | @Base/float.jl:410; ==
1 | @Base/range.jl:883; iterate
1 | @Base/promotion.jl:477; ==
Total snapshots: 5751. Utilization: 100% across all threads and tasks. Use the `groupby` kwarg to break down by thread and/or task

[*]: # Manual memory management
using LinearAlgebra
B = similar(A)
@time for i in 1:1000000
    mul!(B, A, A)
    m = maximum(B)
    # A .= B ./ m
    A .= B
end

[22]: ?mul!
search: mul! rmul! lmul! accumulate! muladd widemul accumulate module Module

[22]: mul!(Y, A, B) -> Y
Calculates the matrix-matrix or matrix-vector product $AB$ and stores the result in Y, overwriting the existing value of Y. Note that Y must not be aliased with either A or B.

Examples

julia> A=[1.0 2.0; 3.0 4.0]; B=[1.0 1.0; 1.0 1.0]; Y = similar(B); mul!(Y, A, B);

```

Simple 0 6 Julia 1.8.5 | Busy Mem: 2.46 GB Saving completed Mode: Command Ln 1, Col 2 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 2.0K/18 17.0K/95 Tue Feb 7 09:42

symmetry.pi.local

JupyterLab Linear Algebra - The Julia Language Mem:2.52 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```

1 | ...e/reduce.jl:664; mapreduce_impl(f::t...
1 | ...e/reduce.jl:630; isbadzero
1 | @Base/float.jl:410; ==
1 | @Base/range.jl:883; iterate
1 | @Base/promotion.jl:477; ==
Total snapshots: 5751. Utilization: 100% across all threads and tasks. Use the `groupby` kwarg to break down by thread and/or task

[*]: # Manual memory management
using LinearAlgebra
B = similar(A)
@time for i in 1:1000000
    mul!(B, A, A)
    m = maximum(B)
    # A .= B ./ m
    # A .= B
end

[22]: ?mul!
search: mul! rmul! lmul! accumulate! muladd widemul accumulate module Module

[22]: mul!(Y, A, B) -> Y
Calculates the matrix-matrix or matrix-vector product $AB$ and stores the result in Y, overwriting the existing value of Y. Note that Y must not be aliased with either A or B.

Examples

julia> A=[1.0 2.0; 3.0 4.0]; B=[1.0 1.0; 1.0 1.0]; Y = similar(B); mul!(Y, A, B);

```

Simple 0 6 Julia 1.8.5 | Idle Mem: 2.52 GB Saving completed Mode: Command Ln 1, Col 2 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 7.0KB/s 14.6KB/s Tue Feb 7 09:43

symmetry.pi.local Linear Algebra - The Julia Language Mem:2.45 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```

1 | ...e/reduce.jl:664; mapreduce_impl(f::t...
1 | ...e/reduce.jl:630; isbadzero
1 | @Base/float.jl:410; ==
1 | @Base/range.jl:883; iterate
1 | @Base/promotion.jl:477; ==
Total snapshots: 5751. Utilization: 100% across all threads and tasks. Use the `groupby` kwarg to break down by thread and/or task
[ ]: # Put code into a function
[28]: # Manual memory management
using LinearAlgebra
B = similar(A)
@time for i in 1:1000000
    mul!(B, A, A)
    m = maximum(B)
    # A .= B ./ m
    # A .= B
end
1.862937 seconds (1000.00 k allocations: 15.259 MiB, 0.38% gc time)
[22]: ?mul!
search: mul! rmul! lmul! accumulate! muladd widemul accumulate module Module
[22]: mul!(Y, A, B) -> Y
Calculates the matrix-matrix or matrix-vector product $AB$ and stores the result in Y, overwriting the existing value of Y. Note that Y must not be aliased with either A or B.

```

Simple 0 6 Julia 1.8.5 | Idle Mem: 2.45 GB Saving completed Mode: Edit Ln 2, Col 1 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 2.54KB/s 31.4KB/s Tue Feb 7 09:44

symmetry.pi.local

JupyterLab Linear Algebra - The Julia Language Mem:2.36 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```

1 | ...e/reduce.jl:664; mapreduce_impl(f::t...
1 | ...e/reduce.jl:630; isbadzero
1 | @Base/float.jl:410; ==
1 | @Base/range.jl:883; iterate
1 | @Base/promotion.jl:477; ==
Total snapshots: 5751. Utilization: 100% across all threads and tasks. Use the `groupby` kwarg to break down by thread and/or task

```

[29]: # Put code into a function

```

function squaremany(A)
    for i in 1:1000000
        A = A * A
        A /= maximum(A)
    end
    return A
end

```

[29]: squaremany (generic function with 1 method)

[*]: @time squaremany(A)

[28]: # Manual memory management

```

using LinearAlgebra
B = similar(A)
@time for i in 1:1000000
    mul!(B, A, A)
    m = maximum(B)
    # A .= B ./ m
    # A .= B
end

```

1.862937 seconds (1000.00 k allocations: 15.259 MiB, 0.38% gc time)

Simple 0 s 6 Julia 1.8.5 | Busy Mem: 2.36 GB Saving completed Mode: Command Ln 6, Col 5 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 4.0K/18.5KB/s Tue Feb 7 09:44

symmetry.pi.local

JupyterLab Linear Algebra - The Julia Language Mem:2.37 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```

