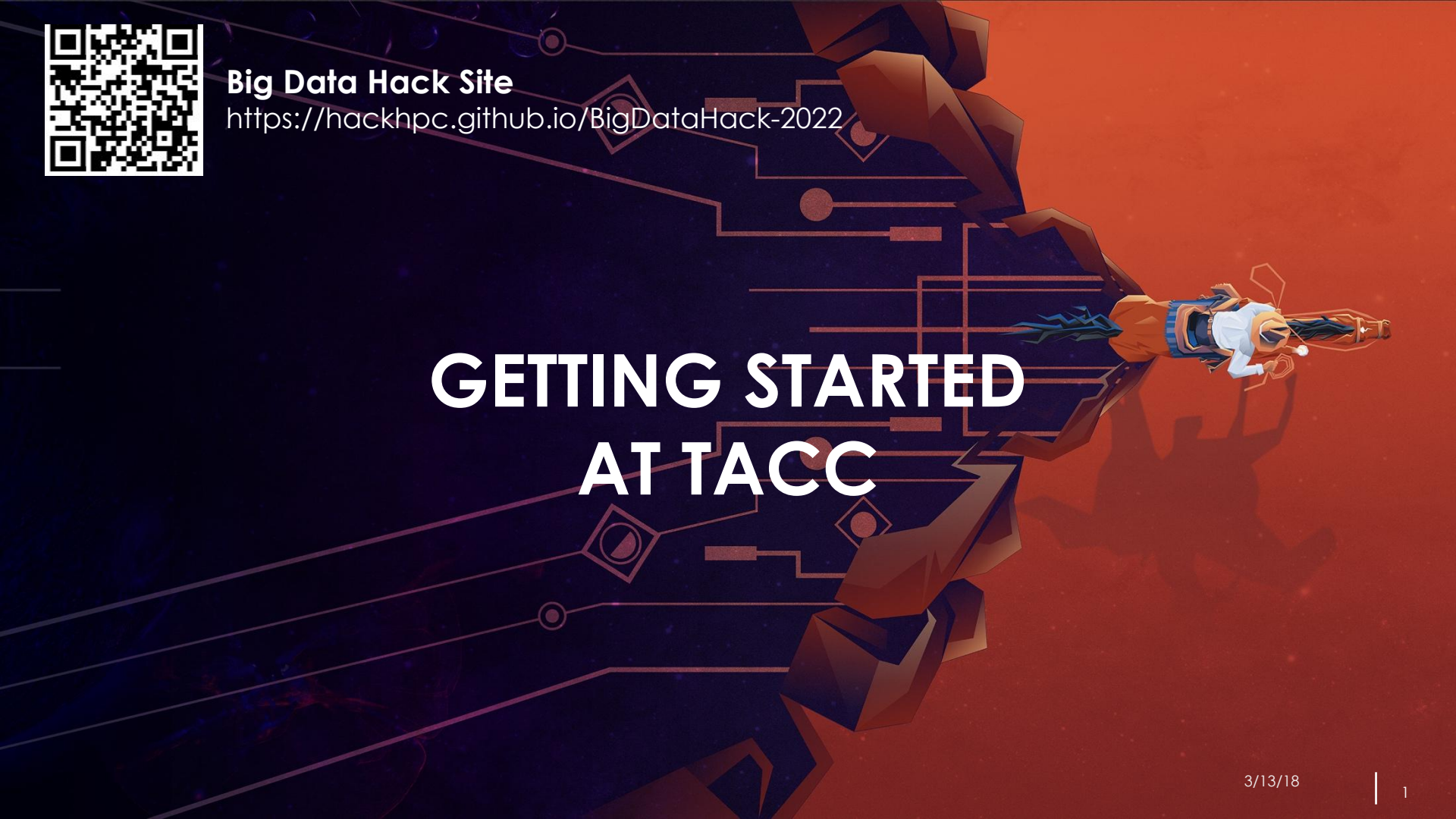
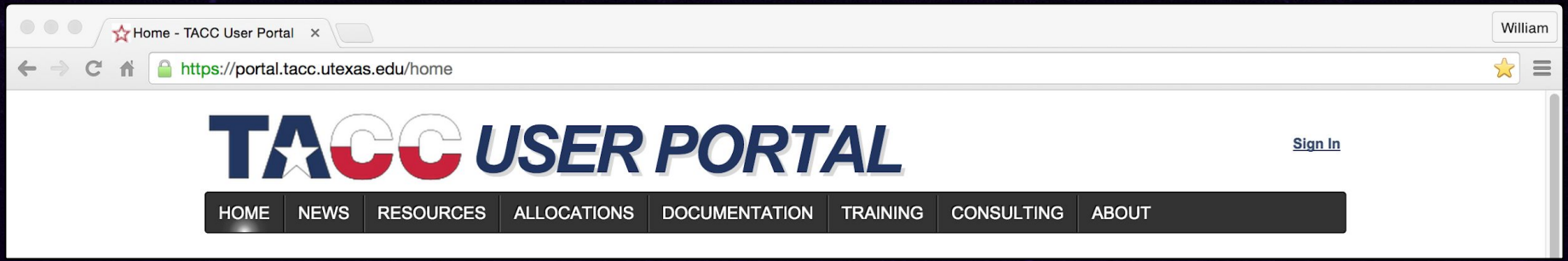
Big Data Hack Site
https://hackhpc.github.io/BigDataHack-2022



GETTING STARTED AT TACC

DesignSafe



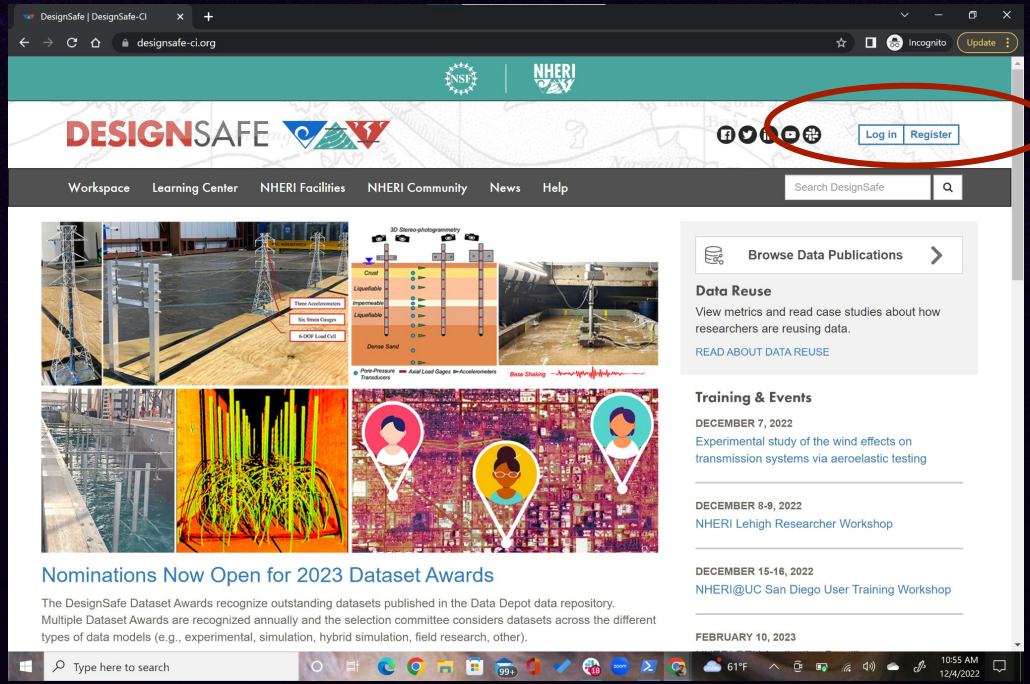
- Create accounts
- Request allocations
- Extensive system documentation
- Online training and workshops
- Software installation – availability / help
- File tickets for assistance

