# Continuum Data Analytics Stack

Pycon DE 2013, Köln

Yves Hilpisch
*Continuum Analytics Europe GmbH*
yves@continuum.io
@dyjh

CONTINUUM ANALYTICS

# Agenda

- Big Data and Python
- Architecting for Data
- Continuum's Stack

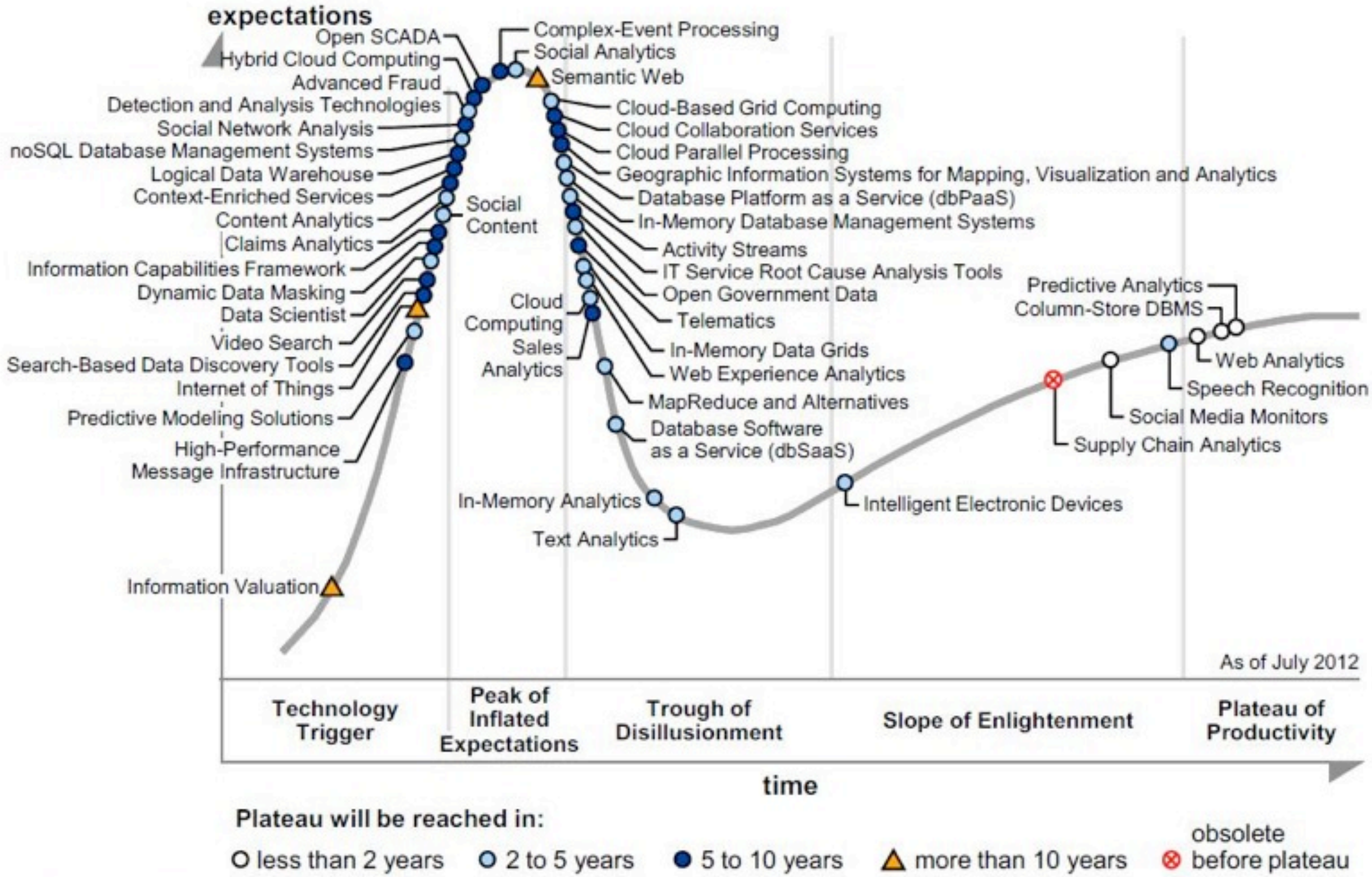
# Big Data and Python

# Origin of “Big Data” Movement

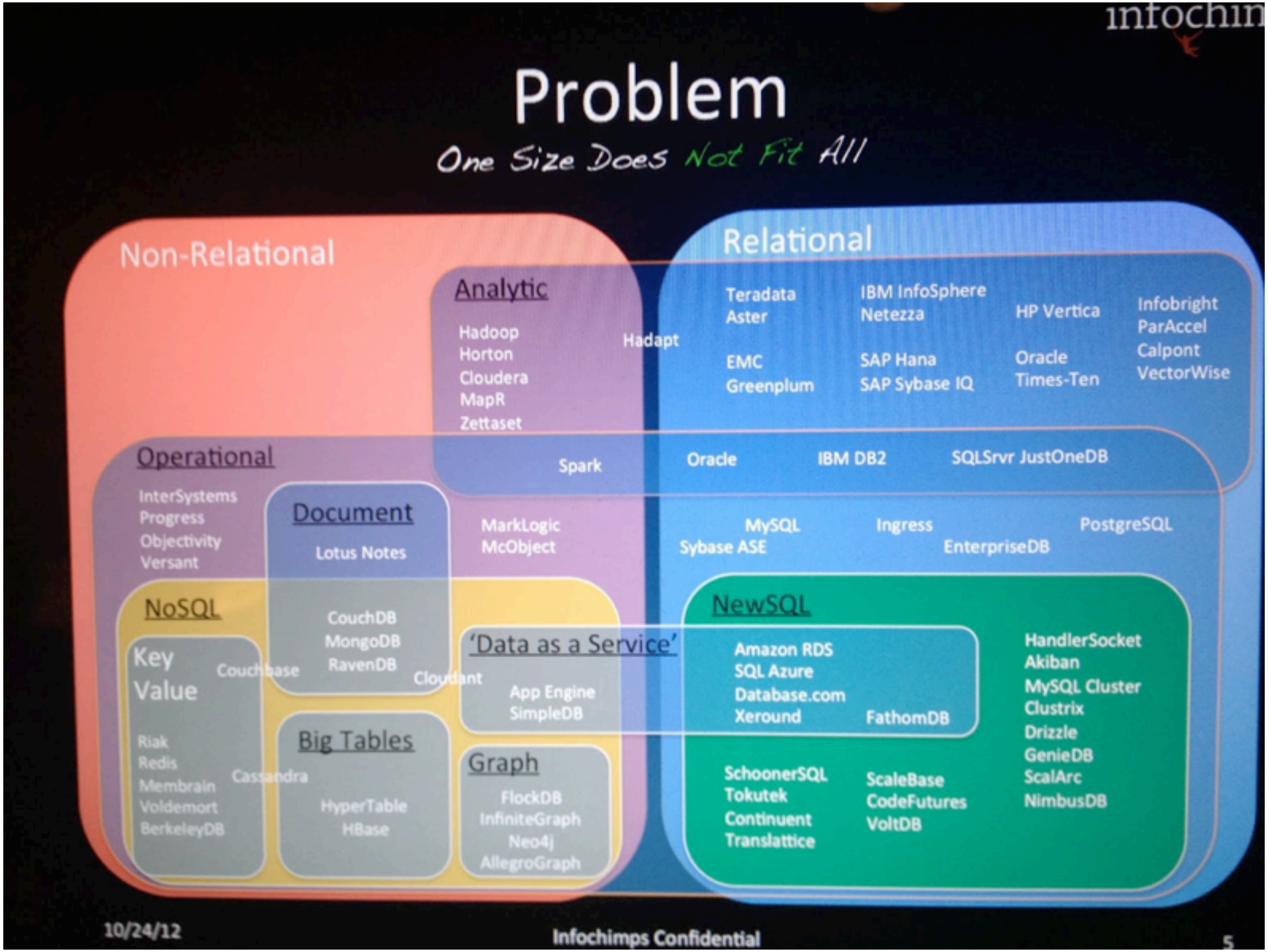
- Storage disruption: plummeting HDD costs, cloud-based storage
  - also: I/O evolution: 10gE SANs, Flash drives
- ETL disruption: Hadoop/Hive/HBase
- Basic analytics & statistics: “counting things”
- Facebook, Twitter, Youtube, Instagram ...

# Big Data (circa 2012)

Figure 1. Hype Cycle for Big Data, 2012




Source: Gartner (July 2012)



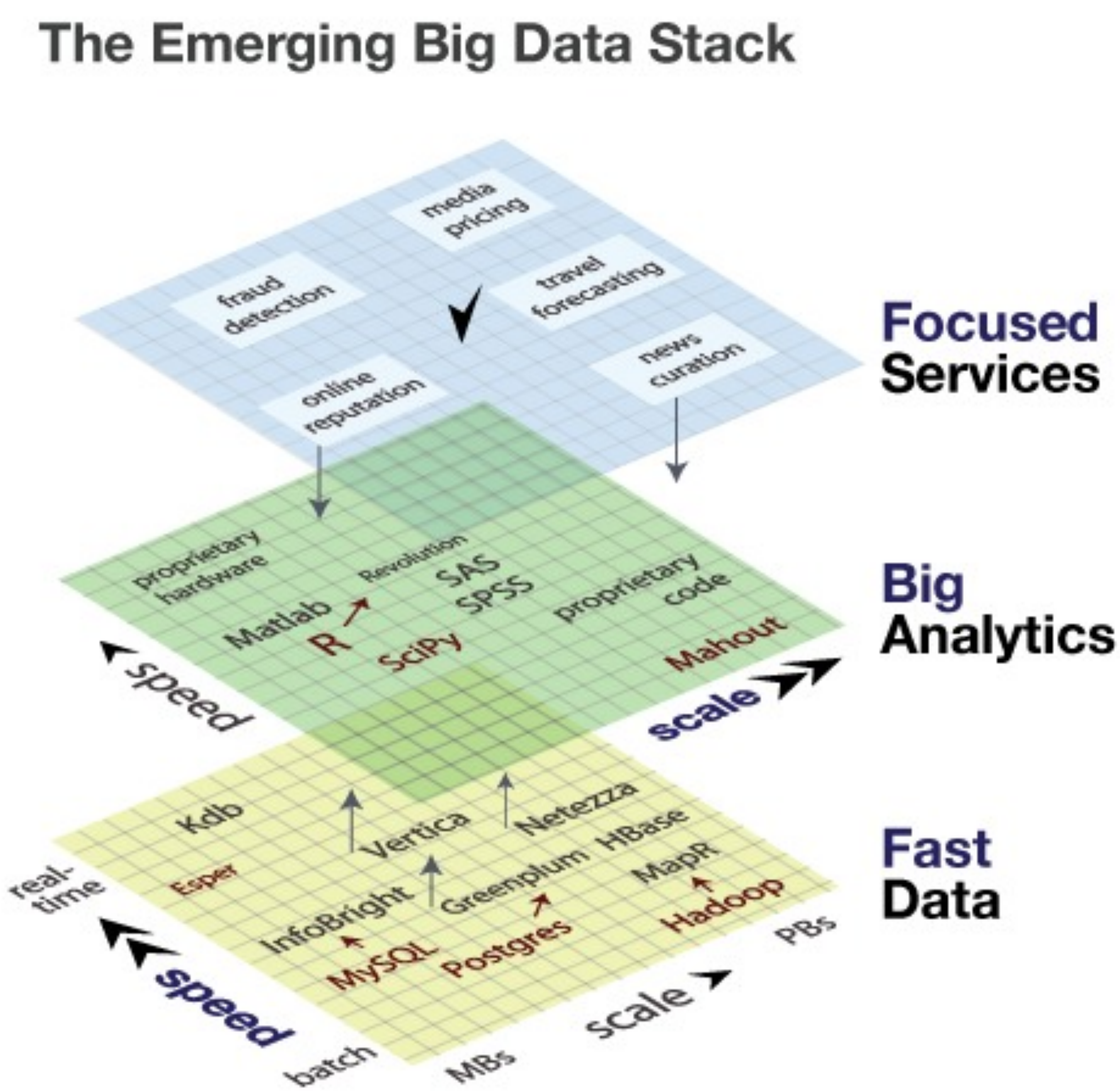

http://techcrunch.com/2012/10/27/big-data-right-now-five-trendy-open-source-technologies/

