Bcolz

An Example Of Data Containers Applying The Principles Of New Hardware

Francesc Alted
Freelance Consultant
http://www.blosc.org/professional-services.html

Advanced Scientific Programming in Python
Reading, UK
September, 2016

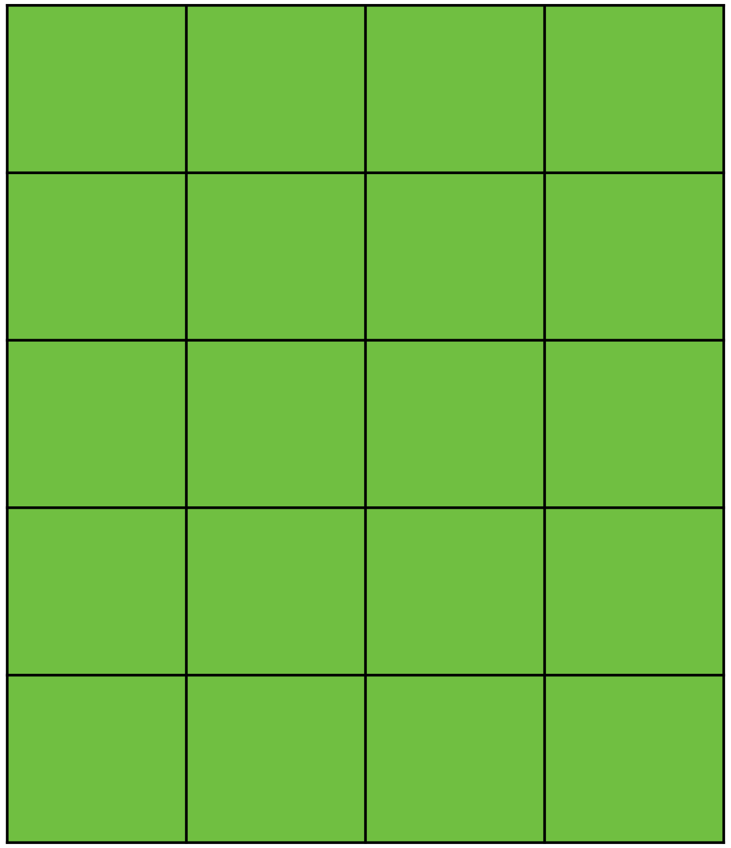
What is bcolz?

- bcolz provides data containers that can be used in a similar way than the ones in NumPy or Pandas

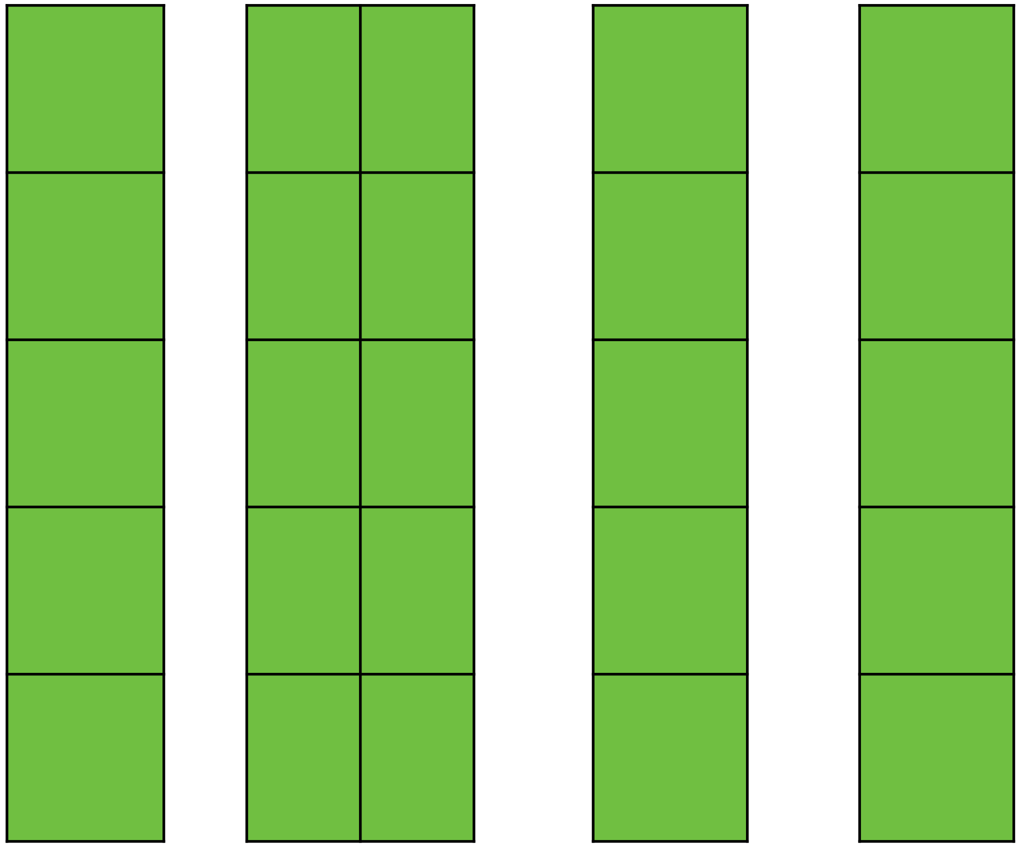
- The main difference is that data storage is **chunked**, not **contiguous**