https://portal.tacc.utexas.edu

PROJECTS AND ALLOCATIONS

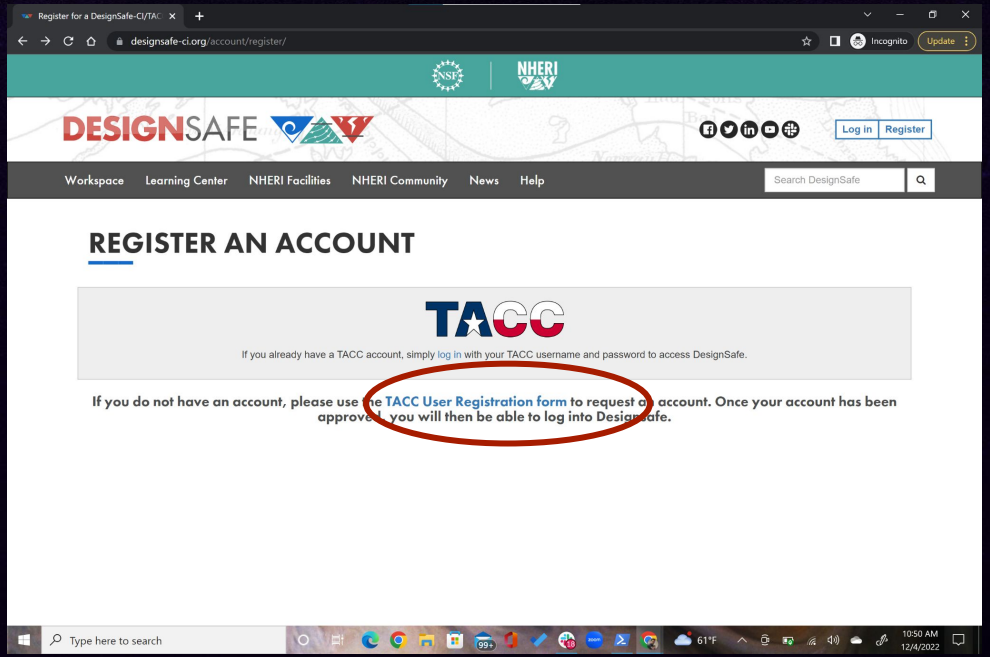
- **Project:** A group of faculty, students, and / or staff working on a common, specific research goal. Each project has one designated PI.
- **Allocation:** A specific allotment of resources (including CPU hours and / or storage space) to be shared by all users associated with a project. Each project may have multiple allocations.
- **All PIs must go through Project & Allocation system to start using TACC resources.**

SIGNING UP FOR AN ACCOUNT



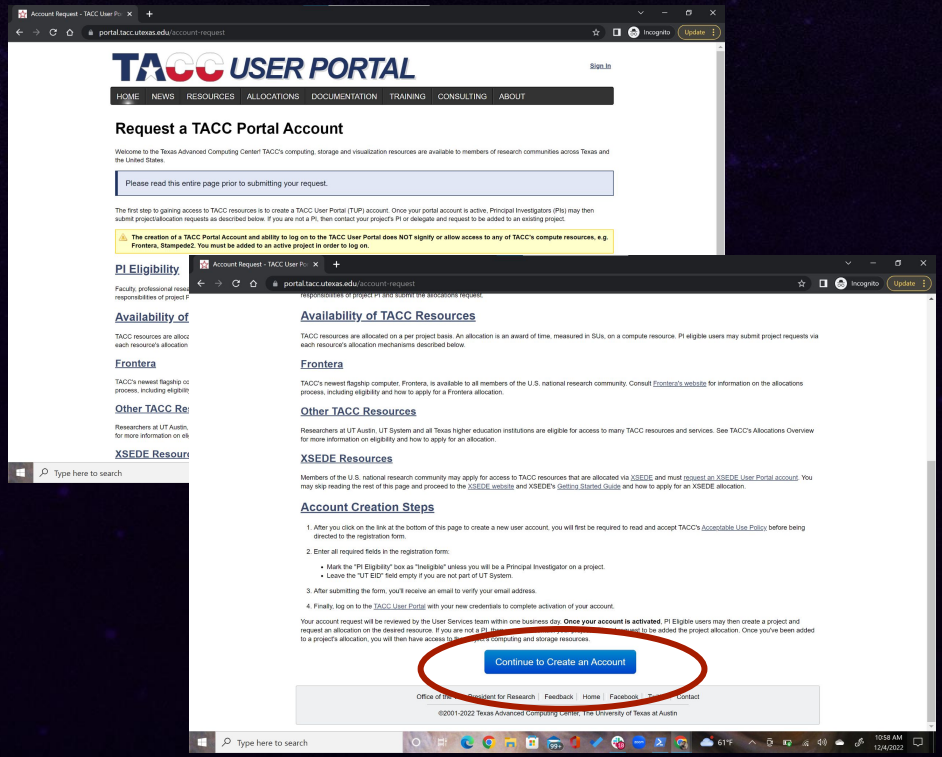
- Click "Register"

SIGNING UP FOR AN ACCOUNT



- The next page summarizes the project allocation project.
- This blue button you can click

CREATING AN ACCOUNT



- Please do not abuse our systems.
- Bro is watching you!

CREATING AN ACCOUNT

page, http://www.tacc.utexas.edu.

TACC Citation

Please reference TACC in any research report, journal or publication that requires citation of any author's work. The recognition of the TACC resources you used to perform research is important for acquiring funding for the next generation hardware, support services, and our Research & Development activities in HPC, visualization, data storage, and grid infrastructure. The minimal content of a citation should include:

> Texas Advanced Computing Center (TACC)
> The University of Texas at Austin

Our suggested acknowledgement is *:

> The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing {HPC, visualization, or storage} resources that have contributed to the research results reported within this paper. URL: http://www.tacc.utexas.edu * Select one or more of the items within the braces, {}. URL: http://www.tacc.utexas.edu

* Select one or more of the items within the braces, {}.

Document Revision History

Date	Sections Affected	Modified By	Description
December 1, 2014		Nathaniel Mendoza	Yearly Review
December 1, 2015		Nathaniel Mendoza	Yearly Review
December 1, 2016		Nathaniel Mendoza	Yearly Review
April 3, 2018	Cryptocurrencies added	Nathaniel Mendoza	Yearly Review