1 | ...e/reduce.jl:664; mapreduce_impl(f::t...
1 | ...e/reduce.jl:630; isbadzero
1 | @Base/float.jl:410; ==
1 | @Base/range.jl:883; iterate
1 | @Base/promotion.jl:477; ==
Total snapshots: 5751. Utilization: 100% across all threads and tasks. Use the `groupby` kwarg to break down by thread and/or task
[29]: # Put code into a function
function squaremany(A)
    for i in 1:1000000
        A = A * A
        A /= maximum(A)
    end
    return A
end
[29]: squaremany (generic function with 1 method)
[30]: @time squaremany(A)
4.548773 seconds (2.00 M allocations: 13.709 GiB, 7.90% gc time, 0.14% compilation time)
[30]: 30x30 Matrix{Float64}:
NaN NaN NaN NaN NaN NaN NaN NaN ... NaN NaN NaN NaN NaN NaN NaN
NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN

```

Simple 0 6 Julia 1.8.5 | Idle Mem: 2.37 GB Saving completed Mode: Command Ln 6, Col 5 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 9.2K/24.9KB/s Tue Feb 7 09:45

symmetry.pi.local

JupyterLab Linear Algebra - The Julia Language Mem:2.48 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```

return A
end

```

[29]: squaremany (generic function with 1 method)

[31]: `A = randn(30, 30)`
`@time squaremany(A);`
4.714627 seconds (2.00 M allocations: 13.709 GiB, 7.82% gc time)

[32]: `# Manual memory management`
`using LinearAlgebra`
`function squaremany_fast!(A)`
 `B = similar(A)`
 `for i in 1:1000000`
 `mul!(B, A, A)`
 `B .= A / maximum(A)`
 `end`
 `return A`
`end`

[32]: squaremany_fast! (generic function with 1 method)

[22]: `?mul!`
search: `mul!` `rmul!` `lmul!` `accumulate!` `muladd` `widemul` `accumulate` `module` `Module`

[22]: `mul!(Y, A, B) -> Y`
Calculates the matrix-matrix or matrix-vector product `AB` and stores the result in `Y`, overwriting the existing value of `Y`. Note that `Y` must not be aliased with either `A` or `B`.

Simple 0 s 6 Julia 1.8.5 | Idle Mem: 2.48 GB Saving completed Mode: Edit Ln 1, Col 6 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 2.5K/11.0KB/s Tue Feb 7 09:46

symmetry.pi.local

JupyterLab Linear Algebra - The Julia Language Mem:2.51 GB

File Edit View Run Kernel Git Tabs Settings Help

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```
return A
end

[29]: squaremany (generic function with 1 method)

[31]: A = randn(30, 30)
@time squaremany(A);

4.714627 seconds (2.00 M allocations: 13.709 GiB, 7.82% gc time)

[32]: # Manual memory management
using LinearAlgebra
function squaremany_fast!(A)
    B = similar(A)
    for i in 1:1000000
        mul!(B, A, A)
        B .= A / maximum(A)
    end
    return A
end

[32]: squaremany_fast! (generic function with 1 method)

[33]: @time squaremany_fast!(A);

4.066152 seconds (1.02 M allocations: 6.856 GiB, 5.13% gc time, 0.80% compilation time)

[ ]:
```

Simple 0 s 6 Julia 1.8.5 | Idle Mem: 2.51 GB Saving completed Mode: Command Ln 1, Col 1 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 0.5K/18.5KB/s Tue Feb 7 09:47

symmetry.pi.local

JupyterLab Linear Algebra - The Julia Language Mem:2.40 GB

File Edit View Run Kernel Git Tabs Settings Help

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```
return A
end

[29]: squaremany (generic function with 1 method)

[31]: A = randn(30, 30)
@time squaremany(A);

4.714627 seconds (2.00 M allocations: 13.709 GiB, 7.82% gc time)

[34]: # Manual memory management
using LinearAlgebra
function squaremany_fast!(A)
    B = similar(A)
    for i in 1:1000000
        mul!(B, A, A)
        A .= B ./ maximum(B)
    end
    return A
end

[34]: squaremany_fast! (generic function with 1 method)

[36]: A = randn(30, 30)
@time squaremany_fast!(A);

3.106352 seconds (1 allocation: 7.188 KiB)

[ ]:
```

Simple 0 s 6 Julia 1.8.5 | Idle Mem: 2.40 GB Saving completed Mode: Command Ln 1, Col 1 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 8.4K/28.6KB/s Tue Feb 7 09:47

symmetry.pi.local

JupyterLab Linear Algebra - The Julia Language Mem:2.48 GB

File Edit View Run Kernel Git Tabs Settings Help

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```
return A
end
```

[29]: squaremany (generic function with 1 method)

```
[31]: A = randn(30, 30)
@time squaremany(A);
```

4.714627 seconds (2.00 M allocations: 13.709 GiB, 7.82% gc time)

```
[34]: # Manual memory management
using LinearAlgebra
function squaremany_fast!(A)
    B = similar(A)
    for i in 1:1000000
        mul!(B, A, A)
        A .= B ./ maximum(B)
    end
    return A
end
```

[34]: squaremany_fast! (generic function with 1 method)

```
[36]: A = randn(30, 30)
@time squaremany_fast!(A);
```

3.106352 seconds (1 allocation: 7.188 KiB)

[]:

Simple 0 s 6 Julia 1.8.5 | Idle Mem: 2.48 GB Saving completed Mode: Edit Ln 4, Col 19 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom 1.5KB/s 4.0KB/s Charged 100% Tue Feb 7 09:50

symmetry.pi.local

JupyterLab Linear Algebra - The Julia Language Mem:2.66 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```

1 ...e/reduce.jl:664; mapreduce_impl(f::t...
1 ...e/reduce.jl:630; isbadzero
1 @Base/float.jl:410; ==
1 @Base/range.jl:883; iterate
1 @Base/promotion.jl:477; ==
Total snapshots: 5751. Utilization: 100% across all threads and tasks. Use the `groupby` kwarg to break down by thread and/or task

```

[29]: # Put code into a function

```

function squaremany(A)
    for i in 1:1000000
        A = A * A
        A /= maximum(A)
    end
    return A
end

```

[29]: squaremany (generic function with 1 method)