- Also, it provides a layer for achieving independence of storage media: either **memory** or **disk** can be used.

bcolz Implements Two Flavors of Data Containers



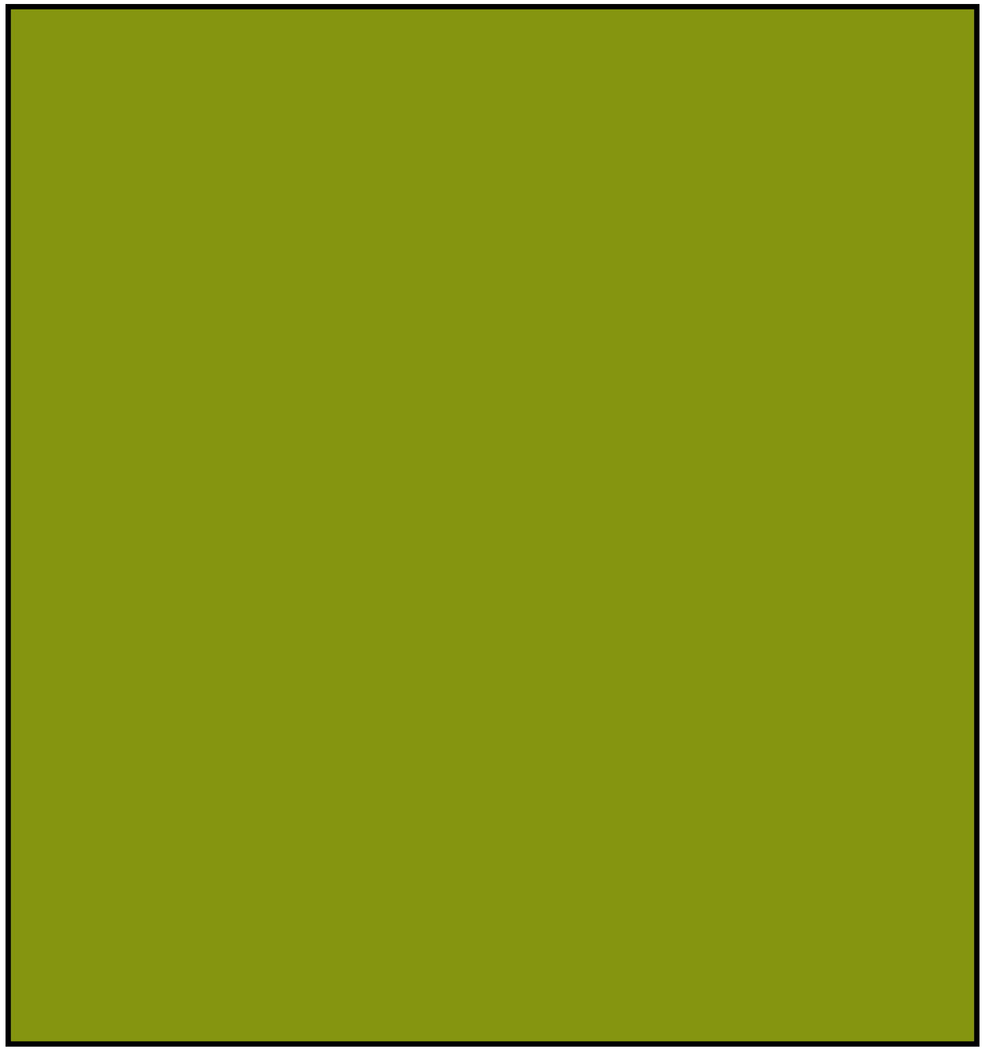
carray: homogenous, n-dim data types



ctable: heterogeneous types, columnar

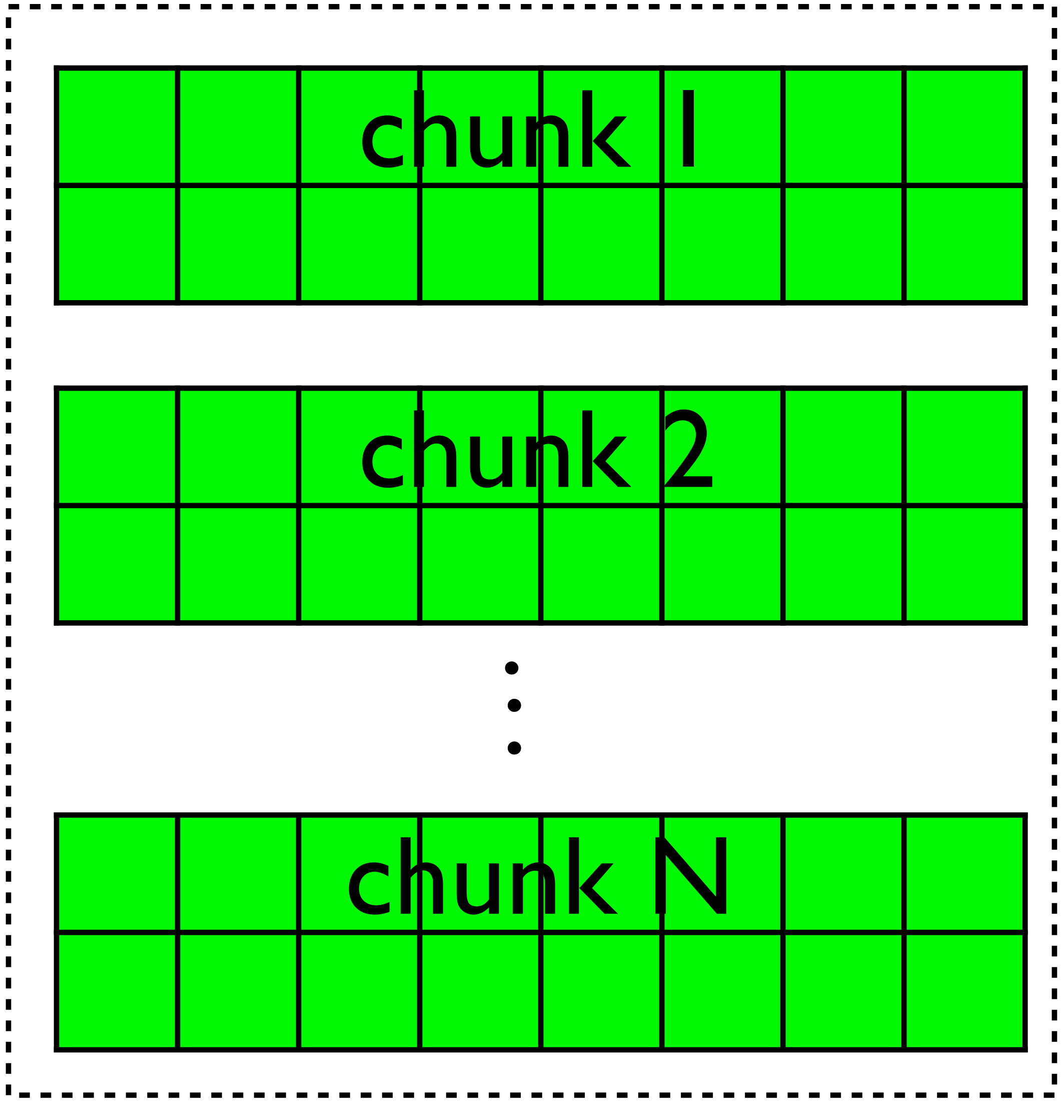