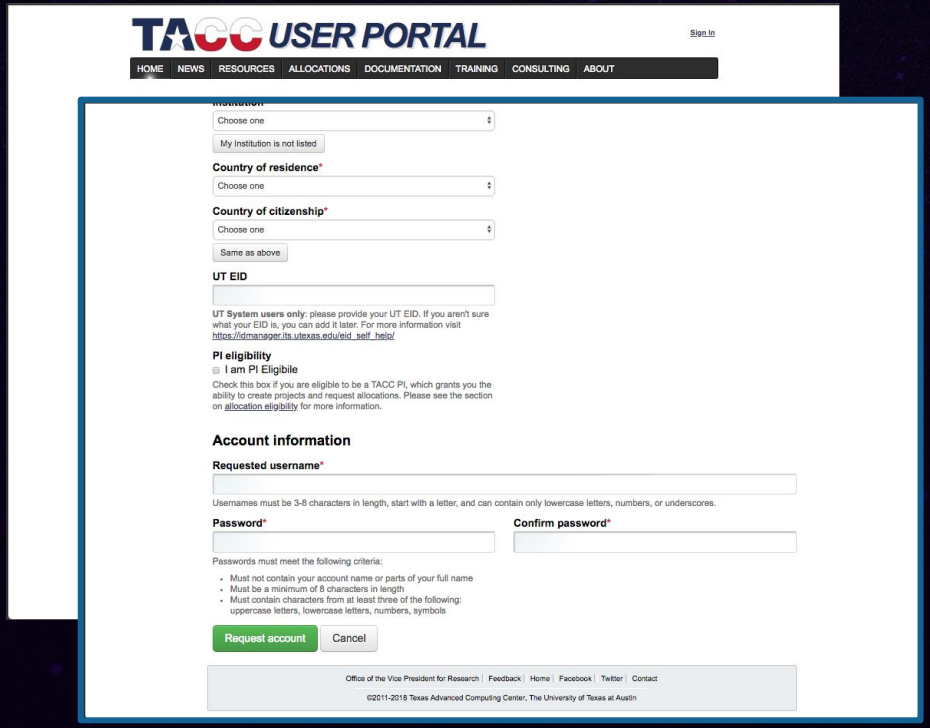
agree to the TACC Acceptable Use Policy | Cancel

Office of the Vice President for Research | Feedback | Home | Facebook | Twitter | Contact

https://portal.tacc.utexas.edu/account-request?p_p_id=createaccount_WAR_createaccountportlet&p_p_lifecycle=1&p_p_state=normal&p_p_mode=view&p_p_col_id=column-1&p_p_col_count=1&_createacco...

- Please do not abuse our systems.
- Bro is watching you!

CREATING AN ACCOUNT



- Fill out the form
- Use your Institute affiliated Email Address (i.e not Gmail, Yahoo, Hotmail, etc…)
- **Do not check the "PI Eligibility" checkbox**
- Continue to setup multi factor authentication

WELCOME TO TACC

Introductions - Icebreaker

Charlie Dey (*TACC*)

Je'aime Powell (*TACC*)

TACC AT A GLANCE

- Research center located at UT Austin
- ~175 Staff (~70 PhD scientists, ~20 students)
- Funded by UT System, NSF (85% external grants)

- **Users:** >10,000 on 2,300 active projects across all fields
- **Partnerships:** UT Research Cyberinfrastructure (UTRC), Extreme Science and Engineering Discovery Environment (XSEDE), Industry, International

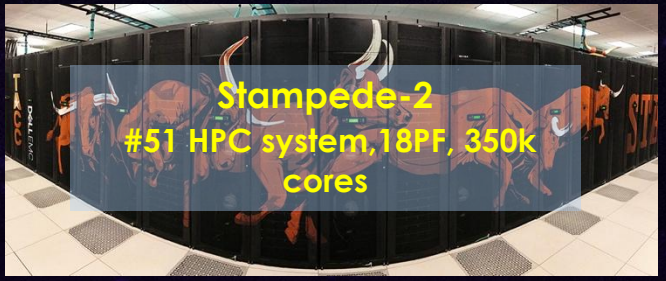
Mission: "To enable discoveries that advance science and society through the application of advanced computing technologies."

TACC AT A GLANCE

- **Capacity and Infrastructure:**
  - A billion compute hours per year
  - >5 billion files, >50 petabytes of data
  - Hundreds of public datasets
  - 18 MW data center

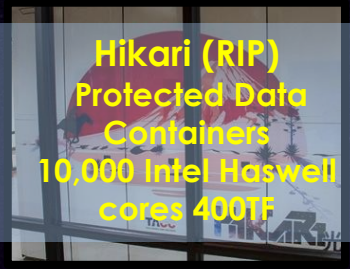
- **Systems and Services:**
  - High performance computing (HPC), high throughput computing (HTC), large scale data storage, cloud computing, visualization
  - Portals and gateways, web service APIs, rich software stacks
  - Consulting, curation and analysis, code optimization, training and outreach

AN ECOSYSTEM FOR EXTREME SCALE SUPERCOMPUTING

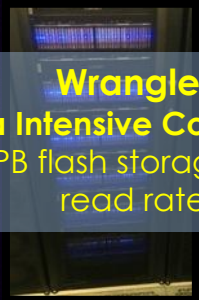


Stampede-2
#51 HPC system,18PF, 350k cores

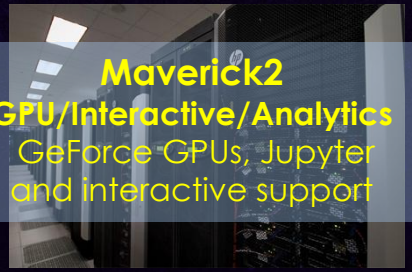
Lonestar 6
Texas-focused HPC/HTC
71k AMD Epyc 7763 cores,
5TF, 20+ GPUs



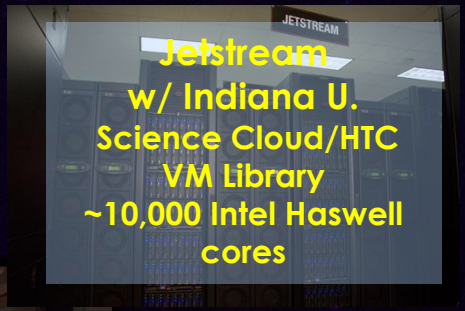
Hikari (RIP)
Protected Data Containers
10,000 Intel Haswell cores 400TF



Wrangler
Data Intensive Computing
0.6 PB flash storage 1 TB/s read rate

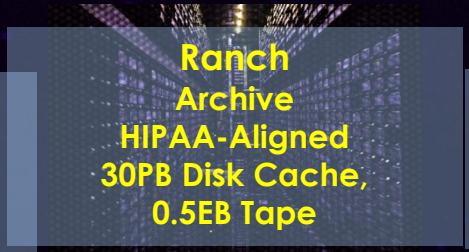


Maverick2
GPU/Interactive/Analytics
GeForce GPUs, Jupyter and interactive support



Jetstream
w/ Indiana U.
Science Cloud/HTC
VM Library
~10,000 Intel Haswell cores

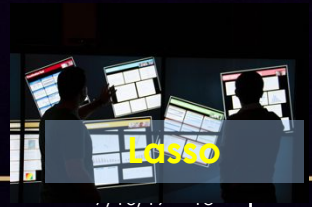
Stockyard
Shared Storage Across TACC
40PB, Lustre



Ranch
Archive
HIPAA-Aligned
30PB Disk Cache,
0.5EB Tape

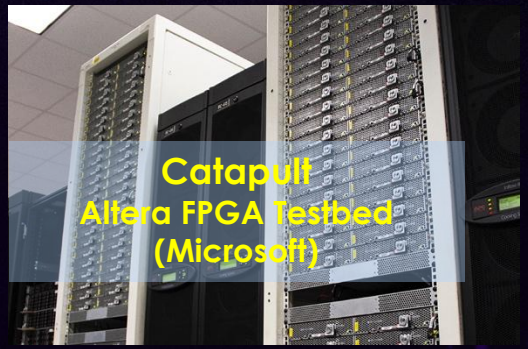
Corral
Published Data Collections
HIPAA-Aligned
20PB Replicated Disk,

Stallion

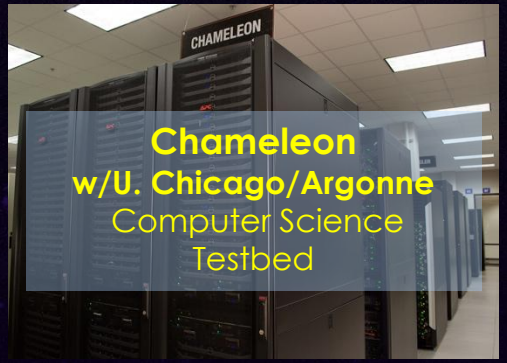


Lasso

EXPERIMENTAL SYSTEMS



Catapult
Altera FPGA Testbed
(Microsoft)