[31]: A = randn(30, 30)
@time squaremany(A);

4.714627 seconds (2.00 M allocations: 13.709 GiB, 7.82% gc time)

[37]: typeof(A)

[37]: Matrix{Float64} (alias for Array{Float64, 2})

[34]: # Manual memory management

```

using LinearAlgebra
function squaremany_fast!(A)

```

Simple 0 s 6 Julia 1.8.5 | Idle Mem: 2.66 GB Saving completed Mode: Command Ln 3, Col 29 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 6.2K/18 38.7KB/s Tue Feb 7 09:52

symmetry.pi.local

JupyterLab Linear Algebra - The Julia Language Mem:2.37 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```
@time squaremany(A);  
4.714627 seconds (2.00 M allocations: 13.709 GiB, 7.82% gc time)  
[37]: typeof(A)  
[37]: Matrix{Float64} (alias for Array{Float64, 2})  
[34]: # Manual memory management  
using LinearAlgebra  
function squaremany_fast!(A)  
    B = similar(A)  
    for i in 1:1000000  
        mul!(B, A, A)  
        A .= B ./ maximum(B)  
    end  
    return A  
end  
[34]: squaremany_fast! (generic function with 1 method)  
[36]: A = randn(30, 30)  
@time squaremany_fast!(A);  
3.106352 seconds (1 allocation: 7.188 KiB)  
[*]: @profile squaremany_fast!(A);  
[ ]: |
```

Simple 0 s 6 Julia 1.8.5 | Busy Mem: 2.37 GB Saving completed Mode: Edit Ln 1, Col 1 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom 0.54KB/s 23.1KB/s Charged 0.54KB/s 23.1KB/s Tue Feb 7 09:53

symmetry.pi.local

JupyterLab Linear Algebra - The Julia Language Mem:2.38 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```
[37]: typeof(A)
[37]: Matrix{Float64} (alias for Array{Float64, 2})

[34]: # Manual memory management
using LinearAlgebra
function squaremany_fast!(A)
    B = similar(A)
    for i in 1:1000000
        mul!(B, A, A)
        A .= B ./ maximum(B)
    end
    return A
end

[34]: squaremany_fast! (generic function with 1 method)

[36]: A = randn(30, 30)
@time squaremany_fast!(A);
3.106352 seconds (1 allocation: 7.188 KiB)

[39]: Profile.clear()
@profile squaremany_fast!(A);

[ ]: Profil|
```

Simple 0 s 6 Julia 1.8.5 | Idle Mem: 2.38 GB Saving completed Mode: Edit Ln 1, Col 7 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 2.0KB/s 39.3KB/s Tue Feb 7 09:53

symmetry.pi.local

JupyterLab Linear Algebra - The Julia Language Mem:2.39 GB

File Edit View Run Kernel Git Tabs Settings Help

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```

[40]: Profile.print()

Overhead | [+additional indent] Count File:Line; Function
-----|-----
3|3  @Base/range.jl:321; steprange_last(start::Int64, st...
3|3  @Base/range.jl:352; steprange_last(start::Int64, st...
3|3  @Base/reduce.jl:0; mapreduce_impl(f::typeof(identi...
1  @Base/reduce.jl:638; mapreduce_impl(f::typeof(identi...
1|1  @Base/reduce.jl:636; getindex
1  @Base/reduce.jl:645; mapreduce_impl(f::typeof(identi...
1  @Base/range.jl:22; Colon
1  @Base/range.jl:24; _colon
1  @Base/range.jl:369; StepRange
1  @Base/range.jl:316; StepRange
1  @Base/range.jl:321; steprange_last(start::Int64,...
3101 @Base/task.jl:484; (::IJulia.var"#15#18")()
3101 ...lia/src/eventloop.jl:8; eventloop(socket::ZMQ.Socket)
3101 @Base/essentials.jl:726; invokelatest
3101 @Base/essentials.jl:729; #invokelatest#2
3101 ...execute_request.jl:67; execute_request(socket::ZMQ.S...
3101 ...SoftGlobalScope.jl:65; softscope_include_string(m:...
3101 @Base/loading.jl:1428; include_string(mapexpr::typ...
3101 @Base/boot.jl:368; eval
1605 In[34]:6; squaremany_fast!(A::Matrix{...
1605 ...ra/src/matmul.jl:276; mul!
1605 ...a/src/matmul.jl:161; mul!
5  ...a/src/matmul.jl:642; gemm_wrapper!(C::Matrix{...
5  ...a/src/matmul.jl:725; lapack_size
5  @Base/char.jl:213; ==
5  ...e/promotion.jl:477; ==
1  @Base/matmul.jl:653; gemm_wrapper!(C::Matrix{...

```

Simple 0 6 Julia 1.8.5 | Idle Mem: 2.39 GB Saving completed Mode: Edit Ln 1, Col 1 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 4.6KHz 17.7KB/s Tue Feb 7 09:53

symmetry.pi.local

JupyterLab Linear Algebra - The Julia Language Mem:2.46 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```
1 @Base/range.jl:369; StepRange
1 @Base/range.jl:316; StepRange
1 1 @Base/range.jl:321; steprange_last(start::Int64,...
3101 @Base/task.jl:484; (::IJulia.var"#15#18"())
3101 ...lia/src/eventloop.jl:8; eventloop(socket::ZMQ.Socket)
3101 @Base/essentials.jl:726; invokelatest
3101 @Base/essentials.jl:729; #invokelatest#2
3101 ...execute_request.jl:67; execute_request(socket::ZMQ.S...
3101 ...SoftGlobalScope.jl:65; softscope_include_string(m:...
3101 @Base/loading.jl:1428; include_string(mapexpr::typ...
3101 @Base/boot.jl:368; eval
1605 In[34]:6; squaremany_fast!(A::Matrix{...
1605 ...ra/src/matmul.jl:276; mul!
1605 ...a/src/matmul.jl:161; mul!
5 ...a/src/matmul.jl:642; gemm_wrapper!(C::Matrix{...
5 ...a/src/matmul.jl:725; lapack_size
5 @Base/char.jl:213; ==
5 ...e/promotion.jl:477; ==
1 ...a/src/matmul.jl:653; gemm_wrapper!(C::Matrix{...
1 ...a/src/matmul.jl:660; gemm_wrapper!(C::Matrix{...
1 @Base/promotion.jl:477; ==
1598 ...a/src/matmul.jl:674; gemm_wrapper!(C::Matrix{...
1 ...nearAlgebra.jl:213; _chkstride1(ok::Bool)
2 ...ra/src/blas.jl:1504; gemm!(transA::Char, tra...
1 @Base/array.jl:150; size
1 @Base/char.jl:213; ==
1 ...e/promotion.jl:477; ==
2 ...ra/src/blas.jl:1506; gemm!(transA::Char, tra...
2 @Base/char.jl:213; ==
2 ...e/promotion.jl:477; ==
1 ...ra/src/blas.jl:1507; gemm!(transA::Char, tra...
```