Contiguous vs Chunked

NumPy container



Contiguous memory

carray container

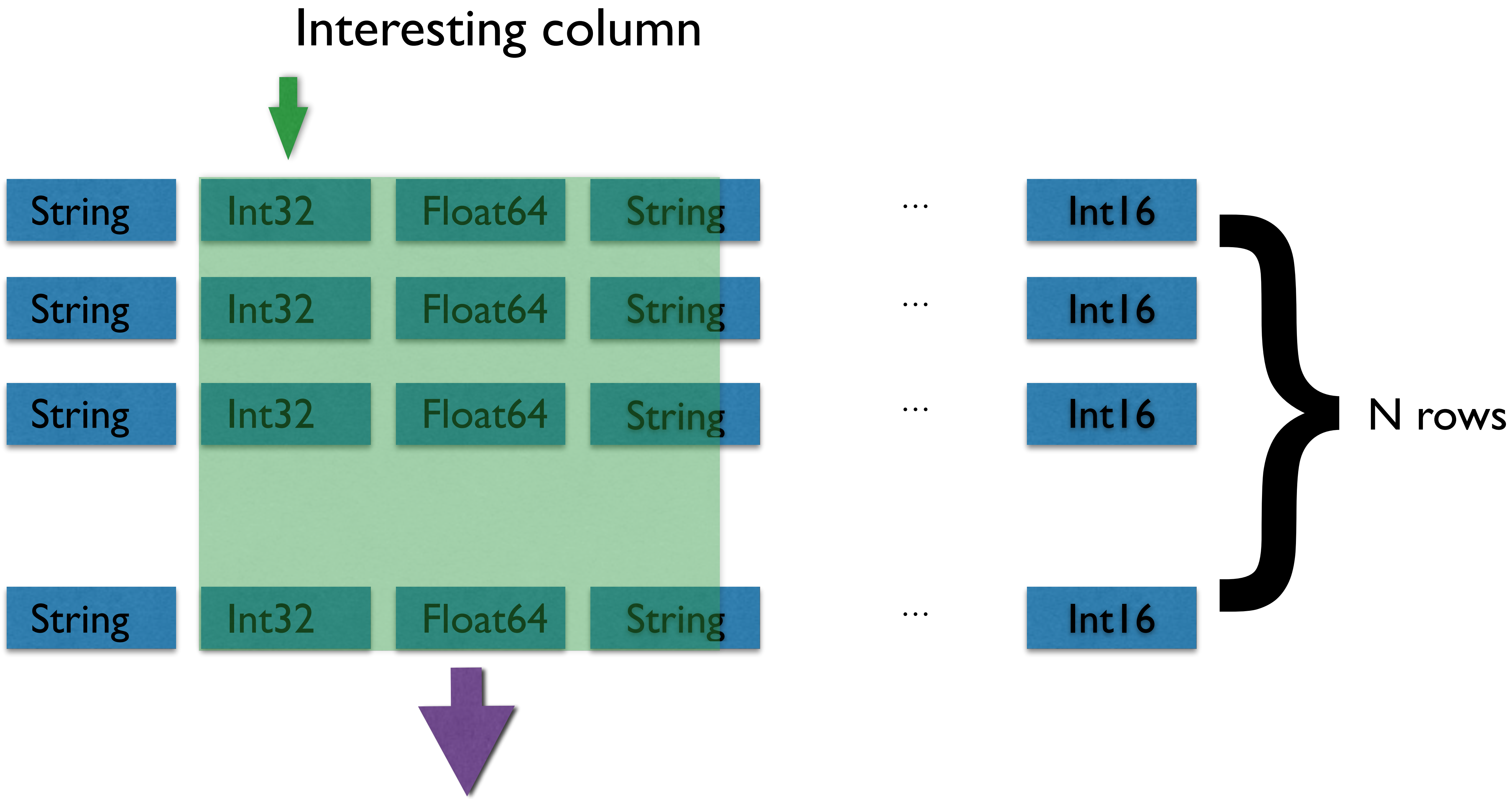


Discontiguous memory

Why Columnar?