Chameleon
w/U. Chicago/Argonne
Computer Science
Testbed

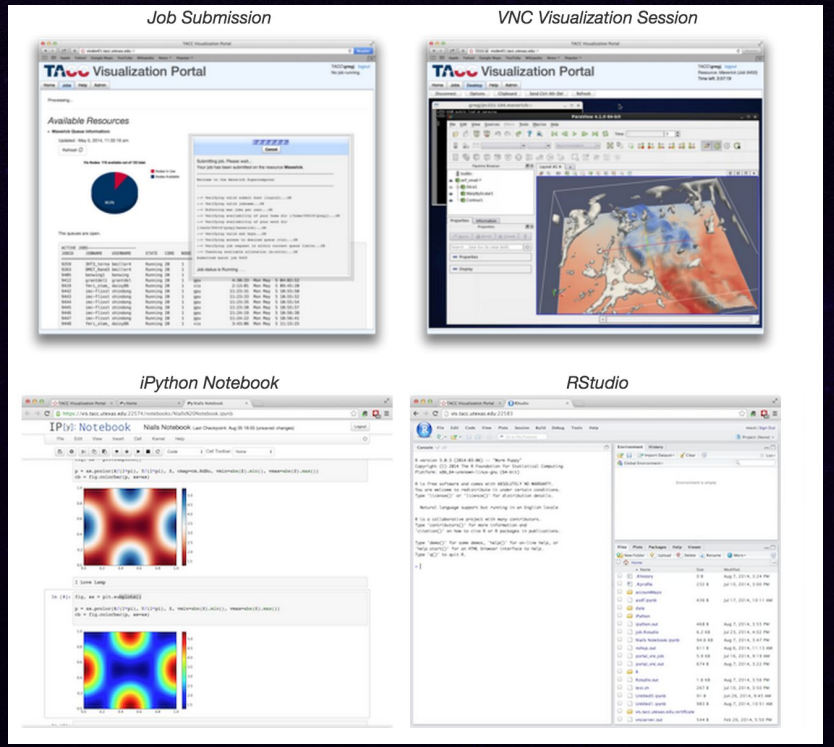
Fabric
Alternate Architectures
(IBM,CAPI,FPGA, GPU)

Discovery
New Processor/Storage
Benchmarking

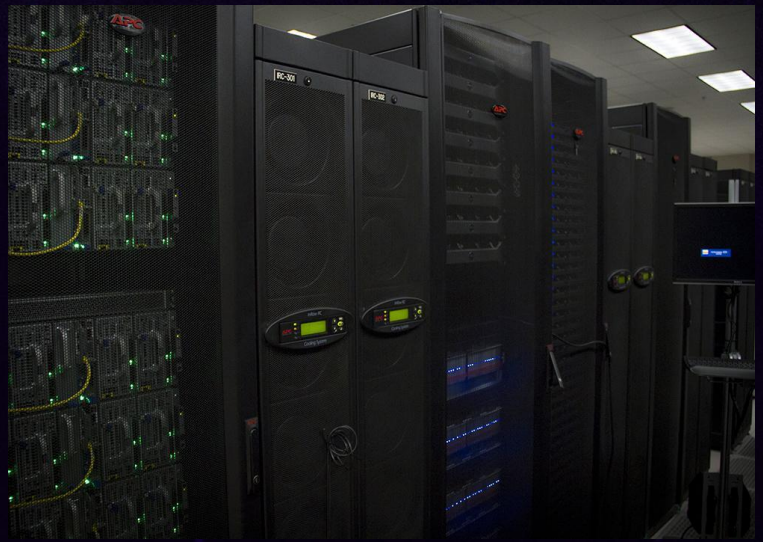
Rustler
Object Storage Testbed

VISUALIZATION, MACHINE LEARNING

- **Maverick 2**
  - 23 nodes, 4 NVIDIA GTX 1080ti per node
  - 4 nodes, 2 NVIDIA V100 per node
  - 3 nodes, 2 NVIDIA P100 per node
  - Machine learning / deep learning



DATA INTENSIVE COMPUTING



- **Wrangler (retired)**
  - >3,000 processor cores for analytics
  - 10 PB storage system; 600 TB DSSD flash storage
  - Aggregate bandwidth >1 TB/s
  - **Only allocable through XSEDE at this time**

DATA AND COLLECTIONS REPOSITORY

- **Corral**
  - 11 PB geo-replicated storage (Austin / Arlington)
  - 5 TB free to all UT System principal investigators
  - $118 per TB per year after the first 5 TB
  - Project data sharing / collection hosting

HIGH PERFORMANCE COMPUTE CLUSTERS

- **Lonestar6**

Lonestar6 hosts 560 compute nodes with 5 TFlops of peak performance per node and 256 GB of DRAM. Node Specifications:

- CPU: 2x AMD EPYC 7763 64-Core Processor ("Milan")
  - 128 cores on two sockets (64 cores / socket)
  - Clock rate: 2.45 GHz (Boost up to 3.5 GHz)
  - RAM: 256 GB (3200 MT/s) DDR4
- GPU: 2x AMD EPYC processes and 2 NVIDIA A100 GPUs
  - each with 40 GB of high bandwidth memory (HBM2).
- Local storage: 144GB /tmp partition on a 288GB SSD.
- **UT System only**

FLAGSHIP HIGH PERFORMANCE COMPUTING SYSTEM: STAMPEDE2

HIGH PERFORMANCE COMPUTE CLUSTERS

- **Stampede2**
  - 18 petaflops of peak performance
  - 4,200 Intel Knights Landing node
    - 68 cores, 96GB of DDR RAM, and 16GB of high speed MCDRAM
  - 1,736 Intel Xeon Skylake nodes
    - 48 cores, 192GB of RAM
  - 240 Intel "Ice Lake" (ICX) nodes
    - 80 cores on 2 sockets (40 cores/socket).
    - 160 hardware threads per node w/ hyperthreading
  - 100 Gb/sec Intel Omni-Path network with a fat tree topology employing six core switches

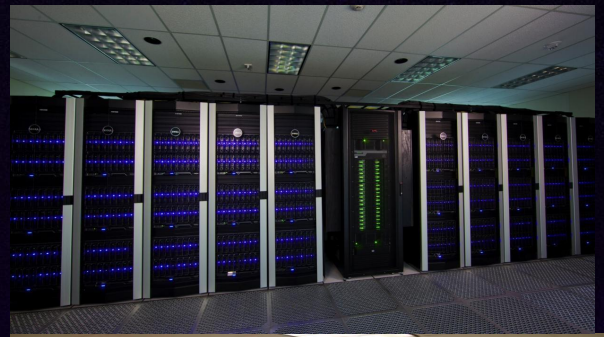
"COMPUTING FOR THE ENDLESS FRONTIER"

Dan Stanzione
Executive Director, Texas Advanced Computing Center
Associate Vice President for Research, The University of Texas at Austin