CONTINUUM ANALYTICS

# The Players

- Data Processing & Low-level infrastructure
- Traditional BI vendors
- New BI startups
- Data-oriented startups
- Analytics-as-a-service
- "Big data" infrastructure platforms (DB & analytical compute as a service)

# Another perspective (2011)



# Observed Trends

- Diversification away from SQL & relational DBs
  - "Messy" data, agile data processing
  - Dynamic schema management
  - Acknowledgement of heterogenous data environment
- Focus on high performance
  - Richer simulations, processing more data
  - Modern hardware revolution (SSDs, GPUs, etc.)
- Advanced visualization
  - Interactive, novel plots
  - Beyond simple reports and dashboards
- Advanced analytics
  - Richer statistical models, Bayesian approaches
  - Machine learning
  - Predictive databases

# Summary Data Trends

- **big data**: be it in terms of volume, complexity or velocity, available data is growing drastically; new technologies, an increasingly connected world, more sophisticated data gathering techniques and devices as well as new cultural attitudes towards social media are among the drivers of this trend
- **real-time economy**: today, decisions have to be made in real-time, business strategies are much shorter lived and the need to cope faster with the ever increasing amount and complexity of decision-relevant data steadily increases

# However, ...

*"Our measurements as well as other recent work shows that the majority of real-world analytic jobs process less than 100 GB of input, but popular infrastructures such as Hadoop/MapReduce were originally designed for petascale processing. We claim that a single "scale-up" server can process each of these jobs and do as well or better than a cluster in terms of performance, cost, power, and server density."*

Raja Appuswamy et al. (2013): "Nobody Ever Got Fired for Buying a Cluster." Microsoft Research, Cambridge UK.

# Architecting for Data

# Data Revolution

"Internet Revolution" True Believer, 1996:
*Businesses that build network-oriented capability into their core will fundamentally outcompete and destroy their competition.*

"Data Revolution" True Believer, 2010:
*Businesses that build data comprehension into their core will destroy their competition over the next 5-10 years*

# Opportunities

- Advanced ML & Predictive DBs will provide transformative insights to nearly every business.

- Mobile & hi-speed connectivity means more dimensions of customer life are being digitized.
  - Every bit of new data makes old data more valuable
  - Analyzing historical data becomes more important

- Developing internal data analysis capability means you can more easily build data products to sell downstream.
  - This is becoming an industry unto itself.

# Technical Challenges

- Hardware & software do not yet make data analysis easy at terabyte scales
- Current analytics are mostly I/O bound. Next generation "advanced" analytics will be compute bound (simulations, distributed LinAlg). *Efficiency matters*.
- Reproducible analytical environment.
- Library & language choices can add "air gaps" between domain expert and analytical infrastructure.

# Business Challenges

- Data exploration is new discipline for most businesses.
- Balancing agility & process for data-oriented processes and analytical libraries.
- Bad data architecture will generally not cause catastrophic failures.
- Instead, will erode your ability to compete.

*It's hard to know when you are sucking.*

# Data Matters

- Data has mass.
- Scalability requires minimizing data-movement (only as necessary).
- Deep/Advanced Analytics needs full computing stack, as accessible as SQL and Excel.
- Data should only move when it has to (to communicate results, to replicate, to back-up) not because the technology doesn't allow access.

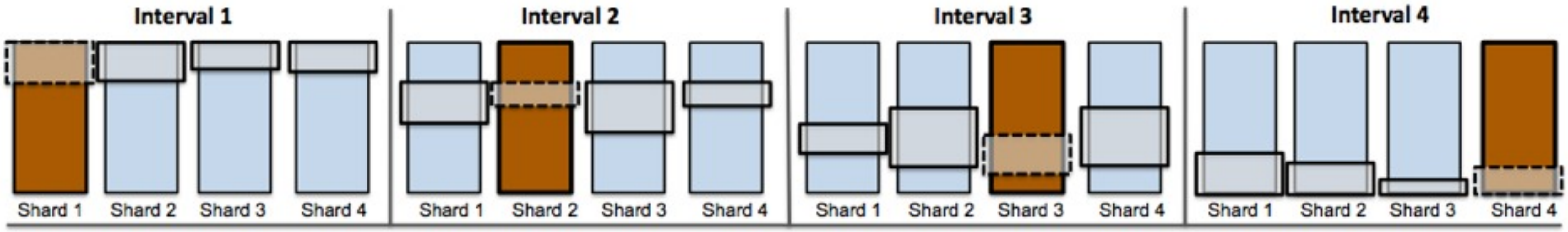
# Algorithms Matter

...a Mac Mini running GraphChi can analyze Twitter's social graph from 2010—which contains 40 million users and 1.2 billion connections—in 59 minutes. "The previous published result on this problem took 400 minutes using a cluster of about 1,000 computers," Guestrin says.

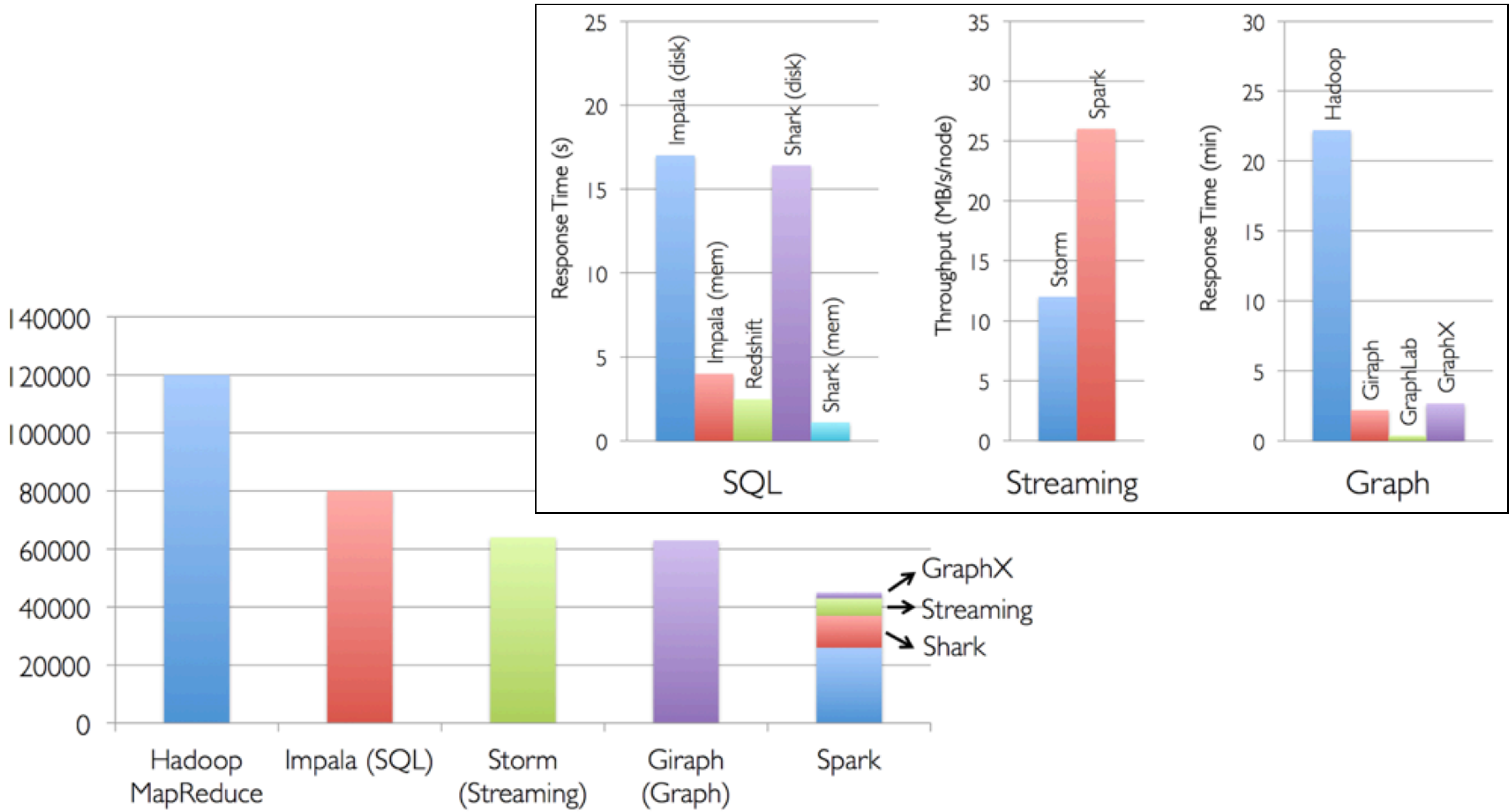
-- *MIT Tech Review*

GraphLab

"...Spark, running on a cluster of 50 machines (100 CPUs) runs five iterations of Pagerank on the twitter-2010 in 486.6 seconds. GraphChi solves the same problem in less than double of the time (790 seconds), with only 2 CPUs."