Simple 0 s 6 Julia 1.8.5 | Idle Mem: 2.46 GB Saving completed Mode: Command Ln 1, Col 1 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 1.0K/23.5KB/s Tue Feb 7 09:56

symmetry.pi.local

JupyterLab Linear Algebra - The Julia Language Mem:2.49 GB

File Edit View Run Kernel Git Tabs Settings Help

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb X

Code git Julia 1.8.5

```

end

[34]: squaremany_fast! (generic function with 1 method)

[36]: A = randn(30, 30)
      @time squaremany_fast!(A);

      3.106352 seconds (1 allocation: 7.188 KiB)

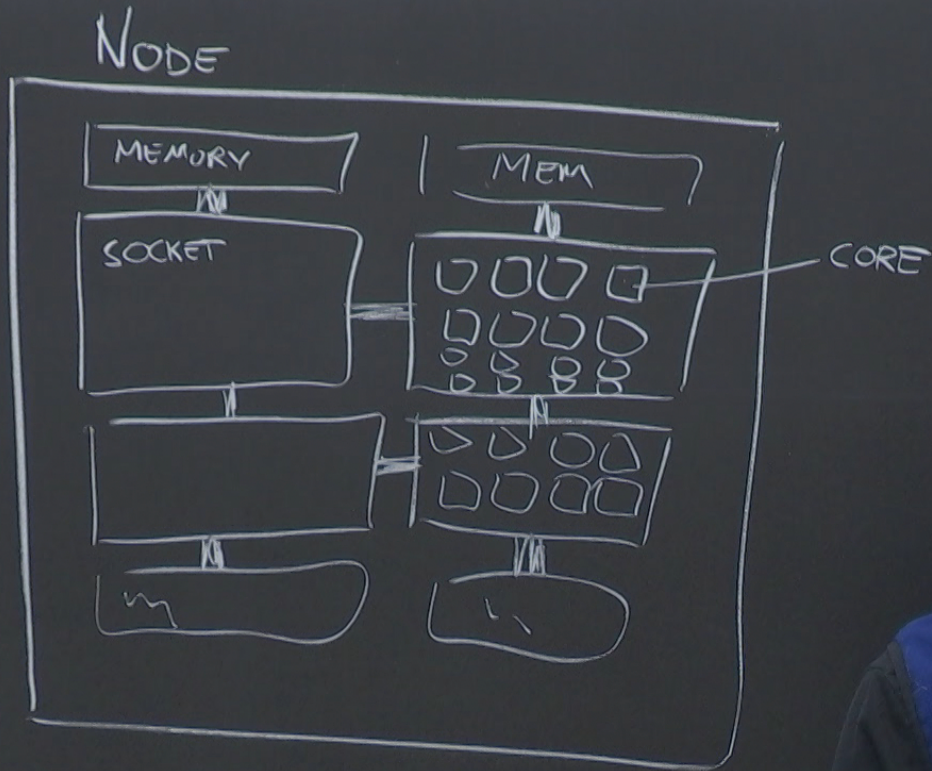
[39]: Profile.clear()
      @profile squaremany_fast!(A);

[40]: Profile.print()

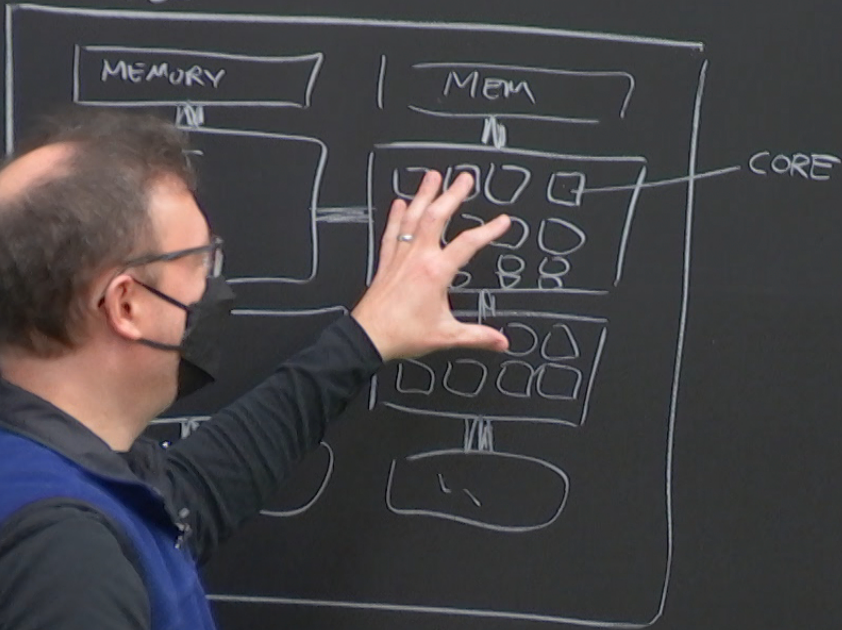
Overhead | [+additional indent] Count File:Line; Function
=====
 3|3  @Base/range.jl:321; steprange_last(start::Int64, st...
 3|3  @Base/range.jl:352; steprange_last(start::Int64, st...
 3|3  @Base/reduce.jl:0; mapreduce_impl(f::typeof(identi...
 1  @Base/reduce.jl:638; mapreduce_impl(f::typeof(identi...
 1|1  @Base/reduce.jl:636; getindex
 1  @Base/reduce.jl:645; mapreduce_impl(f::typeof(identi...
 1  @Base/range.jl:22; Colon
 1  @Base/range.jl:24; _colon
 1  @Base/range.jl:369; StepRange
 1  @Base/range.jl:316; StepRange
 1  1  @Base/range.jl:321; steprange_last(start::Int64,...
3101 @Base/task.jl:484; (::IJulia.var"#15#18")()
3101 ...lia/src/eventloop.jl:8; eventloop(socket::ZMQ.Socket)
3101 @Base/essentials.jl:726; invoke_latest
3101 @Base/essentials.jl:720; #invoke_latest#2

