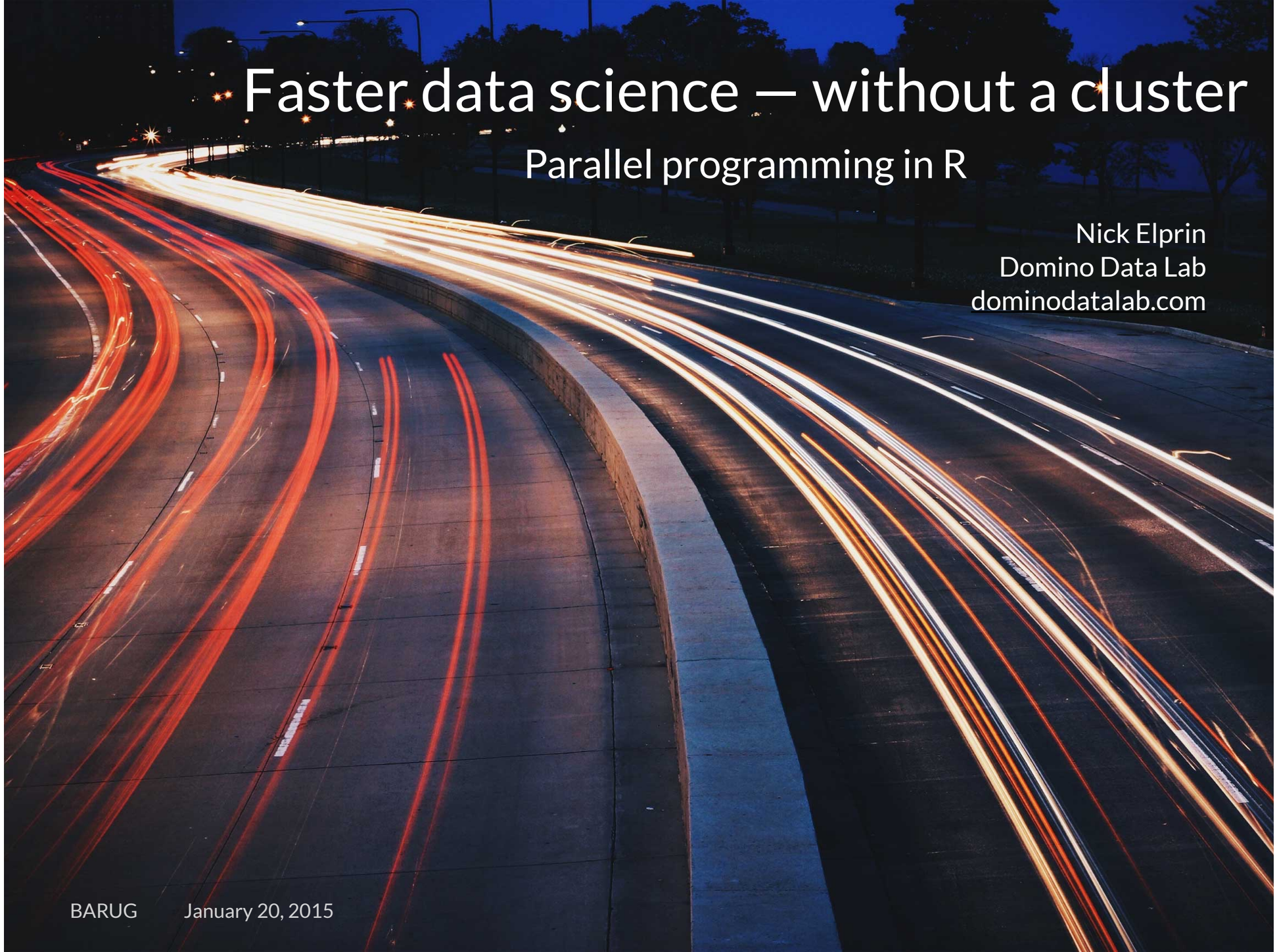


Faster data science — without a cluster

Parallel programming in R

Nick Elprin
Domino Data Lab
dominodatalab.com

BARUG January 20, 2015

Who am I?

- Founder of Domino Data Lab, a software platform for enterprise data science

- Previously built analytical software at a big hedge fund

- BA, MS in computer science

Outline

- Motivation
- Basic conceptual intro to parallelism, general principles and pitfalls
- Parallel programming in R
- Machine learning applications
- Domino
- Questions

Motivation

"Big data is like teenage sex: everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it."