FRONTERA
TACC
NSF
TEXAS

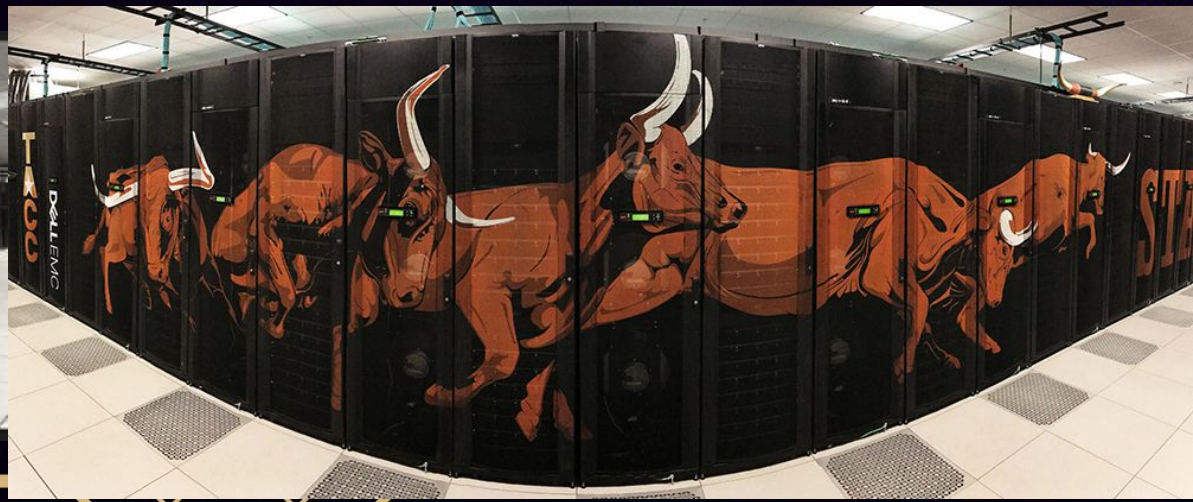
RAPID GROWTH FROM THEN TO NOW…

- 2003 – First Terascale Linux cluster for open science (#26)
- 2004 – NSF funding to join the Teragrid
- 2006 – UT System Partnership to provide Lonestar-3 (#12)
- **2007 - $59M NSF award – largest in UT history – to deploy Ranger, the world's largest open system (#4)**
- 2008 – funding for new Vis software and launch of revamped visualization lab.
- 2009 - $50M iPlant Collaborative award (largest NSF bioinformatics award) moves a major component to TACC, life sciences group launched.
  - In 2009, we reached, 65 employees.



NOW, A WORLD LEADER IN CYBERINFRASTRUCTURE

- 2010, TACC becomes a core partner (1 of 4) in XSEDE, the TeraGrid Replacement
- 2012, Stampede replaces Ranger with new $51.5M NSF Award
- 2013, iPlant is renewed, expanded to $100M
- 2015, Wrangler, first data intensive supercomputer is deployed.
- 2015, Chameleon cloud is launched
- 2015, DesignSafe, the cyberinfrastructure for natural hazard engineering, is launched.
- 2016 Stampede-2 awarded the largest academic system in the United States, 2017-2021.



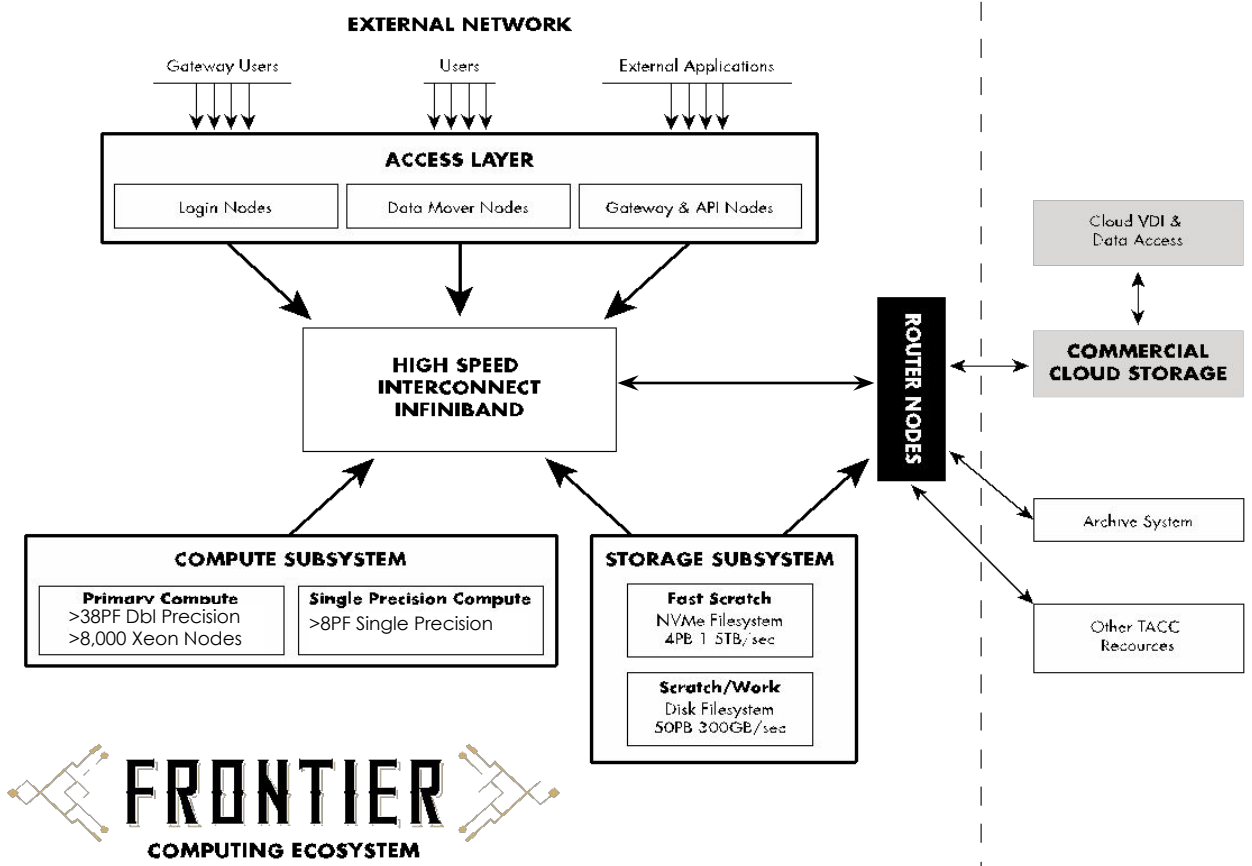
FRONTERA SYSTEM --- PROJECT

- Deploy a system in 2019 for the largest problems scientists and engineers currently face.
- Support and operate this system for 5 years.
- Plan a potential phase 2 system, with 10x the capabilities, for the future challenges scientists will face.