```

Simple 0 6 Julia 1.8.5 | Idle Mem: 2.49 GB Saving completed Mode: Command Ln 1, Col 1 Performance 2023.ipynb



NODE



$\frac{\text{FLOP}}{\text{BYTE}}$

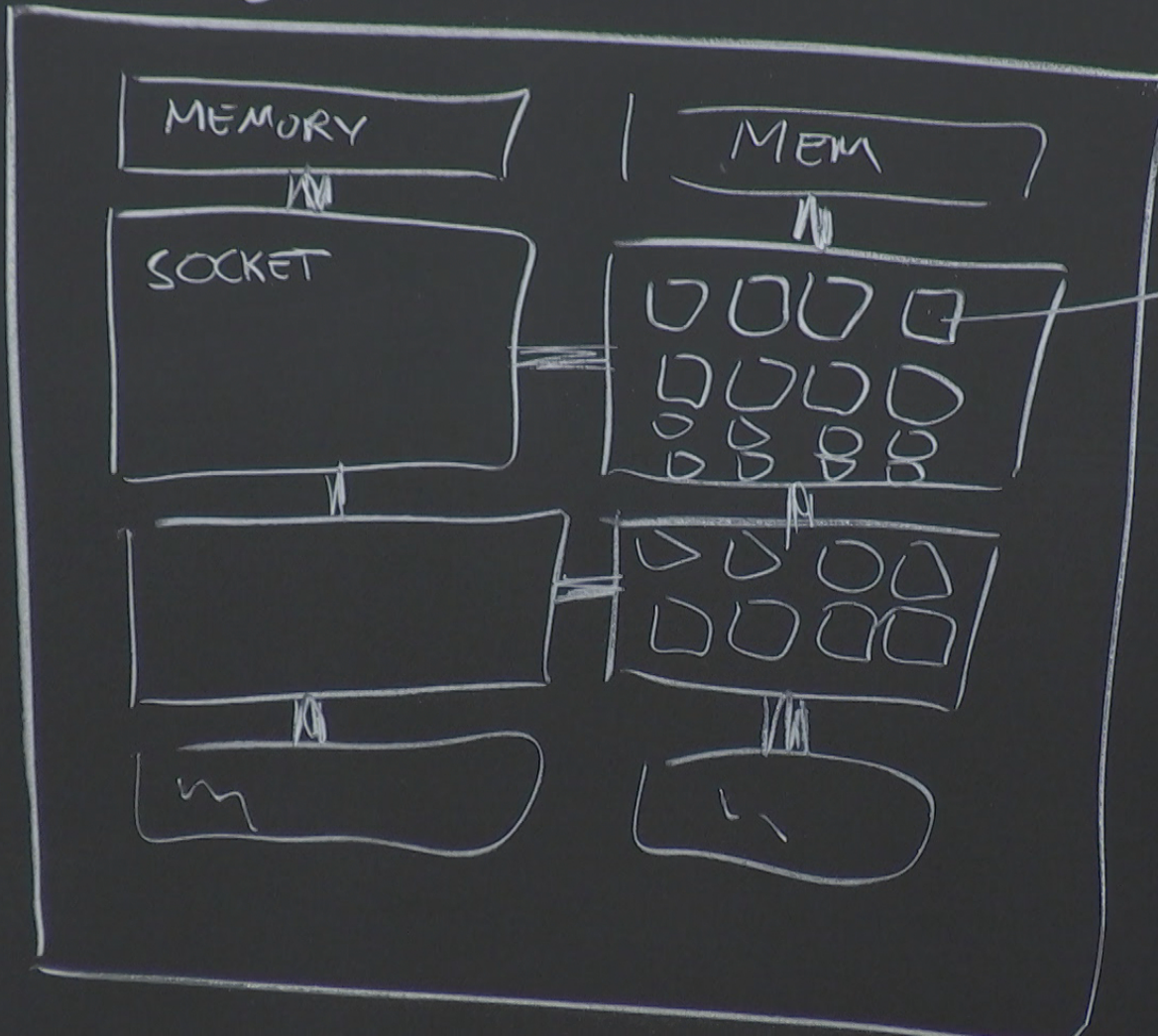
+ - *

adding vectors:

$$a[i] = b[i] + c[i] \quad i \in 1:n$$

n	Flop	$\frac{1}{24}$	$\frac{\text{Flop}}{\text{Byte}}$
24n	Bytes		

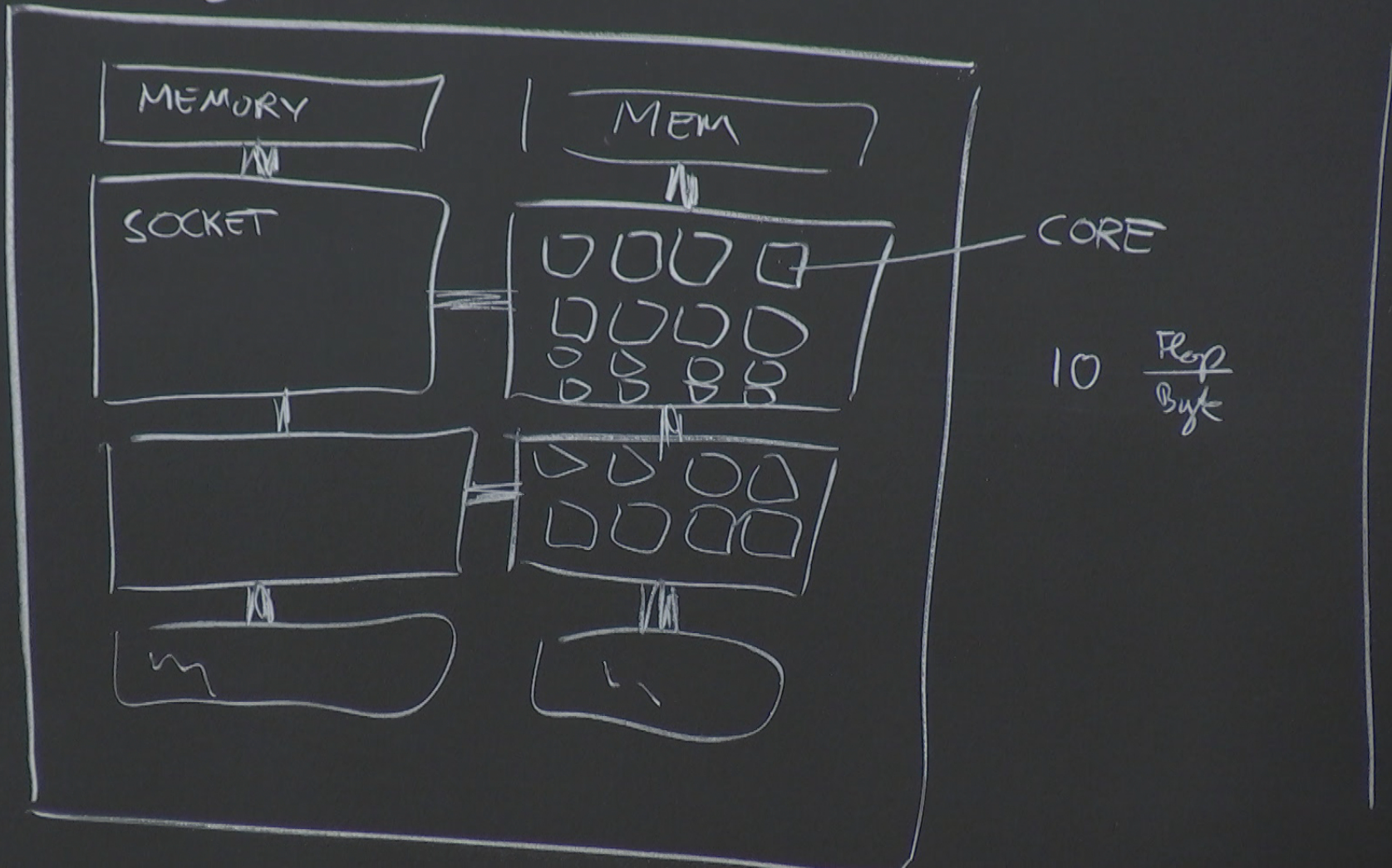
NODE



CORE

10 $\frac{R}{S}$

NODE



$\frac{\text{FLOP}}{\text{BYTE}}$

+ - *

adding vectors:

$$a[i] = b[i] + c[i] \quad i \in 1:n$$

n Flop
24 n Bytes

$$\frac{1}{24} \frac{\text{Flop}}{\text{Byte}} \approx 0.04$$

matrix mult:

$n \times n$ Matrix

$$A = B \cdot C$$

$24 n^2$ Bytes

$\frac{\text{Flop}}{\text{Byte}}$

+ - *

adding vectors: $a[i] = b[i] + c[i] \quad i \in 1:n$

n Flop
24n Bytes

$$\frac{1}{24} \frac{\text{Flop}}{\text{Byte}} \approx 0.04$$

matrix mult:

$n \times n$ Matrix

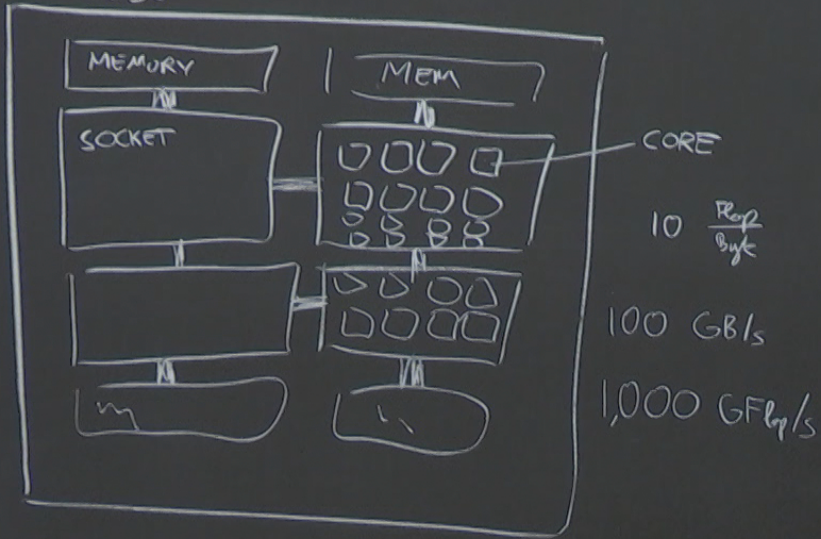
$$A = B \cdot C$$

$24n^2$ Bytes

$2n^3$ Flop

$$\frac{n}{12} \frac{\text{Flop}}{\text{Byte}}$$

NODE



$\frac{\text{Flop}}{\text{Byte}}$

+ - *

adding vectors:

$$a[i] = b[i] + c[i] \quad i \in 1:n$$

n Flop
 $24n$ Bytes

$$\frac{1}{24} \frac{\text{Reg}}{\text{Byte}} \approx 0.04$$

matrix mult:

$n \times n$ Matrix

$$A = B \cdot C$$

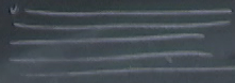
$24n^2$ Bytes

$2n^3$ Flop

$$\frac{n}{12} \frac{\text{Flop}}{\text{Byte}}$$

2.5 GHz

0.4 ns



$$Q = I \cdot t$$

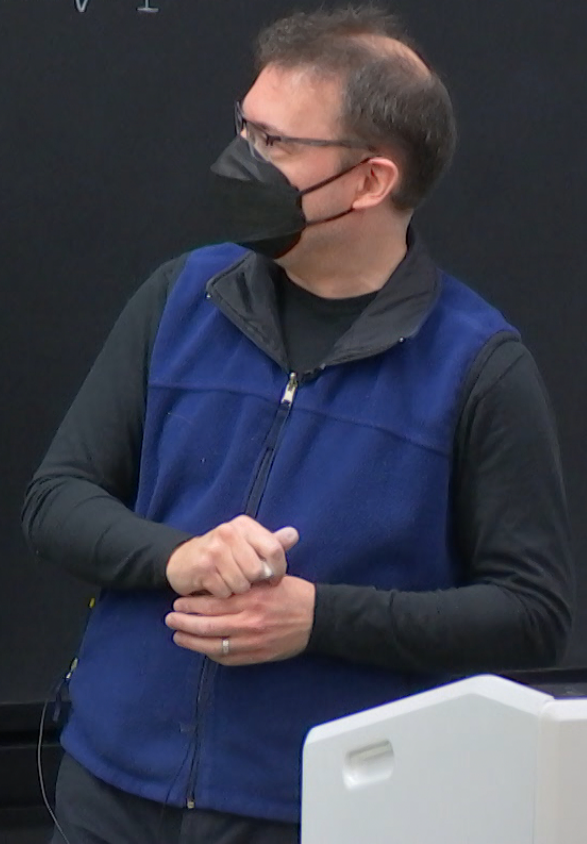
$$V = I \cdot R$$

$$I \sim v$$

$$V \sim v$$

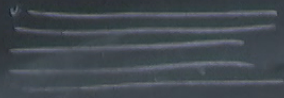
$$P \sim v^2$$

$$P = V \cdot I$$



2.5 GHz

0.4 ns



$$Q = I \cdot t$$

$$V = I \cdot R$$

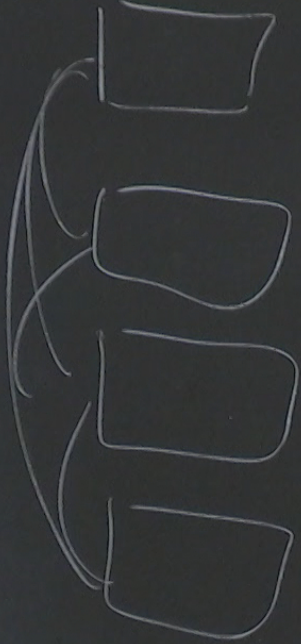
$$I \sim v$$

$$P = V \cdot I$$

$$V \sim v$$

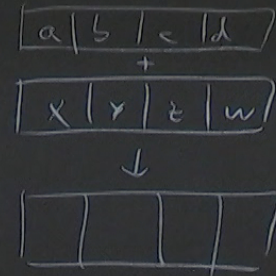
$$P \sim v^2$$

Node

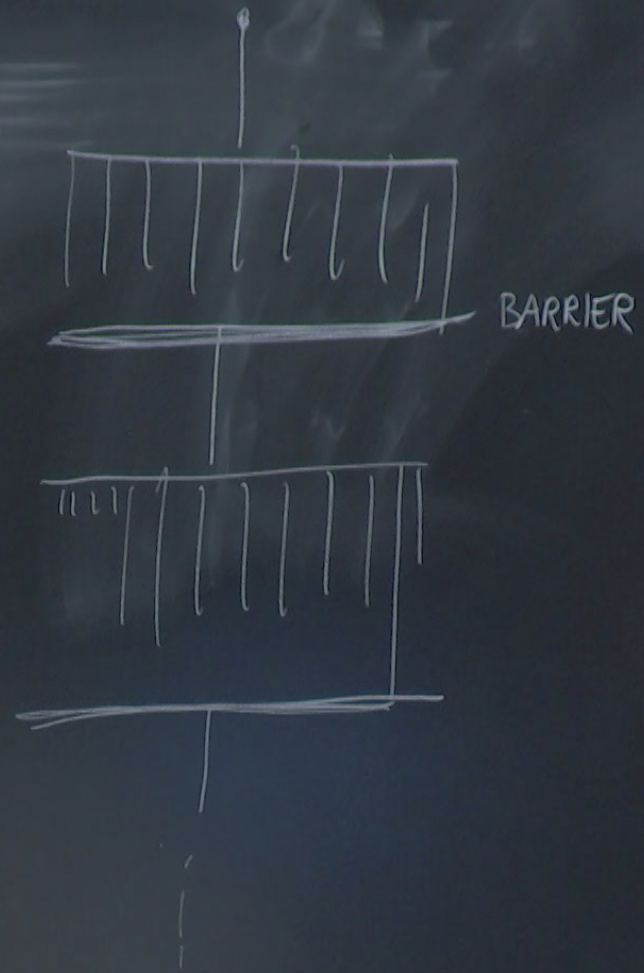


DISTRIBUTED COMPUTING

SIMD parallelization



Bulk synchronous



Safari File Edit View History Bookmarks Develop Window Help zoom Charged 4.2KB/s 12.0KB/s Tue Feb 7 10:36

symmetry.pi.local

JupyterLab Linear Algebra - The Julia Language Mem:2.37 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```
24 | 24 ...oatfuncs.jl:15; signbit
91 | 91 @Base/reduce.jl:659; mapreduce_impl(f::t...
91 | 91 @Base/range.jl:883; iterate
91 | 91 ...romotion.jl:477; ==
```

Total snapshots: 3115. Utilization: 100% across all threads and tasks. Use the `groupby` kwarg to break down by thread and/or task

Distributed Computing

```
[41]: using Distributed
[42]: nworkers()
[42]: 1
[*]: addprocs(4)
[ ]: |
```

Simple 0 s 6 Julia 1.8.5 | Busy Mem: 2.37 GB Saving completed Mode: Edit Ln 1, Col 1 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 0.5KB/s 2.5KB/s Tue Feb 7 10:37

symmetry.pi.local

JupyterLab Linear Algebra - The Julia Language Mem:3.46 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```
24 | ...oatfuncs.jl:15; signbit
91 | @Base/reduce.jl:659; mapreduce_impl(f::t...
91 | @Base/range.jl:883; iterate
91 | ...romotion.jl:477; ==
```

Total snapshots: 3115. Utilization: 100% across all threads and tasks. Use the `groupby` kwarg to break down by thread and/or task

Distributed Computing

```
[41]: using Distributed
[42]: nworkers()
[42]: 1
[43]: addprocs(4)
[43]: 4-element Vector{Int64}:
      2
      3
      4
      5
```

[]:

Simple 0 s 6 Julia 1.8.5 | Idle Mem: 3.46 GB Saving completed Mode: Edit Ln 1, Col 1 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom 2.5KB/s 18.6KB/s Charged Tue Feb 7 10:39

symmetry.pi.local