- Because it adapts better to newer computer architectures

In-Memory Row-Wise Table (Structured NumPy array)



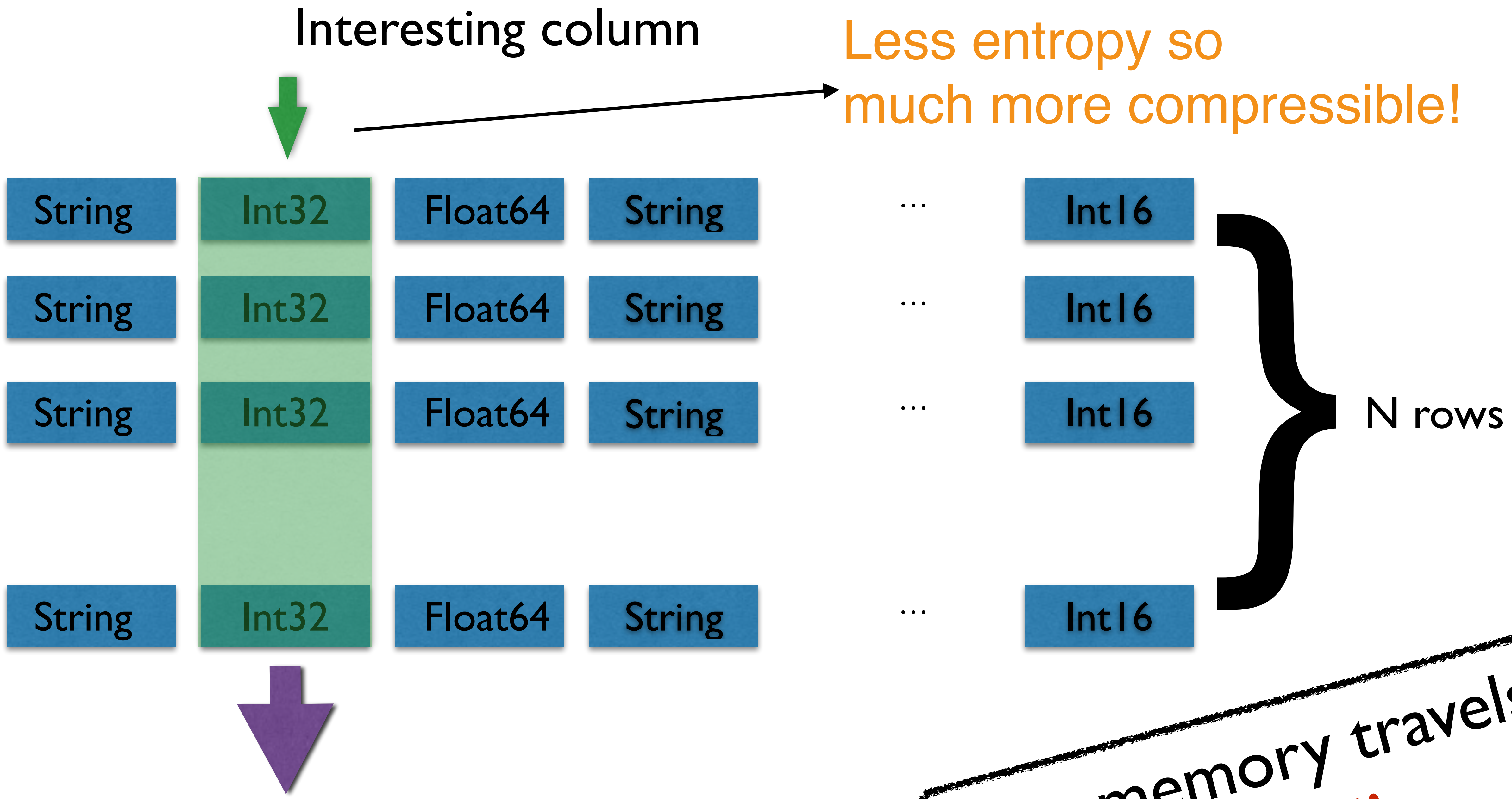
Interesting column

String	Int32	Float64	String	…	Int16
String	Int32	Float64	String	…	Int16
String	Int32	Float64	String	…	Int16
String	Int32	Float64	String	…	Int16

N rows

Interesting Data: N * 4 bytes (Int32)
Actual Data Read: N * 64 bytes (cache line)

In-Memory Column-Wise Table (bcolz *ctable*)



Interesting column

Less entropy so much more compressible!