— Dan Ariely

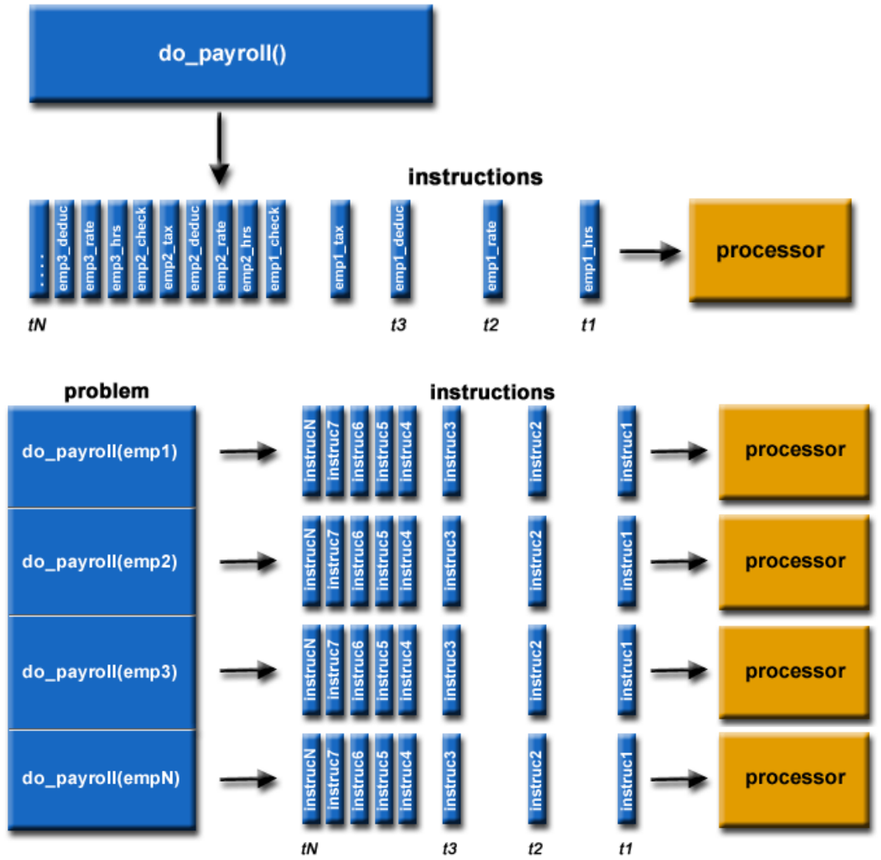
- Lots of "medium data" problems
  - Can fit in memory on one machine
- Lots of naturally parallel problems

- Easy to access large machines
- Clusters are hard
- Not everything fits map-reduce

Model	vCPU	Mem (GiB)	SSD Storage (GB)
r3.large	2	15.25	1 x 32
r3.xlarge	4	30.5	1 x 80
r3.2xlarge	8	61	1 x 160
r3.4xlarge	16	122	1 x 320
r3.8xlarge	32	244	2 x 320

Parallel programing 101

- Think about independent tasks (hint: “for” loops are a good place to start!)
  - Should be CPU-bound tasks
- Warning and pitfalls
  - Not a substitute for good code
  - Overhead
  - Shared resource contention
  - Thrashing



Source: Blaise Barney, Lawrence Livermore National Laboratory

Can parallelize at different “levels”

Experiments

Run different analyses at once

Algorithms

Write your code (or use a package) to parallelize functions or steps within your analysis

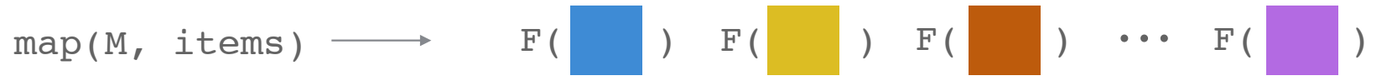
Math ops

Run against underlying libraries that parallelize low-level operations, e.g., openBLAS, ATLAS

Will focus on algorithms, with some brief comments on Experiments

Common Operation: Map

```
M = function(item) {
  manipulatedItem = ...
  manipulatedItem
}
```



So what's map-reduce?

Source: Blaise Barney, Lawrence Livermore National Laboratory

Parallelize tasks to match your resources

Computing something (CPU)

Reading from disk/database

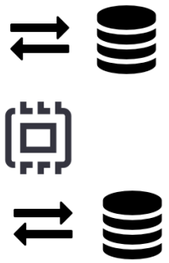
Writing to disk/database

Network IO (e.g., web scraping)

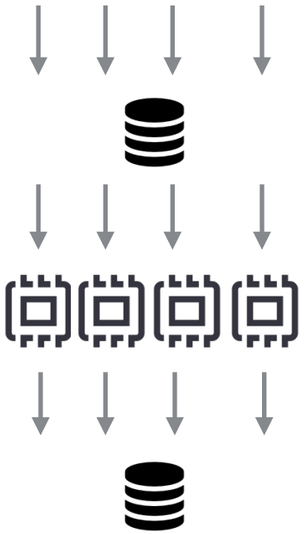
Saturating a resource will create a bottleneck