JupyterLab Distributed Computing - The Julia Language Mem:3.46 GB

File Edit View Run Kernel Git Tabs Settings Help

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```
[44]: nworkers()
[44]: 4
[45]: # serial `map`
a = [i for i in 1:10]
[45]: 10-element Vector{Int64}:
 1
 2
 3
 4
 5
 6
 7
 8
 9
10
[ ]: b = map(x -> 2x+1, a)
```

Simple 0 s 6 Julia 1.8.5 | Idle Mem: 3.46 GB Saving completed Mode: Edit Ln 1, Col 22 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom 1.6KB/s 4.7KB/s Charged 100% Tue Feb 7 10:41

symmetry.pi.local

JupyterLab Distributed Computing - The Julia Language Mem:3.76 GB

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```
11
13
15
17
19
21

[47]: c = pmap(x -> 2x+1, a)

[47]: 10-element Vector{Int64}:
      3
      5
      7
      9
     11
     13
     15
     17
     19
     21

[*]: c = pmap(x -> (sleep(5); 2x+1), a)

[ ]:
```

Simple 0 s 6 Julia 1.8.5 | Busy Mem: 3.76 GB Saving completed Mode: Edit Ln 1, Col 1 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom 1.0KHz 16.7KB/s Charged Tue Feb 7 10:41

symmetry.pi.local Distributed Computing - The Julia Language Mem:3.48 GB

File Edit View Run Kernel Git Tabs Settings Help

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```
[47]: c = pmap(x -> 2x+1, a)
```

```
[47]: 10-element Vector{Int64}:  
 3  
 5  
 7  
 9  
11  
13  
15  
17  
19  
21
```

```
[48]: c = pmap(x -> (sleep(5); 2x+1), a)
```

```
[48]: 10-element Vector{Int64}:  
 3  
 5  
 7  
 9  
11  
13  
15  
17  
19  
21
```

[]:

Simple 0 s 6 Julia 1.8.5 | Idle Mem: 3.48 GB Saving completed Mode: Edit Ln 1, Col 1 Performance 2023.ipynb

Safari File Edit View History Bookmarks Develop Window Help zoom Charged 6.0K/18 10.5K/75 Tue Feb 7 10:41

symmetry.pi.local Distributed Computing - The Julia Language Mem:3.50 GB

File Edit View Run Kernel Git Tabs Settings Help

Differential Equations Project X Wave Equation 2023.ipynb Performance 2023.ipynb

Code git Julia 1.8.5

```
[47]: c = pmap(x -> 2x+1, a)

[47]: 10-element Vector{Int64}:
      3
      5
      7
      9
     11
     13
     15
     17
     19
     21

[49]: @time c = pmap(x -> (sleep(5); 2x+1), a)
      15.101049 seconds (126.62 k allocations: 6.526 MiB, 0.44% compilation time)

[49]: 10-element Vector{Int64}:
      3
      5
      7
      9
     11
     13
     15
     17
     19
     21

[ ]:
```

Simple 0 s 6 Julia 1.8.5 | Idle Mem: 3.50 GB Saving completed Mode: Command Ln 1, Col 1 Performance 2023.ipynb