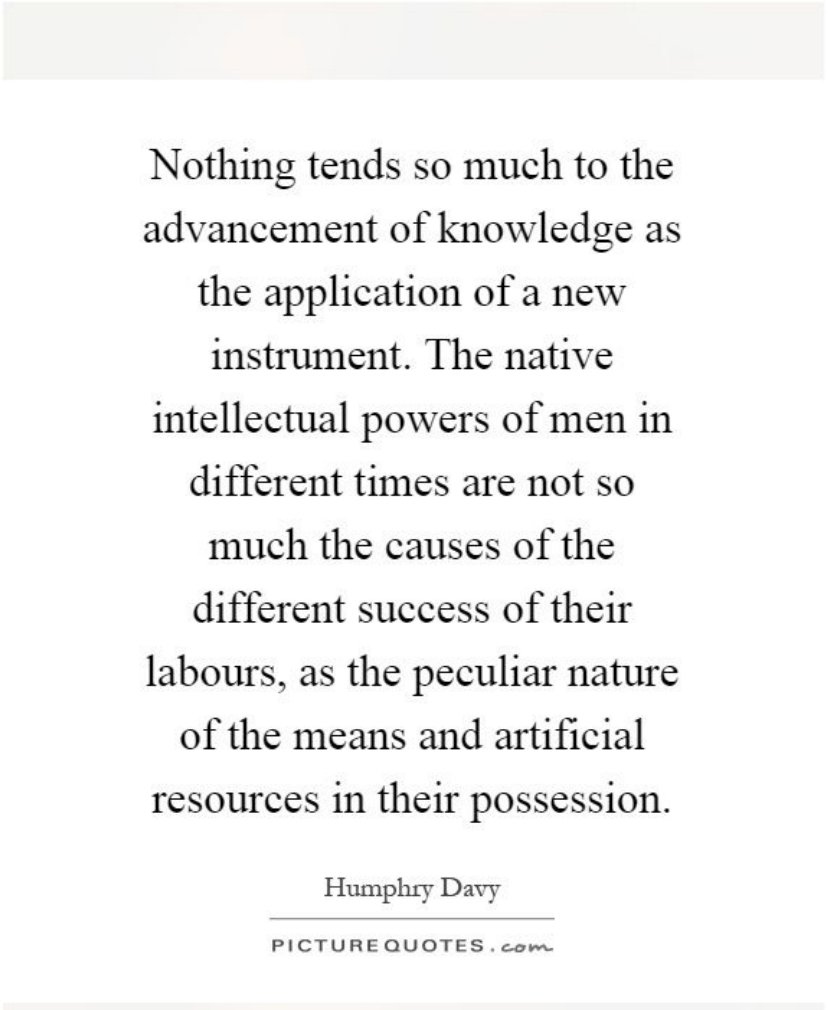
FRONTERA SYSTEM --- HARDWARE

- 8th Fastest Supercomputer in the world
  - #1 for Open Science
- Primary compute system: DellEMC and Intel
  - 35-40 PetaFlops Peak Performance (Next Generation Xeon processors)
- Interconnect: Mellanox HDR and HDR-100 links.
  - Fat Tree topology, 200Gb/s links between switches.
- Storage: DataDirect Networks
  - 50+ PB disk, 3PB of Flash, 1.5TB/sec peak I/O rate.
- Single Precision Compute Subsystem: Nvidia
- Front end for data movers, workflow, API

SYSTEM OVERVIEW



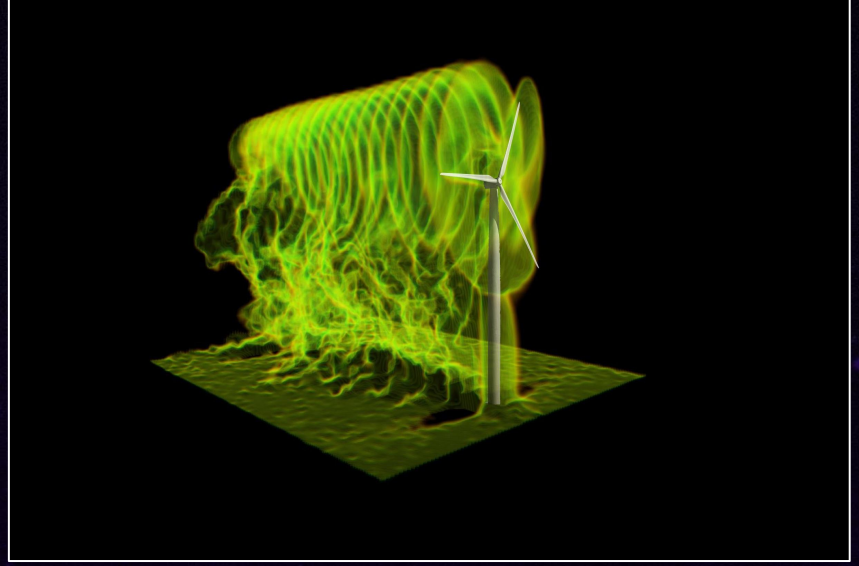
- **Humphry Davy, Inventor of Electrochemistry, 1812**
- (Pretty sure he was talking about our machine).



REAPING POWER FROM WIND FARMS

Multi-Scale Model of Wind Turbines

- Optimized control algorithm improves design choices
- New high-res models add nacelle and tower effects
- Blind comparisons to wind tunnel data demonstrate dramatic improvements in accuracy
- Potential to increase power by 6-7% ($600m/yr nationwide)



"TACC...give[s] us a competitive advantage…"

Graphic from Wind Energy, 2017.

Christian Santoni, Kenneth Carrasquillo, Isnardo Arenas-Navarro, and Stefano Leonardi

UT Dallas, US/European collaboration (UTRC, NSF-PIRE 1243482)

TACC Press Release

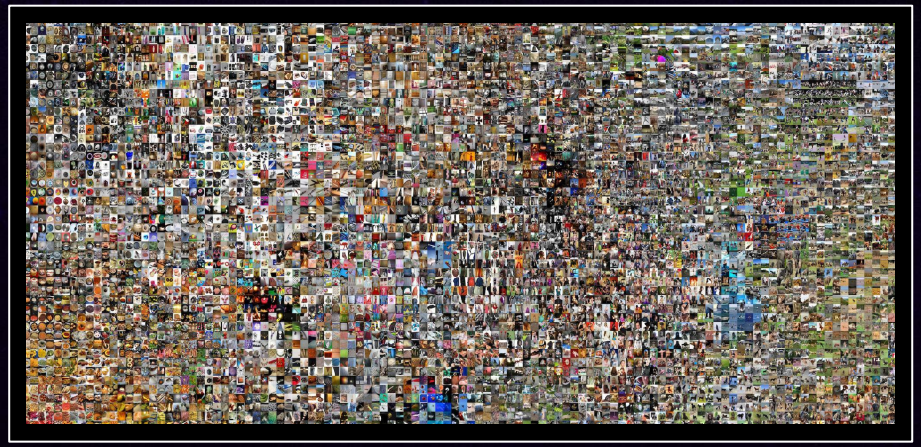
"Using commodity HPC servers...the time to data-driven discovery is reduced and overall efficiency can be significantly increased." (Niall Gaffney, TACC)

RECORD ACHIEVED ON AI BENCHMARK

TACC, Berkeley, Cal Davis collaborate on large-scale AI runs

- Research demonstrating the potential of commodity hardware for AI
- Skylake ImageNet benchmark: (100 epochs, 11 min, 1024 nodes) -- fastest result at time of publication
- Knights Landing ImageNet benchmark (90 epochs, 20 min, 2048 nodes) – 3x faster than Facebook, with higher large-batch accuracy

Graphic credit Andrej Karpathy



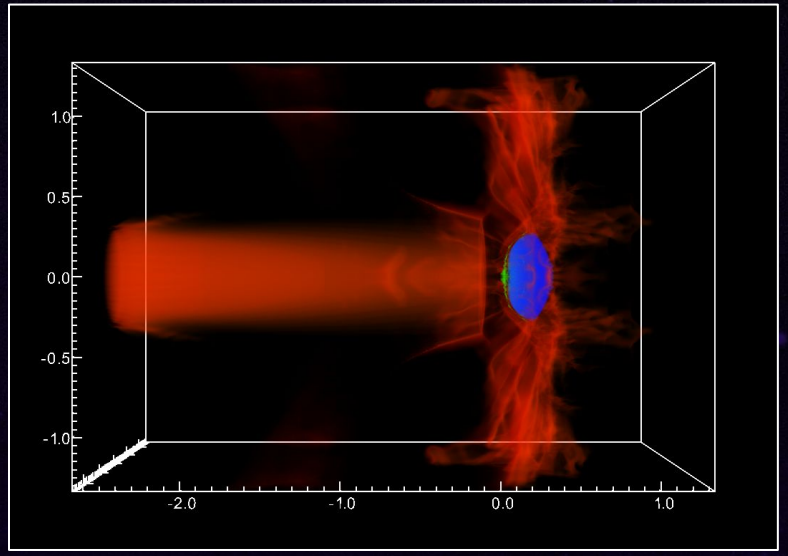
Yang You, Zhao Zhang, Cho-Jui Hsieh, James Demmel, Kurt Keutzer