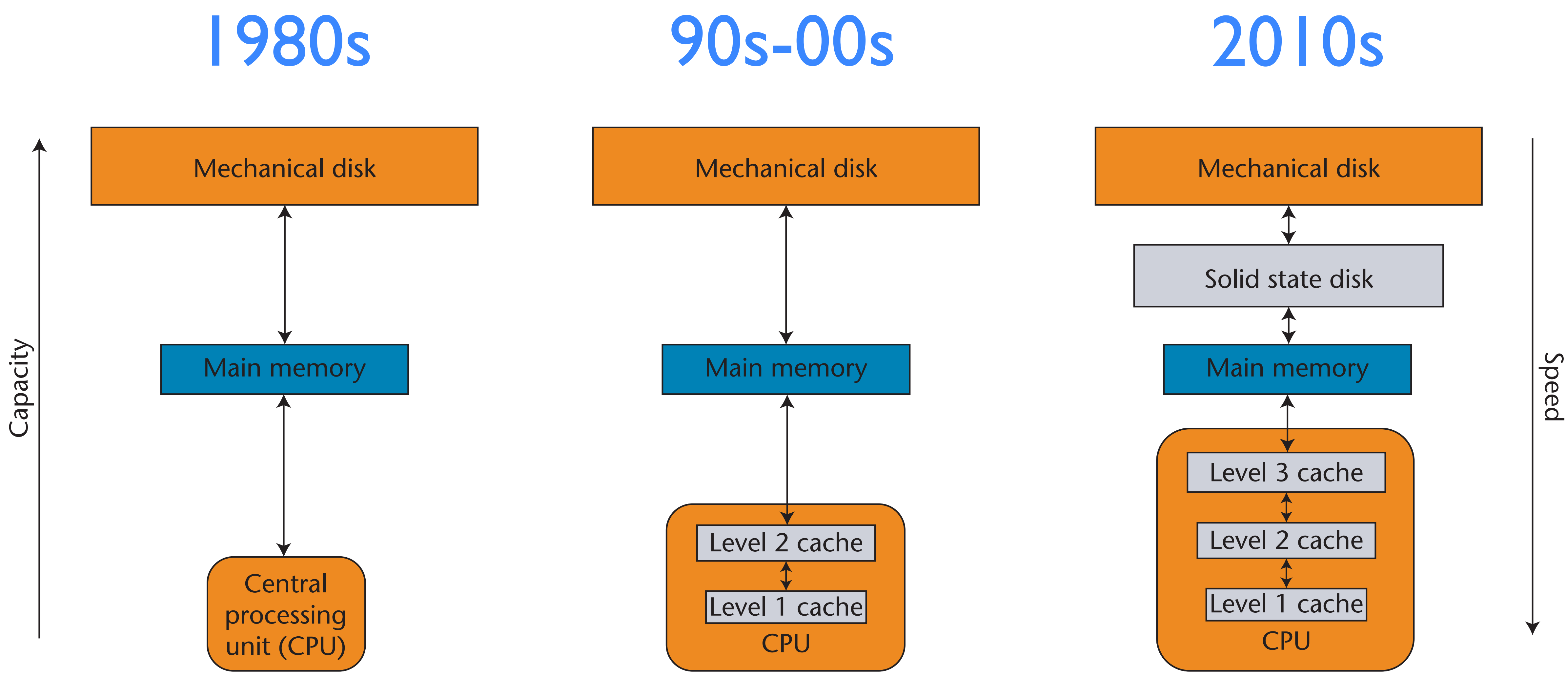


# Berkeley Data Stack (BDAS)

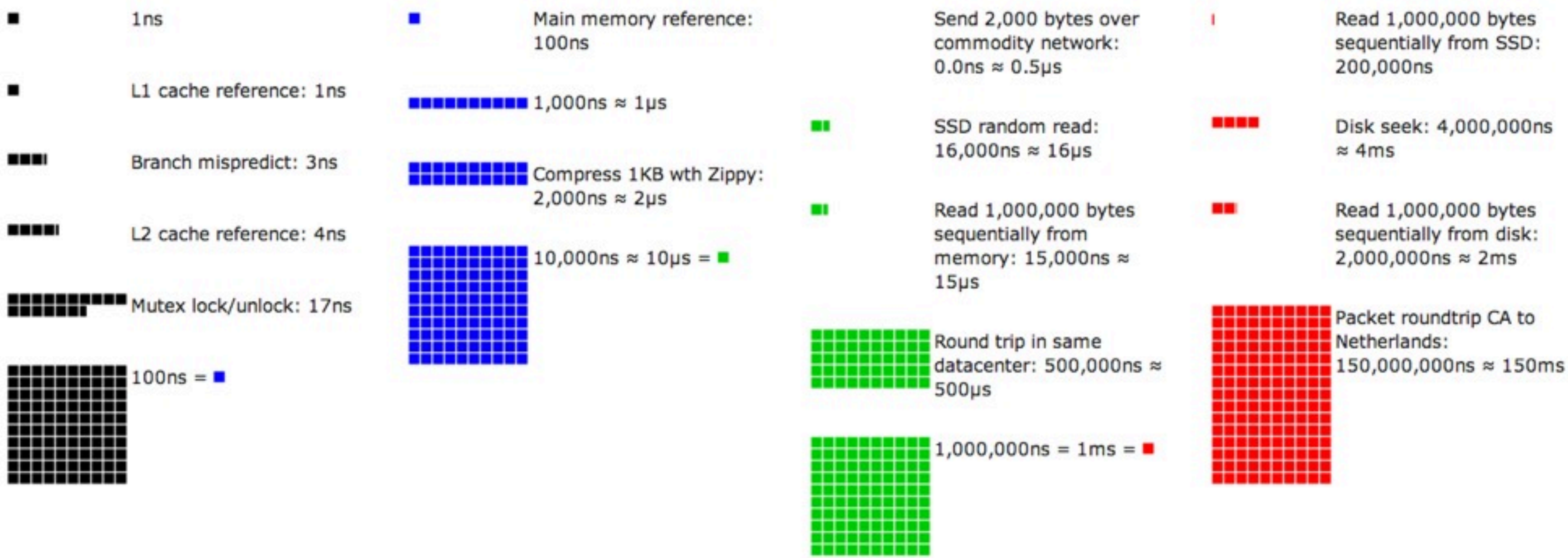



CONTINUUM ANALYTICS

# Memory Matters



# Speed Matters



# Jeff Hammerbacher's Advice

- Instrument everything
- Put all your data in one place
- Data first, questions later
- Store first, structure later (often the data model is dependent on the analysis you'd like to perform)
- Keep raw data forever
- Let everyone party on the data
- Introduce tools to support the whole research cycle (think of the scope of the product as the entire cycle, not just the container)
- Modular and composable infrastructure

# Architecting for Data

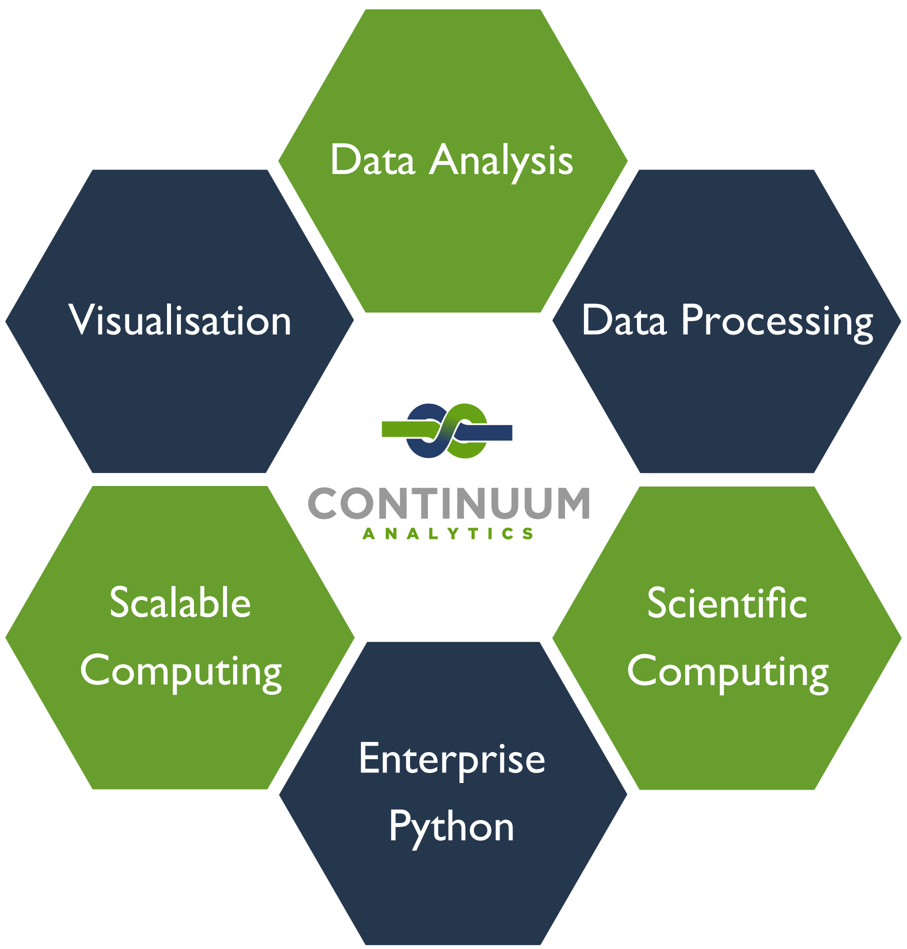
Data exploration as the central task.

Data visualization as a first-class citizen.

Enable agility.

# Continuum´s Stack

# Continuum Analytics




## Domains

- Finance
- Geophysics
- Defense
- Advertising & Web Analytics
- Scientific Computing

## Technologies

- Array/Columnar data processing
- Distributed computing, HPC
- GPU and new vector hardware
- Machine learning, predictive analytics
- Interactive Visualization

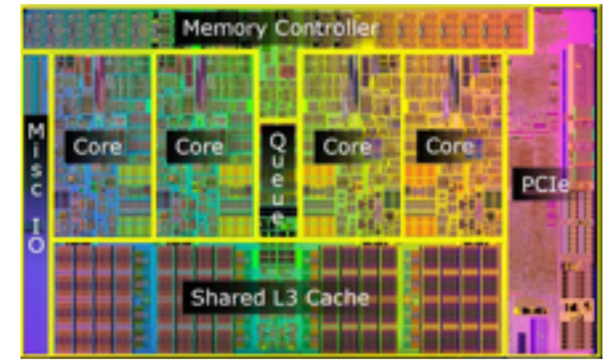
# Mission

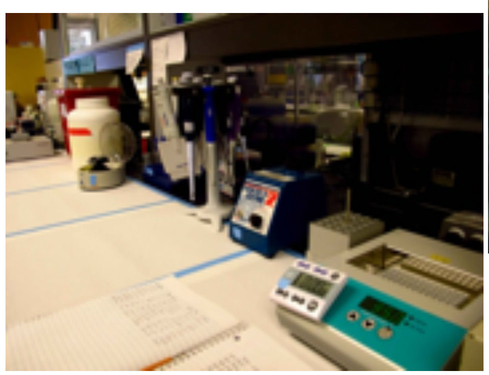
To revolutionize data analytics and visualization by moving high-level Python code and domain expertise closer to data. This vision rests on four pillars:

- **simplicity**: advanced, powerful analytics, accessible to domain experts and business users via a simplified programming paradigm
- **interactivity**: interactive analysis and visualization of massive data sets
- **collaboration**: collaborative, shareable analysis (data, code, results, graphics)
- **scale**: out-of-core, distributed data processing

# Big Picture

Empower domain experts with high-level tools that exploit modern hardware



**Array Oriented Computing**

# Projects

Blaze: High-performance Python library for modern vector computing, distributed and streaming data

Numba: Vectorizing Python compiler for multicore and GPU, using LLVM

Bokeh: Interactive, grammar-based visualization system for large datasets

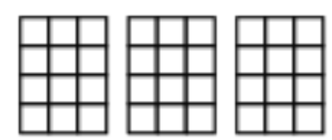
*Common theme:* High-level, expressive language for domain experts; innovative compilers & runtimes for efficient, powerful data transformation

# Blaze Objectives

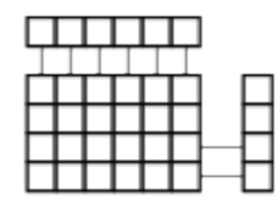
- Flexible descriptor for tabular and semi-structured data

- Seamless handling of:
- On-disk/Out-of-core
- Streaming data
- Distributed data

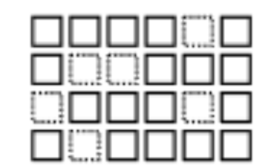
- Uniform treatment of:
- “arrays of structures” and “structures of arrays”
- missing values
- “ragged” shapes
- categorical types
- computed columns