String	Int32	Float64	String	…	Int16
String	Int32	Float64	String	…	Int16
String	Int32	Float64	String	…	Int16
String	Int32	Float64	String	…	Int16

N rows

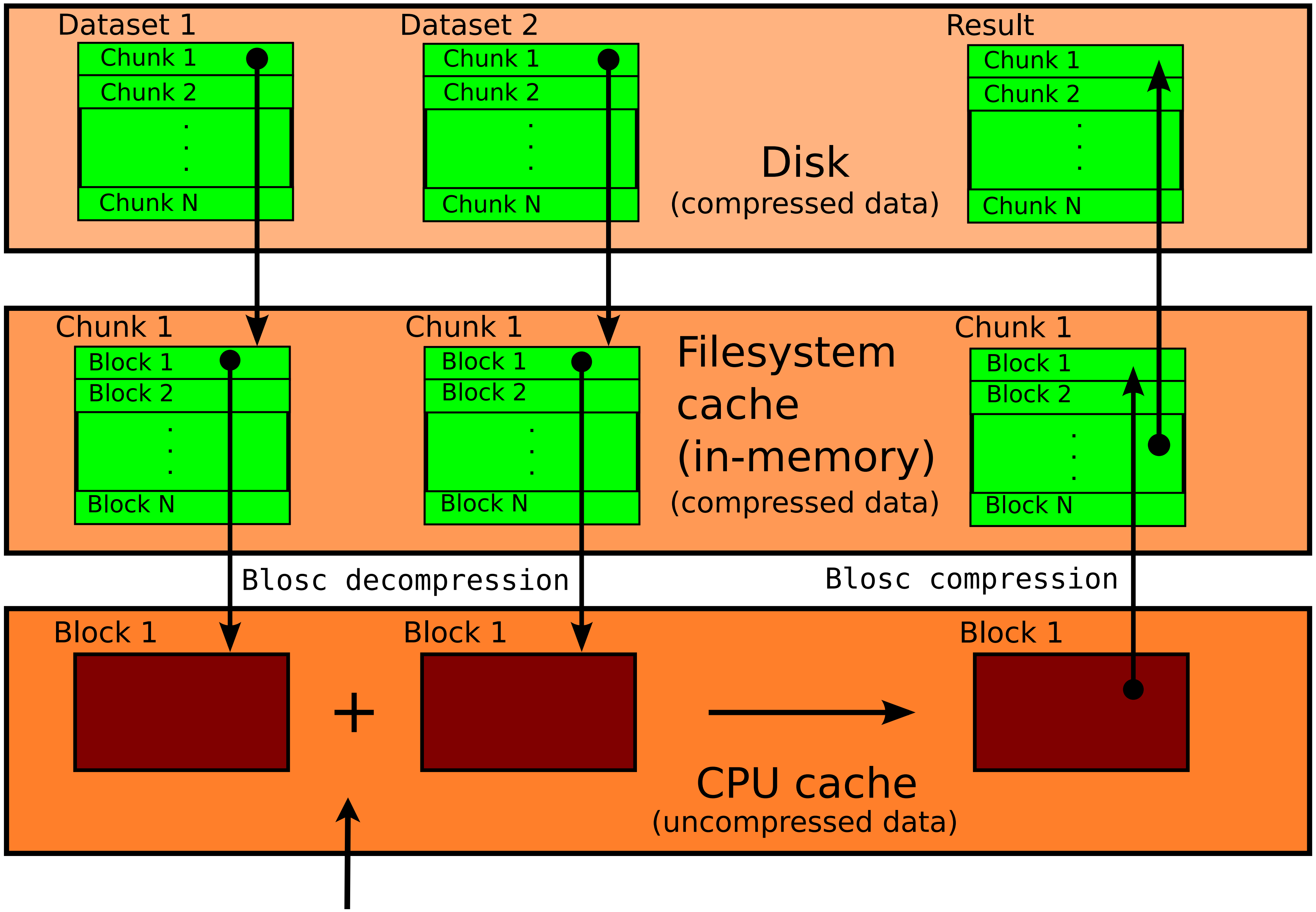
Interesting Data: N * 4 bytes (Int32)
Actual Data Read: N * 4 bytes (Int32)