TACC Press Release

USING KNL TO PROBE SPACE ODDITIES

Ongoing XSEDE collaboration focusing on KNL performance for new, high-resolution version of COSMOS MHD code

- Vectorization and other serial optimizations improved KNL performance by 50%
- COSMOS currently running 60% faster on KNL than Stampede1
- Work on OpenMP-MPI hybrid optimizations now underway
- Impact of performance improvements amounts to millions of core-hours saved



"The science that I do wouldn't be possible without resources like [Stampede2]...resources that certainly a small institution like mine could never support. The fact that we have these national-level resources enables a huge amount of science that just wouldn't get done otherwise." (Chris Fragile)

XSEDE ECSS: Collaboration between PI Chris Fragile (College of Charleston) and Damon McDougall (TACC)

TACC Press Release

MASSIVE DATA SET WORTHY OF ROSS ICE SHELF ITSELF

TACC partners with Lamont-Doherty Earth Observatory (LDEO) to host for one of the country's largest earth sciences data collections

- Managing hundreds of TB using Stampede2, Corral, and Ranch: storage, provenance, visualization, and public access
- Achieved 10x workflow speedup by moving to TACC (from 50 hrs down to 5 hrs for transfer and analysis tasks)

"...partnership...with TACC shows [it's] possible to manage…this level of data in a cost-effective, user-friendly and easily accessible manner…"

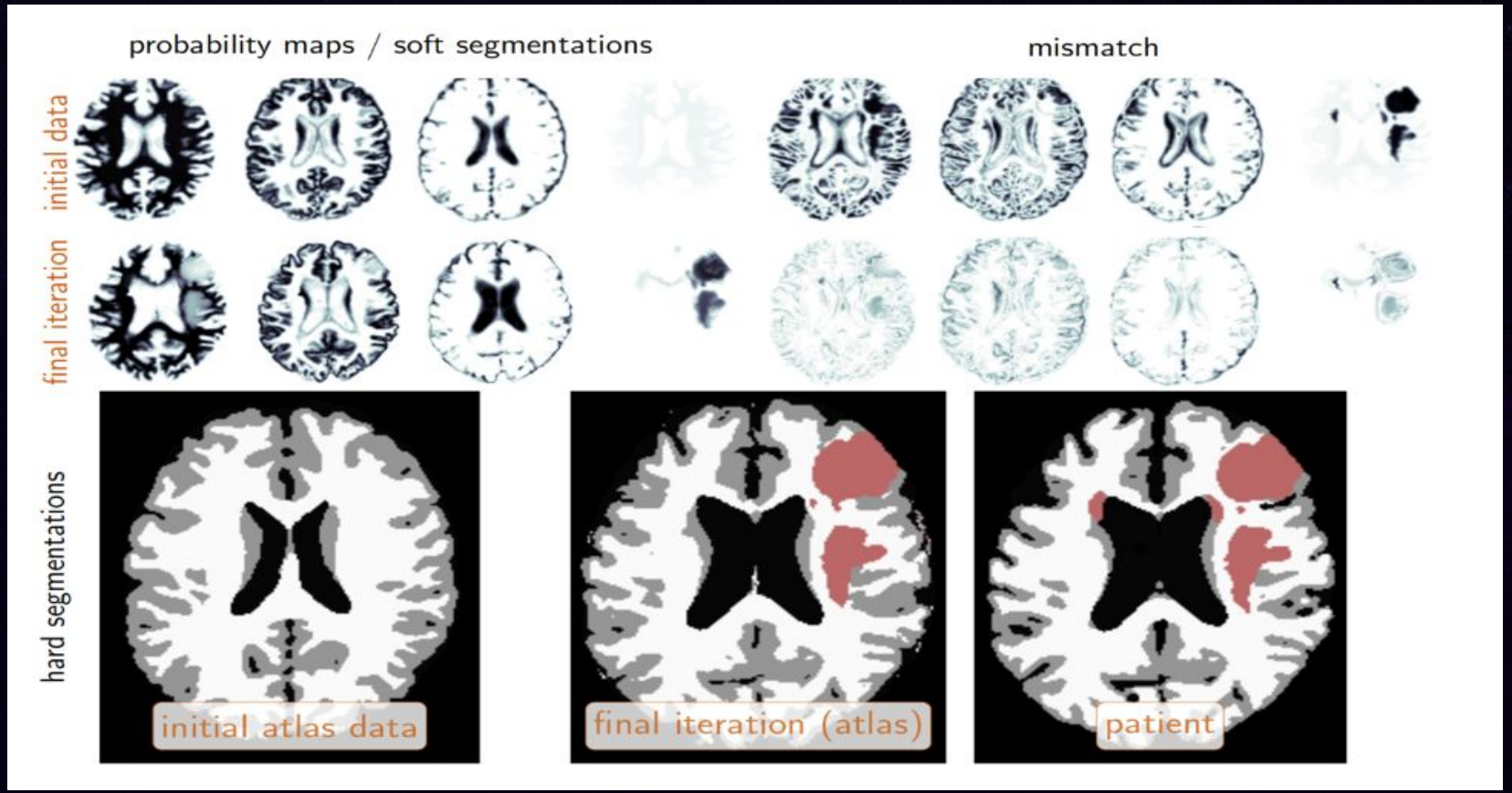
Image courtesy Oceanwide Expeditions.

TACC Press Release

PI Lingling Dong, Columbia University
XSEDE support to multidisciplinary, multi-institutional Rosetta project

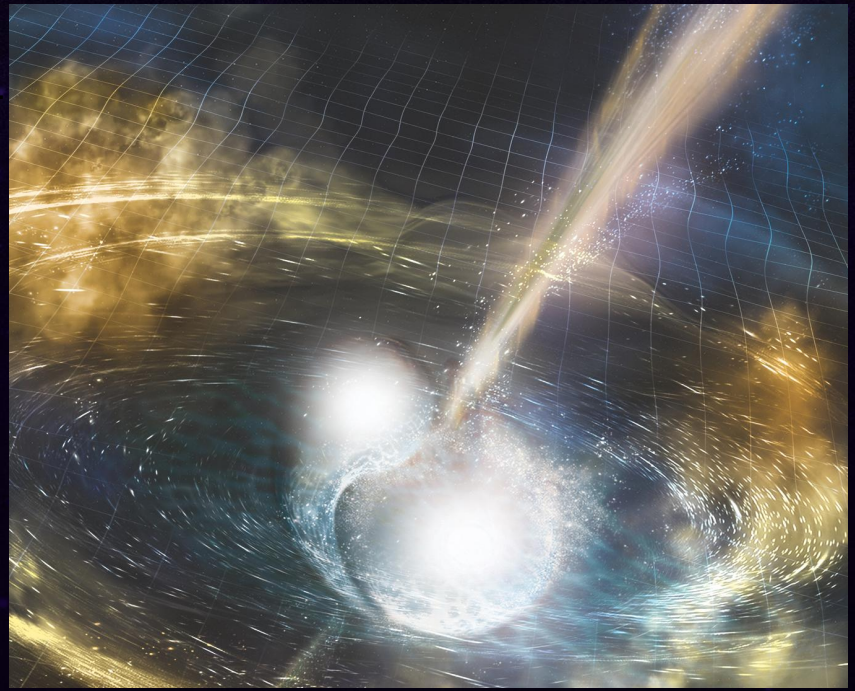
BRAIN TUMOR SEGMENTATION

- A team of researchers led by George Biros from The University of Texas at Austin scored in the top 25% of participants in the Multimodal Brain Tumor Segmentation Challenge 2017 (BRaTS'17) enabled by Stampede2 and other TACC resources.
- In the challenge, research groups presented methods and results of computer-aided identification and classification of brain tumors, as well as different types of cancerous regions.
- The team's method combined biophysical models of tumor growth with machine learning algorithms for the analysis of Magnetic Resonance imaging data of glioma patients.