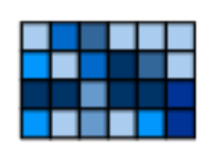
Chunked Arrays



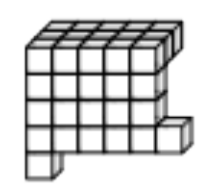
Synthetic Dimensions



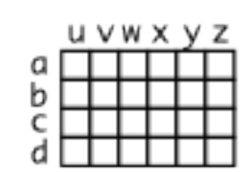
Missing Values



Type Heterogeneity



Shape Heterogeneity

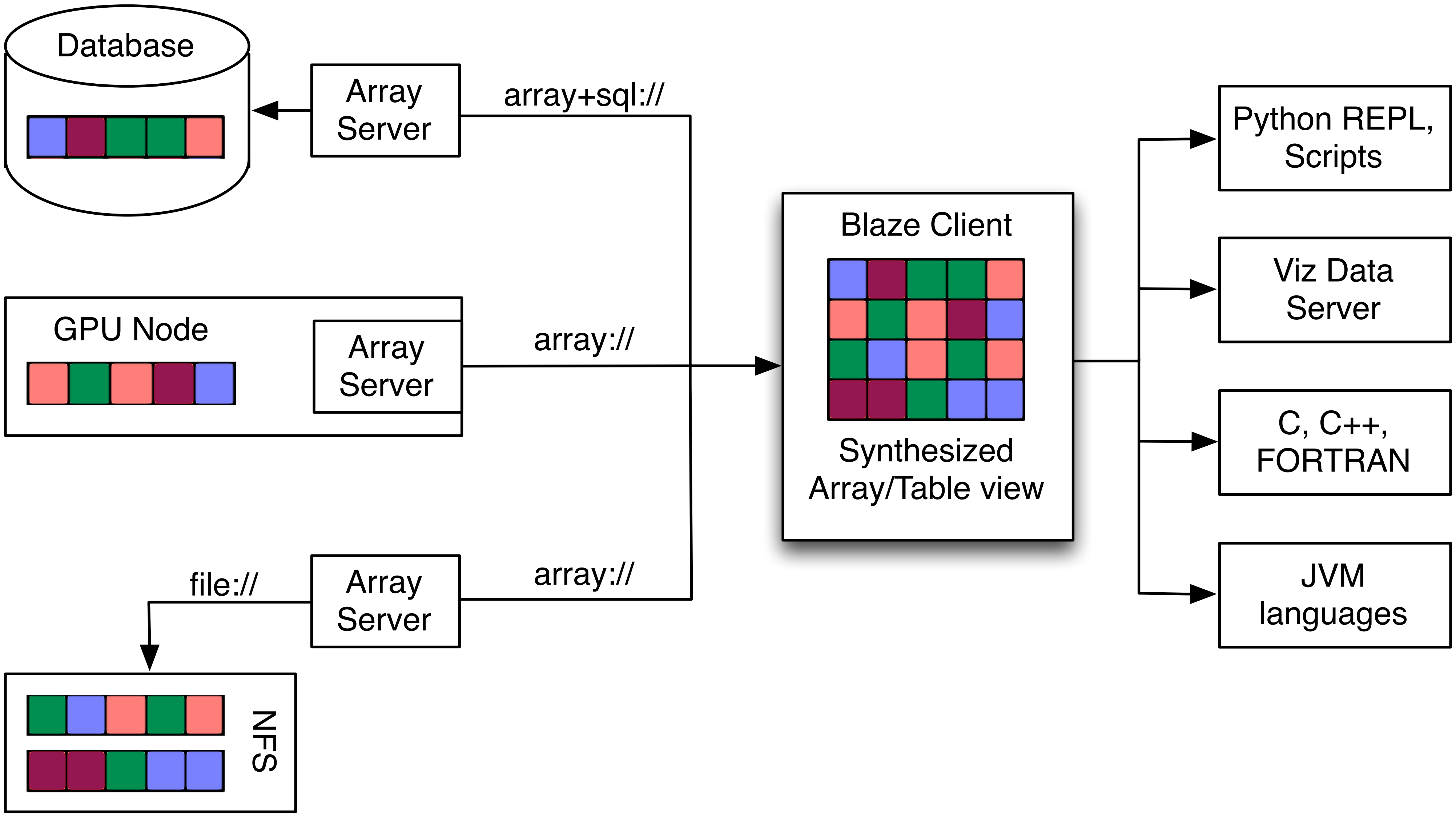



Labeled Arrays

# Blaze Status

- DataShape type grammar
- NumPy-compatible C++ calculation engine (DyND)
- Synthesis of array function kernels (via LLVM)
- Fast time series routines (dynamic time warping for pattern matching)
- Array Server prototype
- BLZ columnar storage format
- 0.2 current release, working on 0.3 ...

# Schematic



Database
Array Server
array+sql://
GPU Node
Array Server
array://
Blaze Client
Synthesized Array/Table view
Python REPL, Scripts
Viz Data Server
C, C++, FORTRAN
JVM languages
file://
Array Server
array://
NFS

# Kiva: Array Server

## Data Shape + Raw JSON = Web Service

```
type KivaLoan = {
  id: int64;
  name: string;
  description: {
    languages: var, string(2);
    texts: json # map<string(2), string>;
  };
  status: string; # LoanStatusType;
  funded_amount: float64;
  basket_amount: json; # Option(float64);
  paid_amount: json; # Option(float64);
  image: {
    id: int64;
    template_id: int64;
  };
  video: json;
  activity: string;
  sector: string;
  use: string;
  delinquent: bool;
  location: {
    country_code: string(2);
    country: string;
    town: json; # Option(string);
    geo: {
      level: string; # GeoLevelType
      pairs: string; # latlong
      type: string; # GeoTypeType
    }
  };
  ....
```

{"id":200533,"name":"Miawand Group","description":{"languages":["en"],"texts":{"en":"Ozer is a member of the Miawand Group. He lives in the 16th district of Kabul, Afghanistan. He lives in a family of eight members. He is single, but is a responsible boy who works hard and supports the whole family. He is a carpenter and is busy working in his shop seven days a week. He needs the loan to purchase wood and needed carpentry tools such as tape measures, rulers and so on.\r\n \r\nHe hopes to make progress through the loan and he is confident that will make his repayments on time and will join for another loan cycle as well. \r\n\r\n"}},"status":"paid","funded_amount":925,"basket_amount":null,"paid_amount":925,"image":{"id":539726,"template_id":1},"video":null,"activity":"Carpentry","sector":"Construction","use":"He wants to buy tools for his carpentry shop","delinquent":null,"location":{"country_code":"AF","country":"Afghanistan","town":"Kabul Afghanistan","geo":{"level":"country","pairs":"33 65","type":"point"}},"partner_id":34,"posted_date":"2010-05-13T20:30:03Z","planned_expiration_date":null,"loan_amount":925,"currency_exchange_loss_amount":null,"borrowers":[{"first_name":"Ozer","last_name":"","gender":"M","pictured":true},{"first_name":"Rohaniy","last_name":"","gender":"M","pictured":true},{"first_name":"Samem","last_name":"","gender":"M","pictured":true}],"terms":{"disbursal_date":"2010-05-13T07:00:00Z","disbursal_currency":"AFN","disbursal_amount":42000,"loan_amount":925,"local_payments":[{"due_date":"2010-06-13T07:00:00Z","amount":4200},{"due_date":"2010-07-13T07:00:00Z","amount":4200},{"due_date":"2010-08-13T07:00:00Z","amount":4200},{"due_date":"2010-09-13T07:00:00Z","amount":4200},{"due_date":"2010-10-13T07:00:00Z","amount":4200},{"due_date":"2010-11-13T08:00:00Z","amount":4200},{"due_date":"2010-12-13T08:00:00Z","amount":4200},{"due_date":"2011-01-13T08:00:00Z","amount":4200},{"due_date":"2011-02-13T08:00:00Z","amount":4200},{"due_date":"2011-03-13T08:00:00Z","amount":4200}],"scheduled_payments": ...


2.9gb of JSON => network-queryable array: ~5 minutes

CONTINUUM ANALYTICS

# Akamai Dataset ETL

| | Hive | Python script |
|---|---|---|
| Hardware | 8x 16 core, 2 GHz (128 cores) | 1x 8 core, 2.2 GHz |
| Memory | RAM: 8x 382 GB<br>HDD: 8x 15k rpm | RAM: 144 GB<br>HDD: 2x 7200rpm |
| Time (traceroute) | 5 hrs, 635M routes | 11 hrs, 113M routes |
| Routes/hr/Ghz | 496k | 584k |

- Python performs ~18% better with almost no optimization
- resulting IPMap can be used for realtime, online query and aggregation

# Querying Traceroute in BLZ format

Meant for dealing with Big Data
(RAM consumption is extremely low)

| | 1k Random | | Full Scan | |
|---|---|---|---|---|
| | Time | RAM | Time | RAM |
| BLZ (disk) | 3.5s | 0.04mb | 2.9s | 8mb |
| BLZ (mem) | 2.37s | 210mb | 2.4s | 210mb |
| NPY (memmap) | 0.24s | 0.2mb | 0.23s | 602mb |
| NumPy (mem) | .13s | 603mb | 0.23s | 603mb |

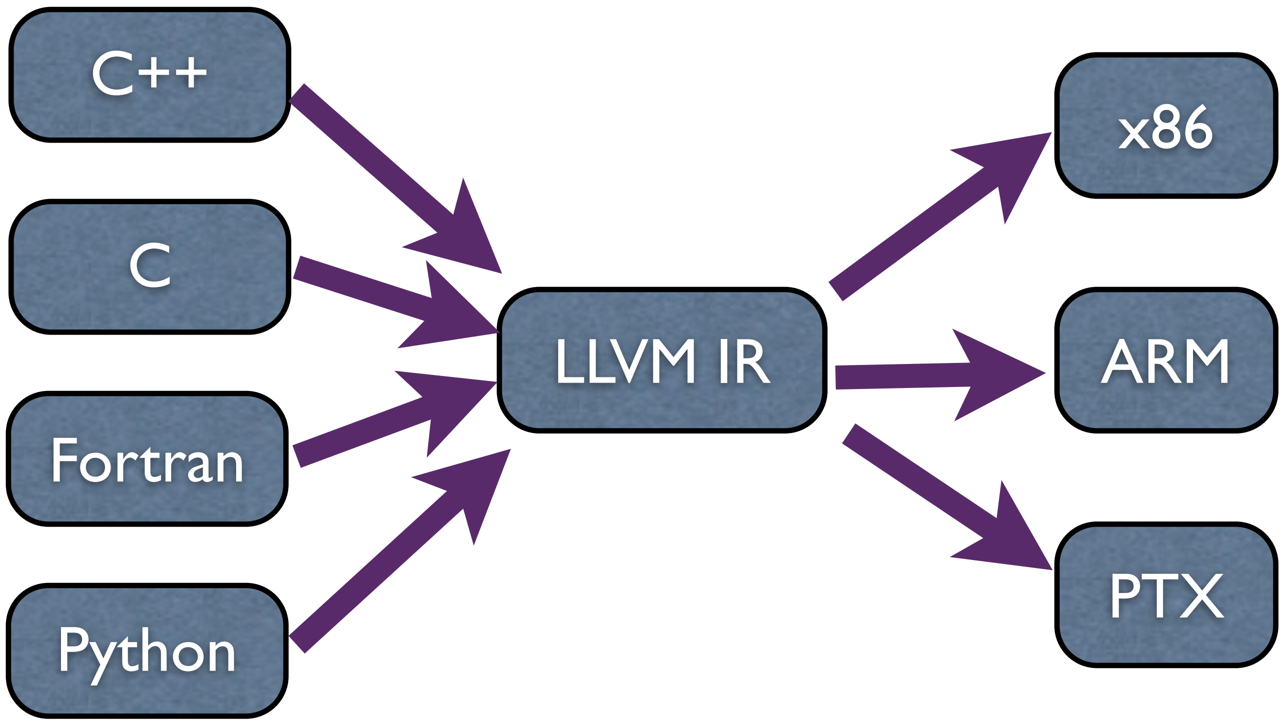
# Numba

- Just-in-time, dynamic compiler for Python
- Optimize data-parallel computations at call time, to take advantage of local hardware configuration
- Compatible with NumPy, Blaze
- Leverage LLVM ecosystem:
  - Optimization passes
  - Inter-op with other languages
  - Variety of backends (e.g. CUDA for GPU support)