Less memory travels **to CPU!**

Out-Of-Core Computations

- Due to the addition of the persistency, carray can perform out-of-core computations seamlessly
- Supports different Virtual Machines:
    - Plain Python
    - numexpr (so you can use multicores)
    - Dask (delayed expression tree evaluation)

How bcolz Does Out-Of-Core Computations



Dataset 1

Chunk 1

Chunk 2

Chunk N

Dataset 2

Chunk 1

Chunk 2

Chunk N

Disk

(compressed data)

Result

Chunk 1

Chunk 2

Chunk N

Chunk 1

Block 1

Block 2

Block N

Chunk 1

Block 1

Block 2

Block N

Filesystem cache (in-memory)

(compressed data)

Chunk 1

Block 1

Block 2

Block N

Blosc decompression

Blosc compression

Block 1

Block 1

Block 1

CPU cache

(uncompressed data)

Virtual Machine : Python, numexpr, Dask

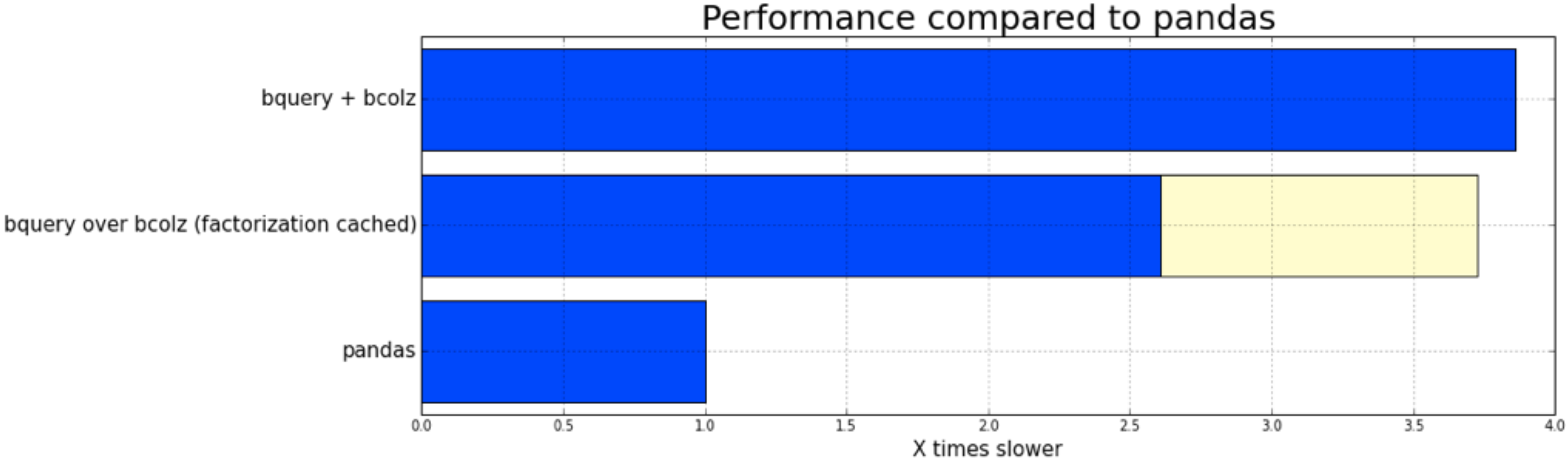
Some Projects Using bcolz

- Visualfabriq's bquery (out-of-core groupby's): https://github.com/visualfabriq/bquery

- Scikit-allel: http://scikit-allel.readthedocs.org/

- Quantopian: http://quantopian.github.io/talks/NeedForSpeed/slides#/

bquery - On-Disk GroupBy



In-memory (pandas) vs on-disk (bquery+bcolz) groupby

"Switching to bcolz enabled us to have a much better scalable architecture yet with near in-memory performance"
— Carst Vaartjes, co-founder visualfabriq