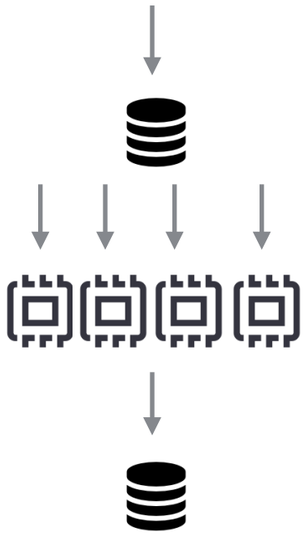
Parallelize tasks to match your resources

```
itemIDs = c(1, 2, … , n)
foreach(i = itemIDs) %dopar% {
  item = fetchData(i)
  result = computeSomething(item)
  saveResult(result)
}
```



```
items = fetchData(c(1, 2, … , n))
results = foreach(i = items) %dopar% {
  computeSomething(item)
}
saveResult(results)
```





Parallel programing in R

- General purpose
  - `parallel`
  - `foreach`
- More specialized
  - `randomForest`
  - `caret`
  - `plyr`

Demo

https://github.com/dominodatalab/parallel-r-examples

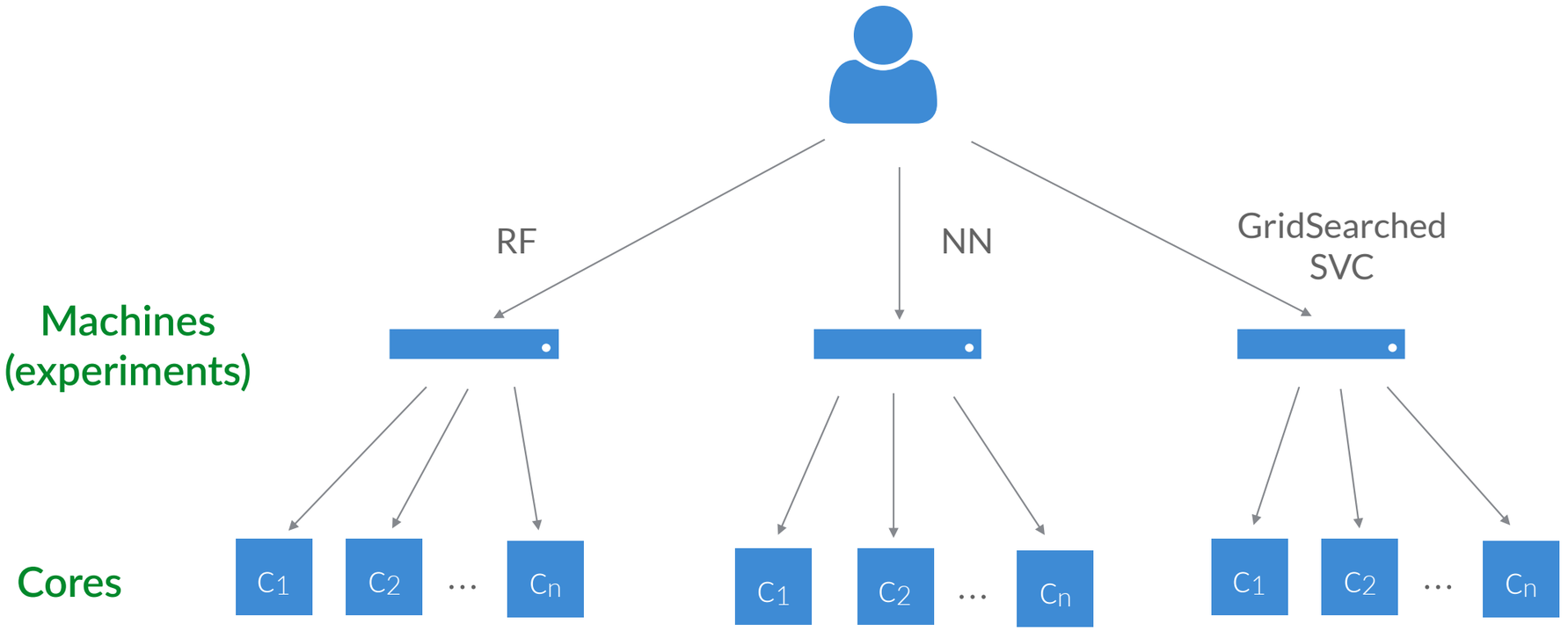
Many ML tasks are naturally parallelized

- Cross-validation
- Grid search
- Random forest
- KMeans
- Neural networks

Demo

https://github.com/dominodatalab/parallel-r-examples

Can compose layers of parallelism



Demo

Going deeper

http://cran.r-project.org/web/views/HighPerformanceComputing.html

http://topepo.github.io/caret/index.html

- Python!
  - Joblib
  - scikit learn (*n_jobs*)
    - `GridSearchCV, RandomForest, KMeans, cross_val_score`
  - IPython Notebook clusters

Webinar on parallel programming in R and Python.

Jan 28, 10:30am
dominodatalab.com/webinar

Check us out

dominodatalab.com

blog.dominodatalab.com

@dominodatalab