# Numba

```cpp
// mean(vector)
template<typename P_numtype>
inline
BZ_FLOATTYPE(BZ_SUMTYPE(P_numtype)) mean(const Vector<P_numtype>& x)
{
    BZPRECONDITION(x.length() > 0);

    typedef BZ_FLOATTYPE(BZ_SUMTYPE(P_numtype)) T_floattype;
    return _bz_vec_sum(x._bz_asVecExpr()) / (T_floattype) x.length();
}
```

```python
@autojit
def sum2d(arr):
    M, N = arr.shape
    result = 0.0
    for i in range(M):
        for j in range(N):
            result += arr[i,j]
    return result
```




Numba turns Python into a “compiled language”

CONTINUUM
ANALYTICS

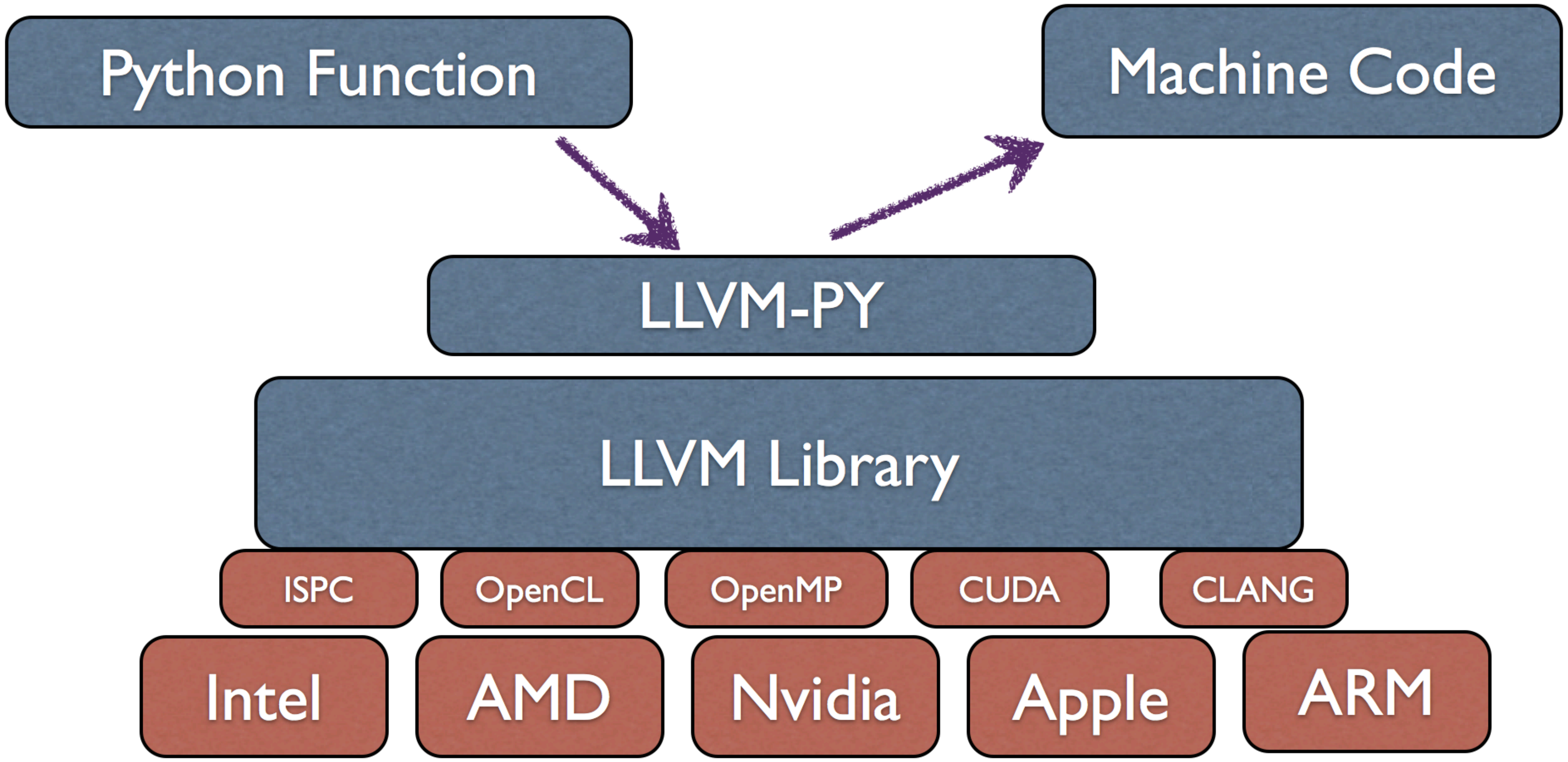
# Example

```python
@numba.autojit
def simple():
    total = 0.0
    for i in range(9999):
        for j in range(1,9999):
            total += (i / j)
    return total
```

Numba

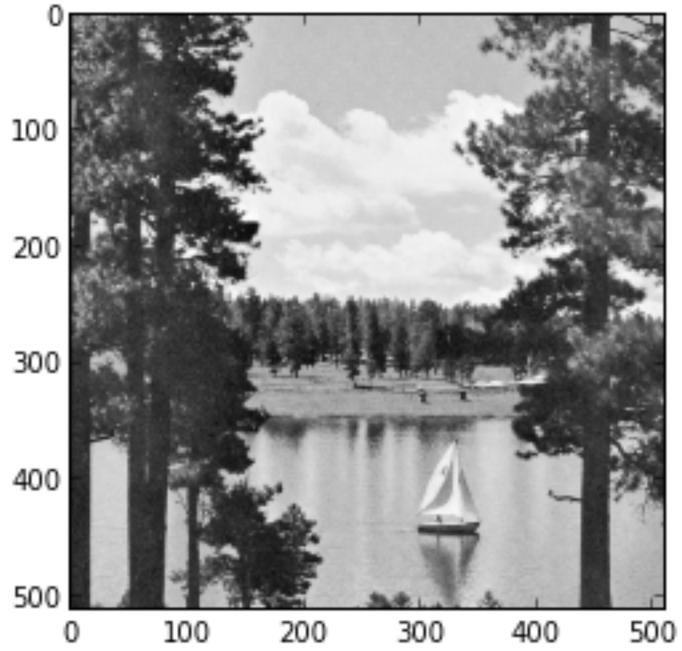
```llvm
define double @__numba_specialized_0___main___2E_simple() nounwind readnone {
entry:
  br label %"for_condition_7:17.preheader"

"for_condition_7:17.preheader":                    ; preds = %"exit_for_7:8", %entry
  %total_28 = phi double [ 0.000000e+00, %entry ], [ %3, %"exit_for_7:8" ]
  %storemerge7 = phi i64 [ 0, %entry ], [ %0, %"exit_for_7:8" ]
  br label %"loop_body_8:12"

"exit_for_6:4":                                    ; preds = %"exit_for_7:8"
  ret double %3

"exit_for_7:8":                                    ; preds = %"loop_body_8:12"
  %0 = add i64 %storemerge7, 1
  %exitcond9 = icmp eq i64 %0, 9999
  br i1 %exitcond9, label %"exit_for_6:4", label %"for_condition_7:17.preheader"

"loop_body_8:12":                                  ; preds = %"loop_body_8:12", %"for_condition_7:17.prel
  %lsr.iv = phi i64 [ %lsr.iv.next, %"loop_body_8:12" ], [ 1, %"for_condition_7:17.preheader" ]
  %total_36 = phi double [ %total_28, %"for_condition_7:17.preheader" ], [ %3, %"loop_body_8:12" ]
  %1 = sdiv i64 %storemerge7, %lsr.iv
  %2 = sitofp i64 %1 to double
  %3 = fadd double %total_36, %2
  %lsr.iv.next = add i64 %lsr.iv, 1
  %exitcond = icmp eq i64 %lsr.iv.next, 9999
  br i1 %exitcond, label %"exit_for_7:8", label %"loop_body_8:12"
}
```

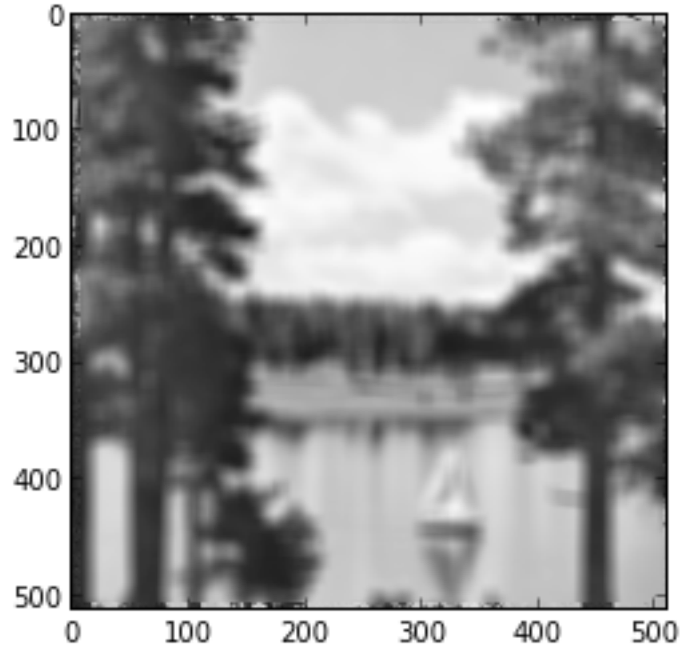
# LLVM-based architecture



# Image Processing

~1500x speed-up




```
@jit('void(f8[:,:],f8[:,:],f8[:,:])')
def filter(image, filt, output):
    M, N = image.shape
    m, n = filt.shape
    for i in range(m//2, M-m//2):
        for j in range(n//2, N-n//2):
            result = 0.0
            for k in range(m):
                for l in range(n):
                    result += image[i+k-m//2,j+l-n//2]*filt[k, l]
            output[i,j] = result
```

CONTINUUM ANALYTICS

# Example: Mandelbrot Vectorized

```
from numbapro import vectorize

sig = 'uint8(uint32, f4, f4, f4, f4, uint32, uint32,
uint32)'

@vectorize([sig], target='gpu')
def mandel(tid, min_x, max_x, min_y, max_y, width,
height, iters):
    pixel_size_x = (max_x - min_x) / width
    pixel_size_y = (max_y - min_y) / height

    x = tid % width
    y = tid / width

    real = min_x + x * pixel_size_x
    imag = min_y + y * pixel_size_y

    c = complex(real, imag)
    z = 0.0j

    for i in range(iters):
        z = z * z + c
        if (z.real * z.real + z.imag * z.imag) >= 4:
            return i
    return 255
```

## Tesla S2050

| Kind | Time | Speed-up |
|---|---|---|
| Python | 263,6 | 1.0x |
| CPU | 2,639 | 100x |
| GPU | 0,1676 | 1573x |

# Example: N-Body Simulation

- Simulation of movement of N bodies (space objects, particles)
- Loop-heavy algorithm to calculate the interactions between all bodies
  - **Pure Python** 70 sec
  - **NumPy** 0.718 sec (= 97x speed-up)
  - **Numba** 0.105 sec (= 7x speed-up = 670 x total)

http://hilpisch.com/Continuum_N_Body_Simulation_Numba_27072013.html

# Bokeh

- Language-based (instead of GUI) visualization system
  - High-level expressions of data binding, statistical transforms, interactivity and linked data
  - Easy to learn, but expressive depth for power users
- Interactive
  - Data space configuration as well as data selection
  - Specified from high-level language constructs
- Web as first class interface target
- Support for large datasets via intelligent downsampling ("abstract rendering")

# Bokeh

Inspirations:

- Chaco: interactive, viz pipeline for large data
- Protovis & Stencil :
  Binding visual Glyphs to data and expressions
- ggplot2: faceting, statistical overlays

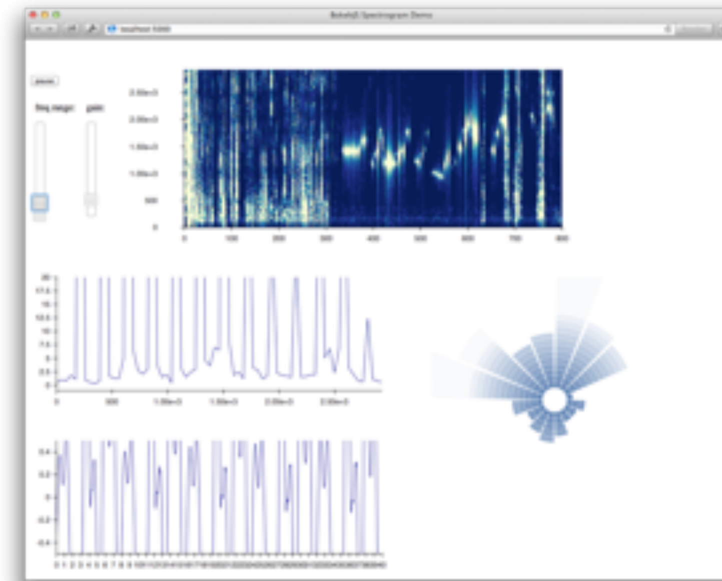
Design goal:

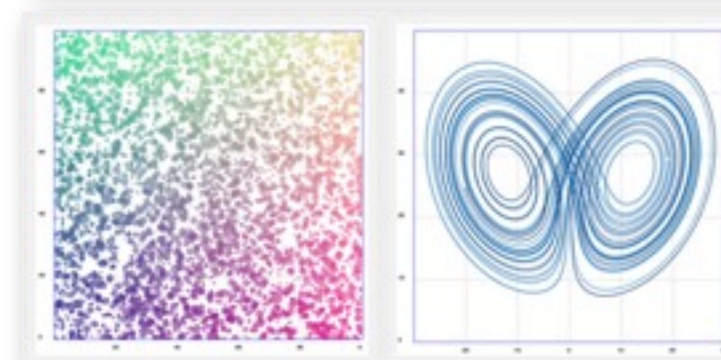
Accessible, extensible, interactive plotting for the web ...

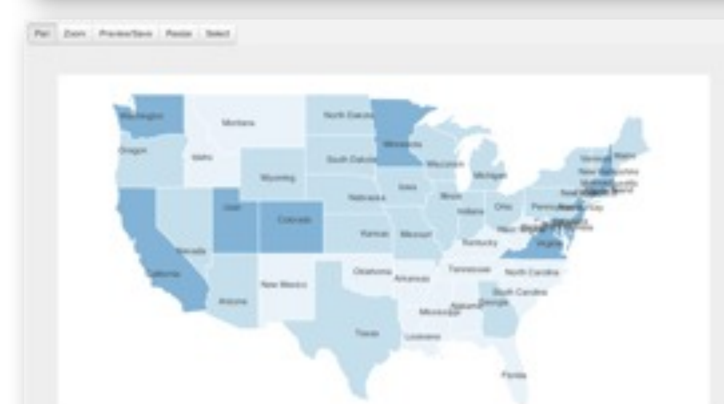
... even for non-Javascript programmers

# Bokeh & BokehJS Demos

- BokehJS demos
- Audio Spectrogram
- Bokeh Examples
  - Low-level Python interface
  - IPython Notebook integration
  - ggplot example



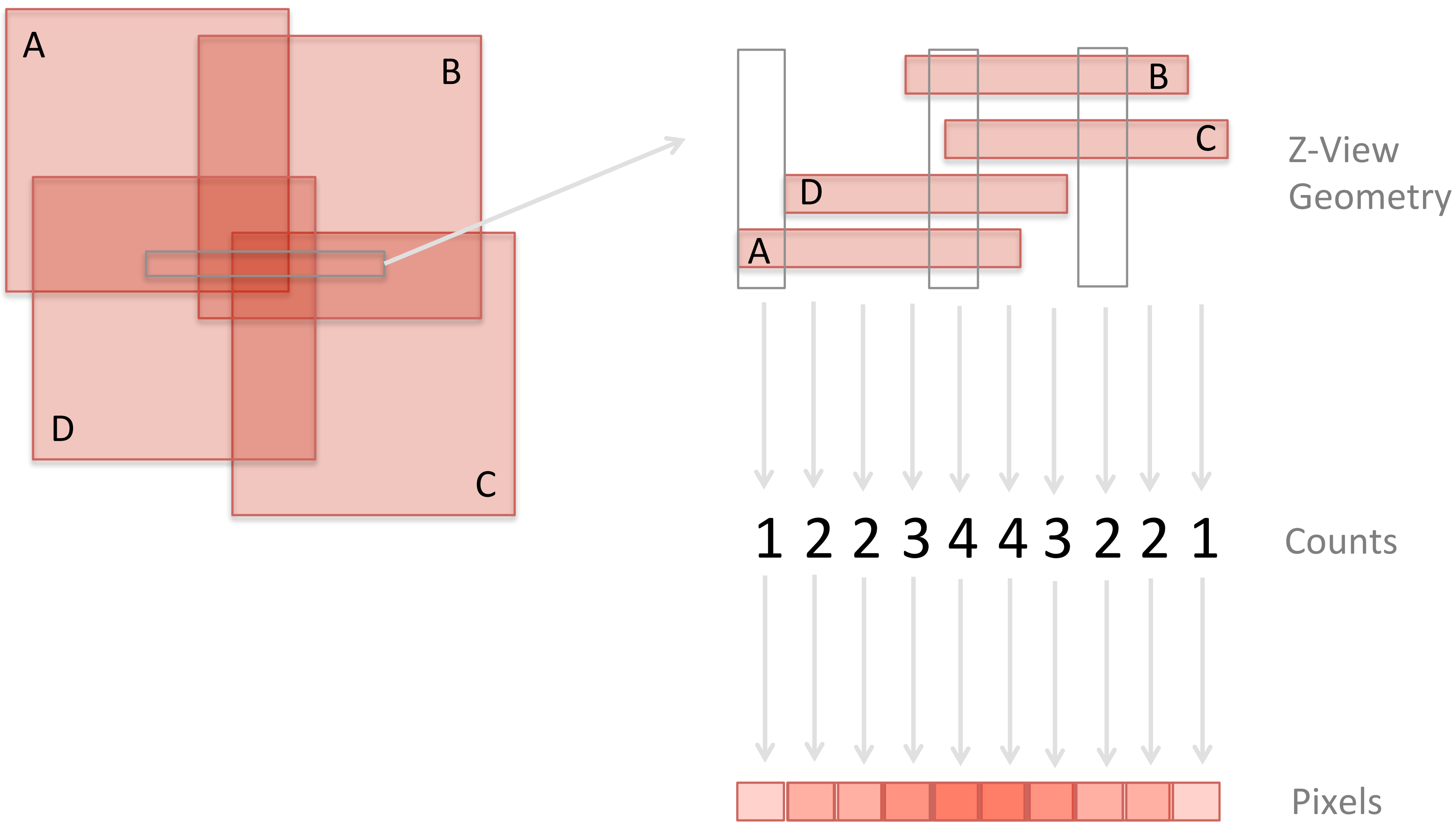




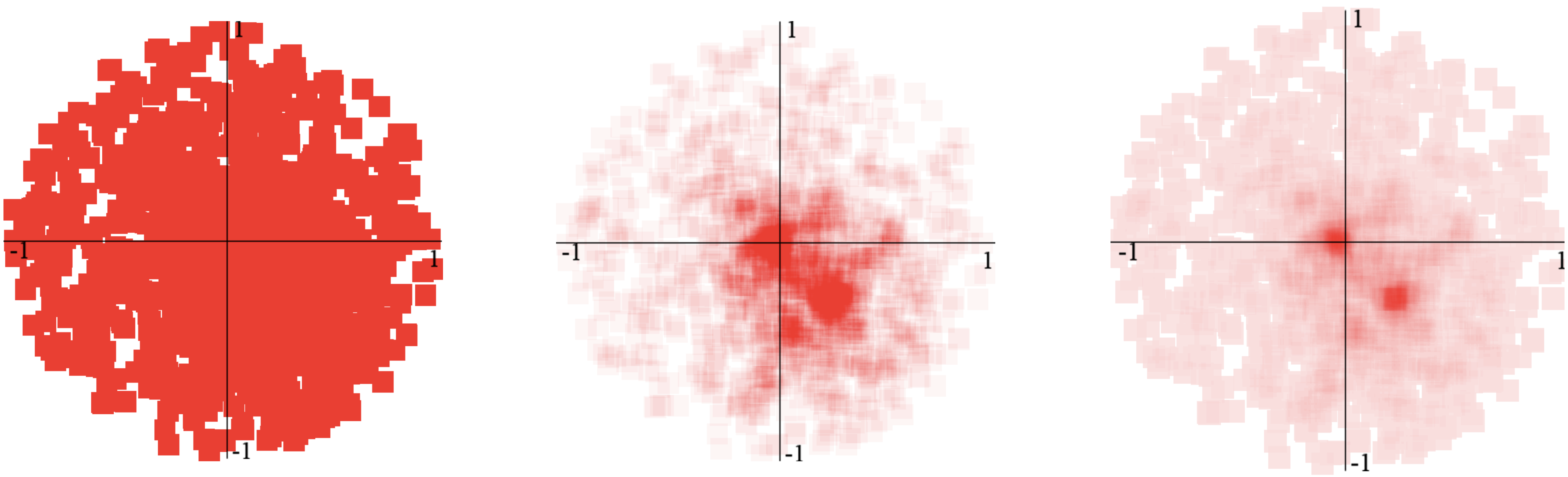
# Abstract Rendering

## Pixels are Bins…

and always have been

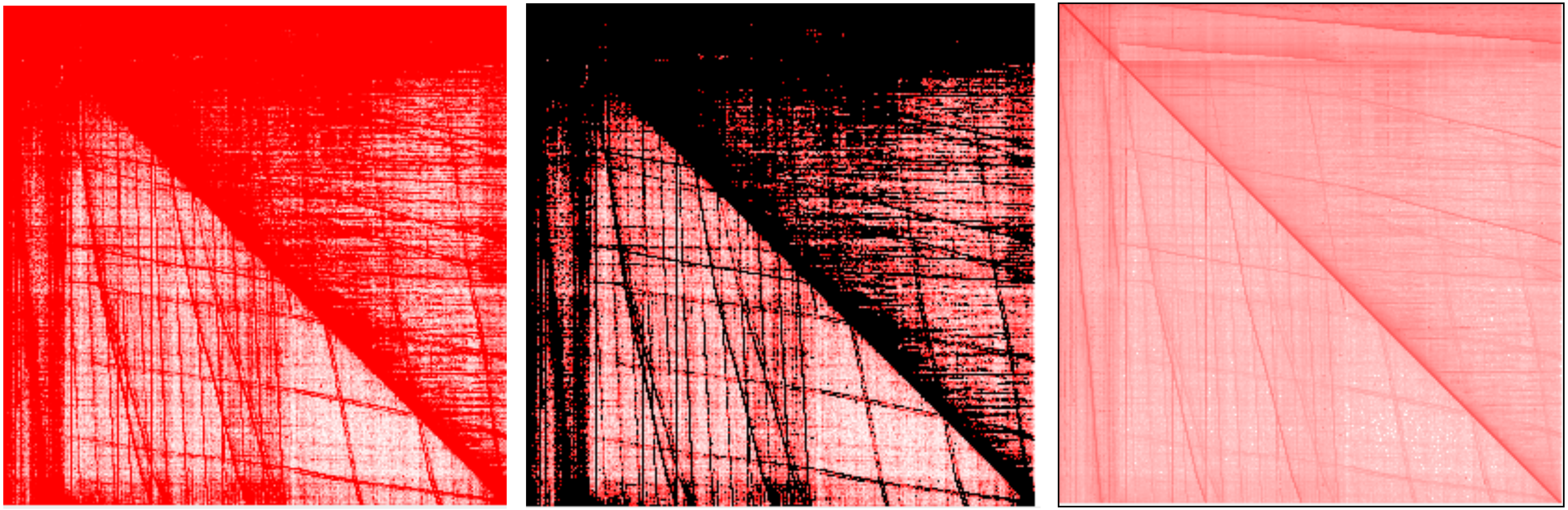


# Hi-def Alpha



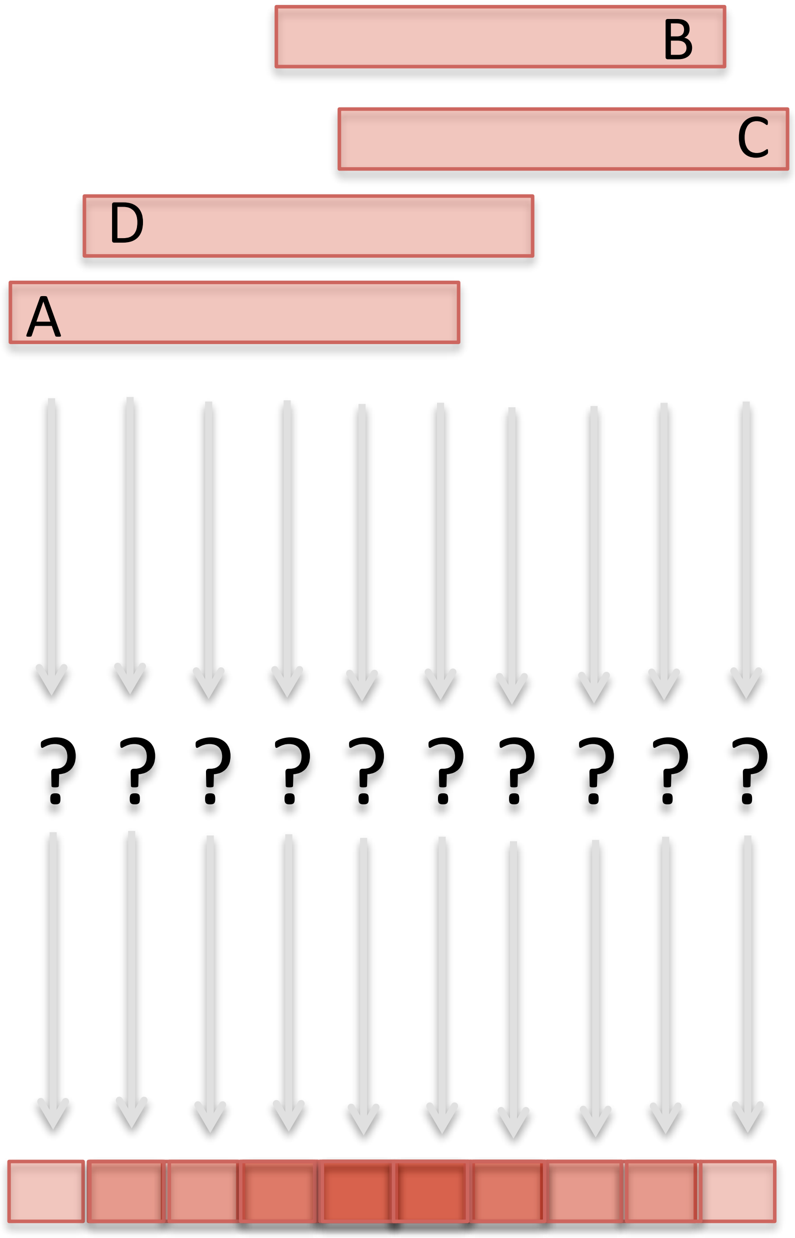
# Kiva: Abstract Rendering

Basic Abstract Rendering can identify trouble spots in standard plots, and also offer automatic tone mapping, taking perception into account.



37 mil elements, showing adjacency between entities in Kiva dataset

# Abstract Rendering




Geometry

Reduce

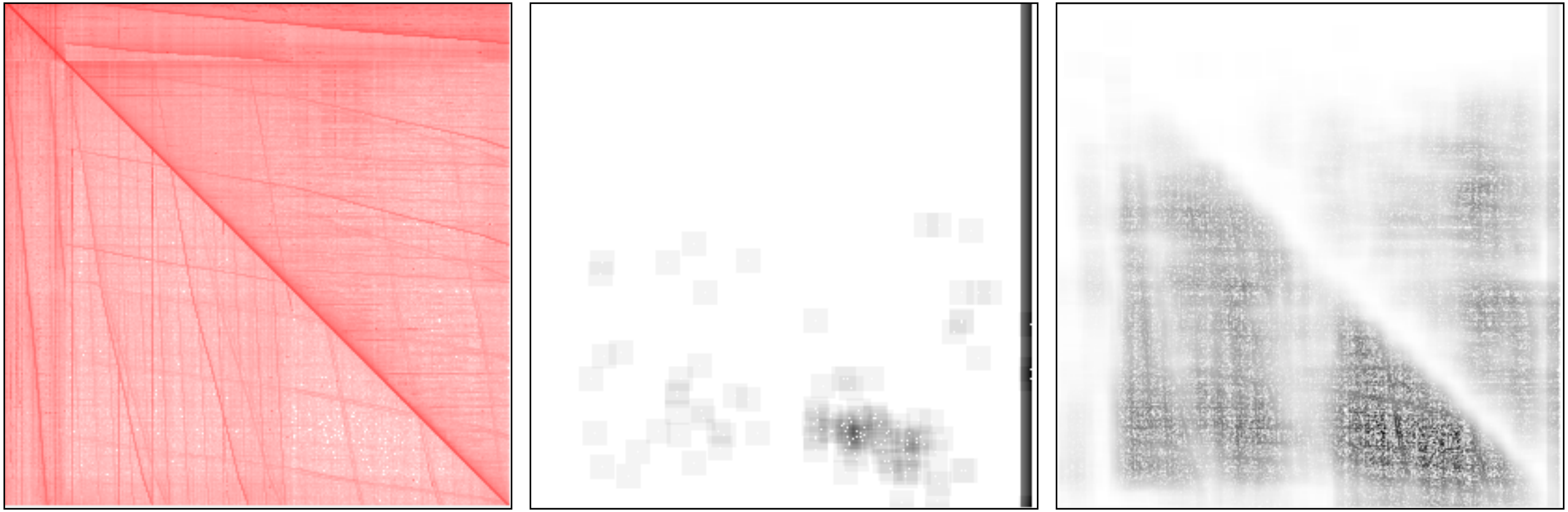
Aggregates (“Abstract” Pixels)

Transfer

Pixels

# Abstract Rendering of Sparsity

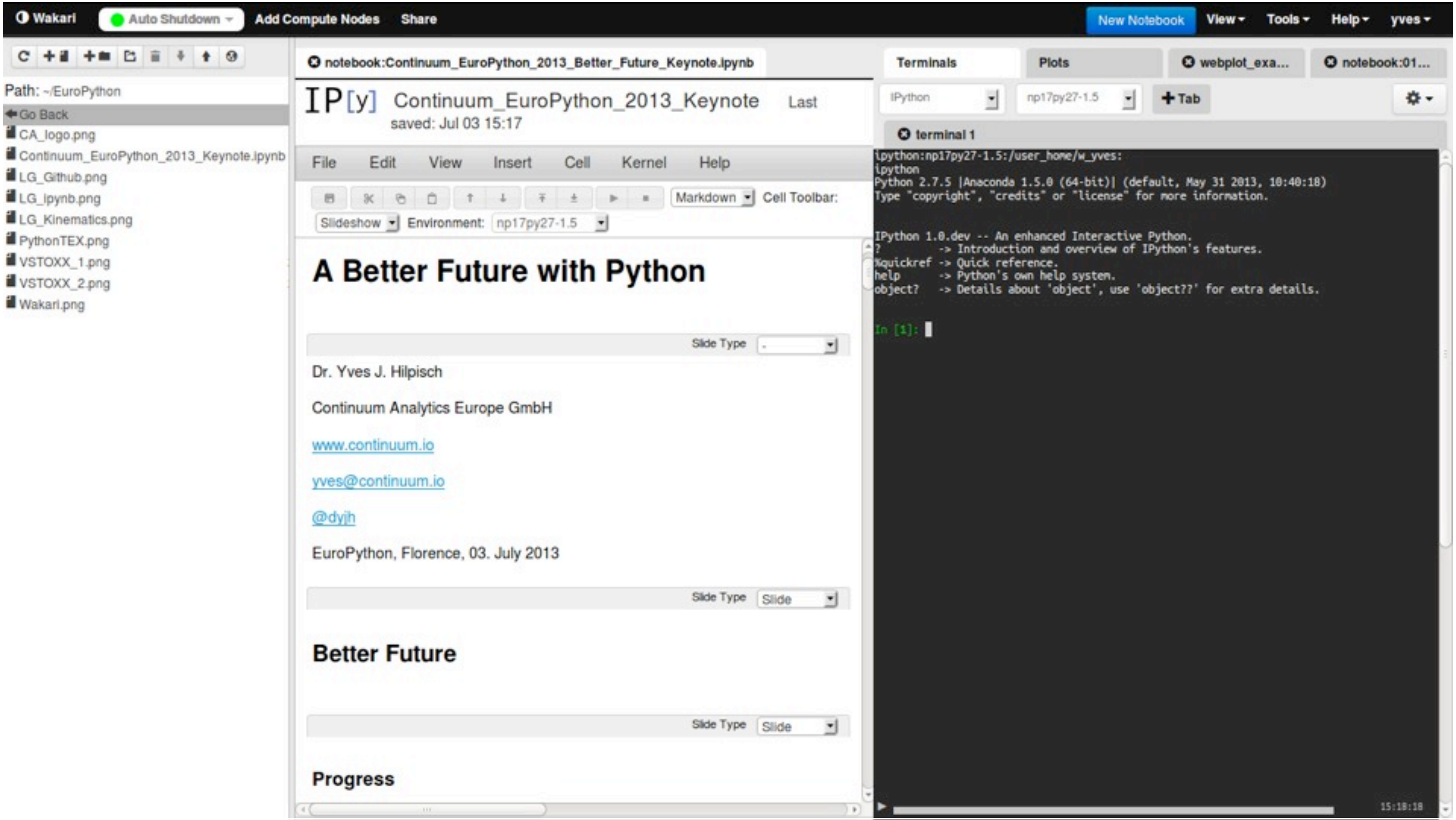
"Drawing the Dark" in Kiva Example



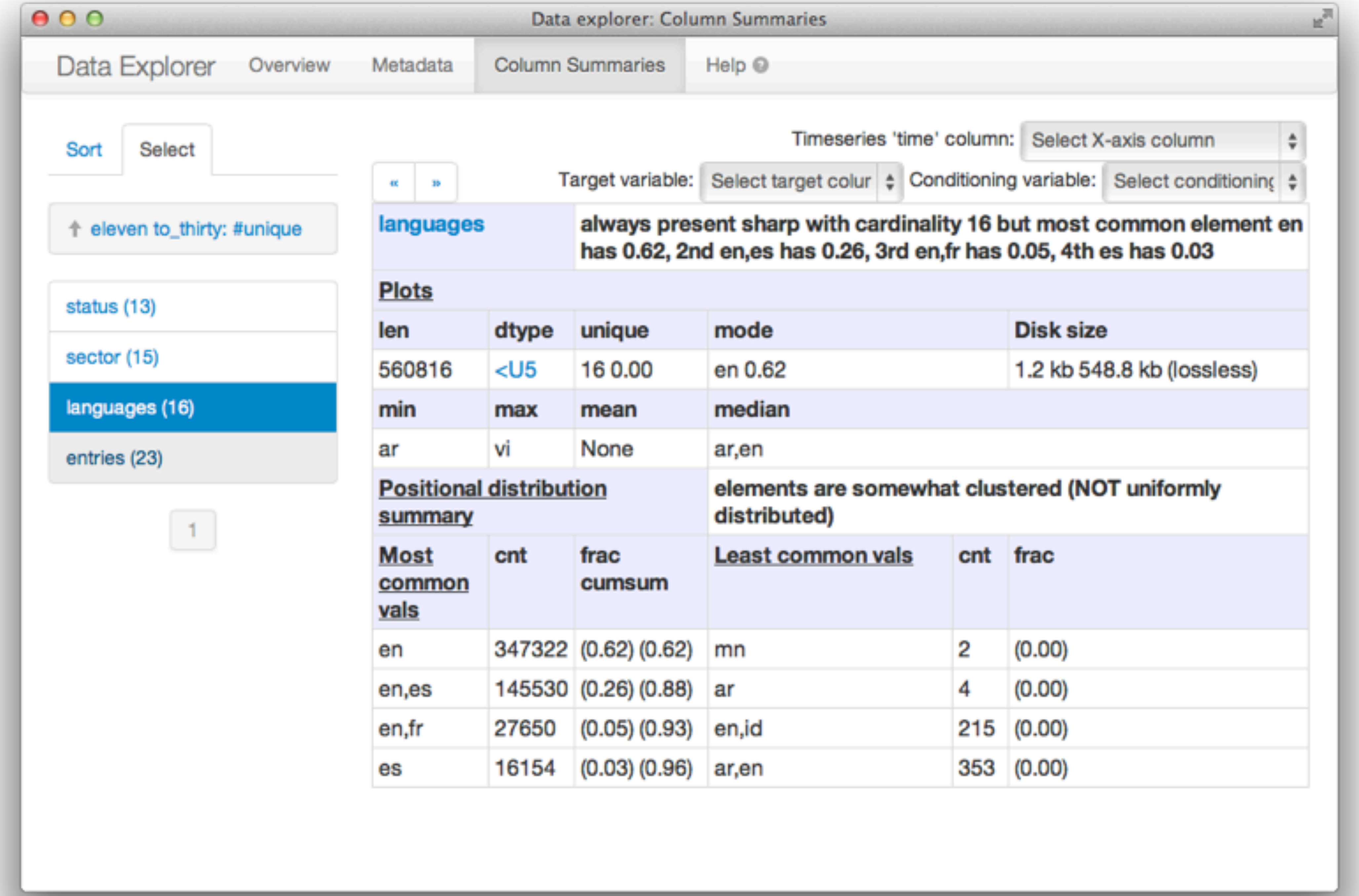
Akin to mapping the ocean trenches; typical viz starts at sea level & goes up.

# www.wakari.io

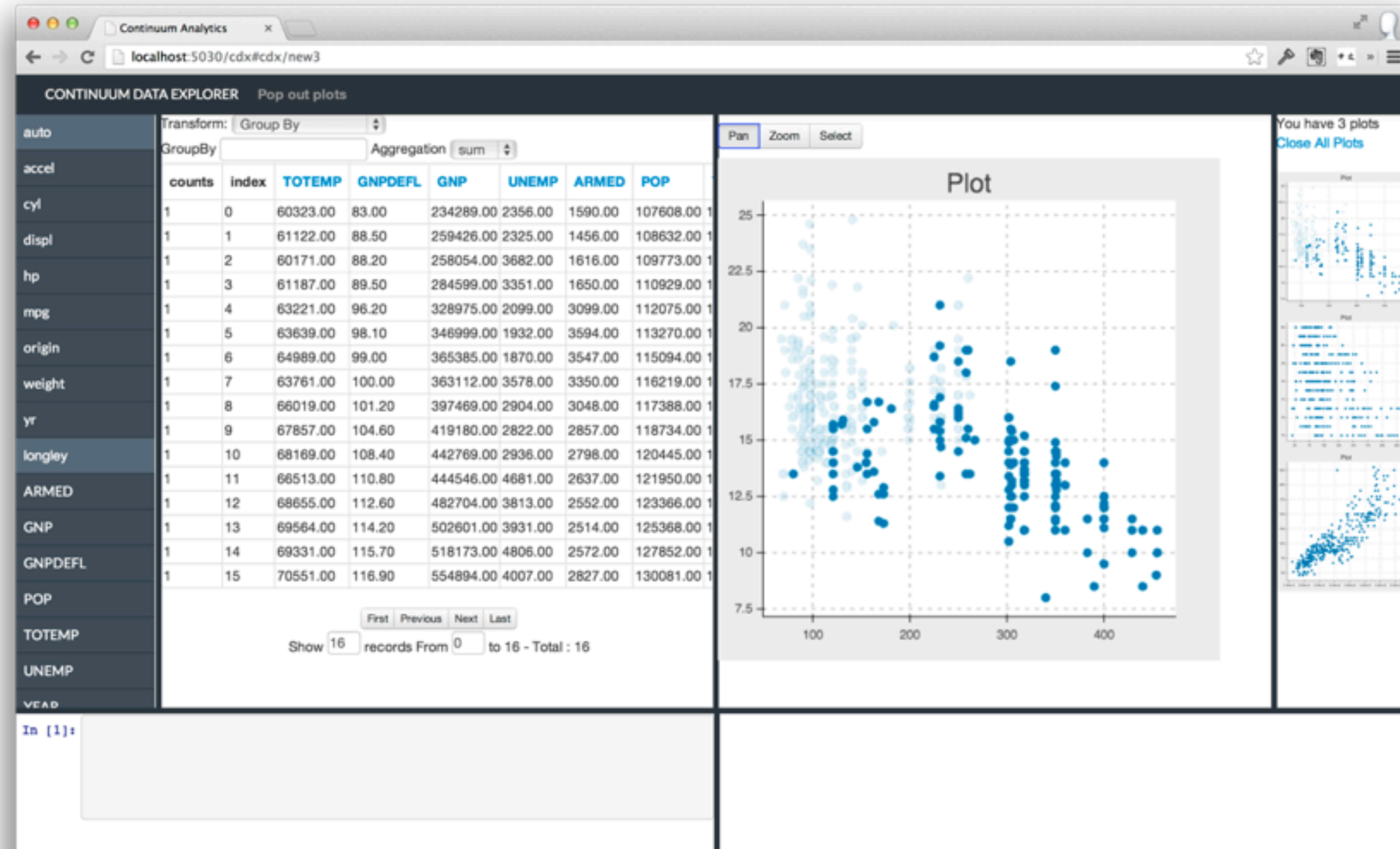
- Cloud-hosted Python analytics environment
- Full Linux sandbox for every user
- IPython notebook
- Interactive Javascript plotting
- Easy to share notebooks & code with other users
- **Free plan**: 512mb memory, 10gb disk
- Premium plans include: more powerful machines, more memory/disk, SSH access, cluster support


Wakari
Auto Shutdown
Add Compute Nodes
Share
New Notebook
View
Tools
Help
yves
Path: ~/EuroPython
Go Back
CA_logo.png
Continuum_EuroPython_2013_Keynote.ipynb
LG_Github.png
LG_ipynb.png
LG_Kinematics.png
PythonTEX.png
VSTOXX_1.png
VSTOXX_2.png
Wakari.png
notebook:Continuum_EuroPython_2013_Better_Future_Keynote.ipynb
IP[y] Continuum_EuroPython_2013_Keynote Last saved: Jul 03 15:17
File Edit View Insert Cell Kernel Help
Markdown
Cell Toolbar:
Slideshow
Environment: np17py27-1.5
A Better Future with Python
Slide Type
Dr. Yves J. Hilpisch
Continuum Analytics Europe GmbH
www.continuum.io
yves@continuum.io
@dyjh
EuroPython, Florence, 03. July 2013
Slide Type Slide
Better Future
Slide Type Slide
Progress
Terminals
Plots
webplot_exa...
notebook:01...
IPython
np17py27-1.5
Tab
terminal 1
ipython:np17py27-1.5:/user_home/w_yves:
ipython
Python 2.7.5 |Anaconda 1.5.0 (64-bit)| (default, May 31 2013, 10:40:18)
Type "copyright", "credits" or "license" for more information.
IPython 1.0.dev -- An enhanced Interactive Python.
? -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.
In [1]:
15:18:18

# Data Summary Explorer



# Continuum Data Explorer (CDX)




CONTINUUM
ANALYTICS

Continuum Analytics Europe GmbH
Rathausstrasse 75-79
66333 Voelklingen
Germany

www.continuum.io
europe@continuum.io

Dr. Yves J. Hilpisch | @dyjh