probability maps / soft segmentations
mismatch
initial data
final iteration
hard segmentations
initial atlas data
final iteration (atlas)
patient

The Laser Interferometer Gravitational-Wave Observatory (LIGO)

waves

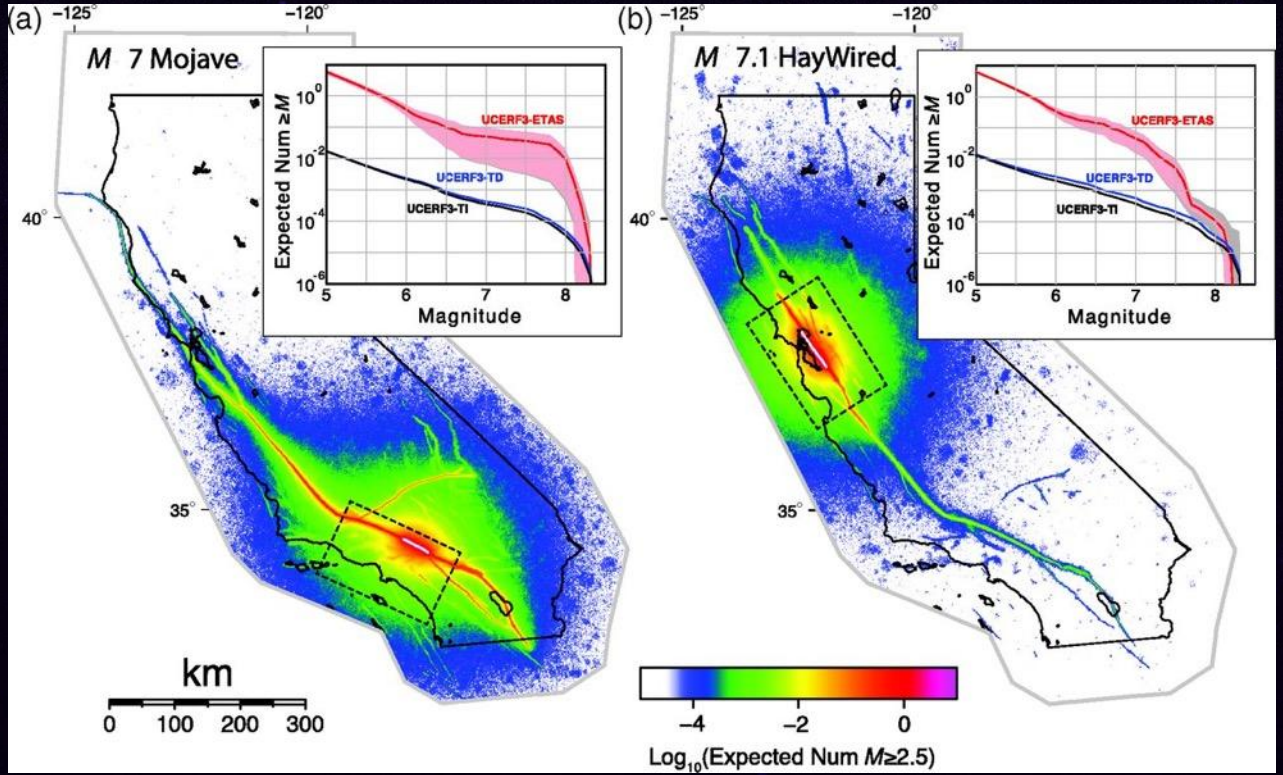


- TACC provided consulting services to LIGO researchers, 10x improvement in runtime
- seven million core hours on Stampede to help analyze the first detected gravitational

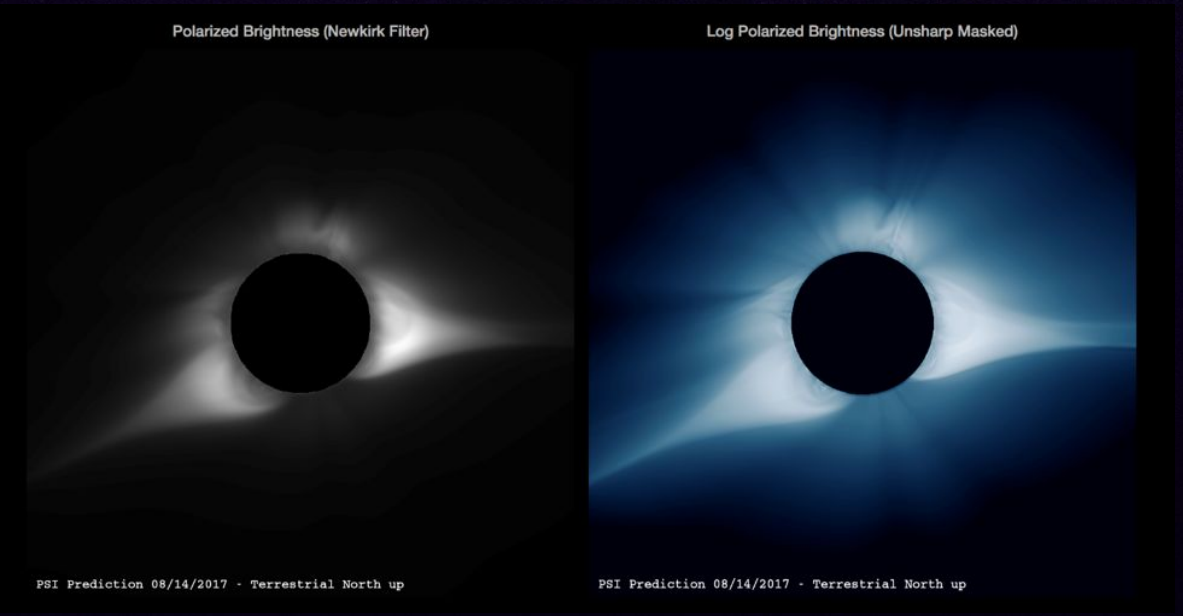
"Stampede was an excellent tool for improving our understanding of the universe we live in, from the smallest scale of sub-atomic particles to detecting gravitational waves that have traveled a million light-years to the earth, and a lot of exciting science and engineering in between"

Stuart Anderson, Caltech
LIGO Research Manager

AFTERSHOCK FORECASTING

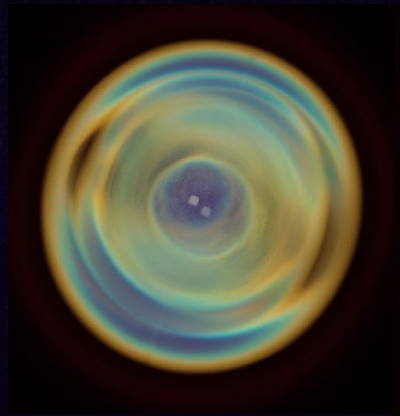


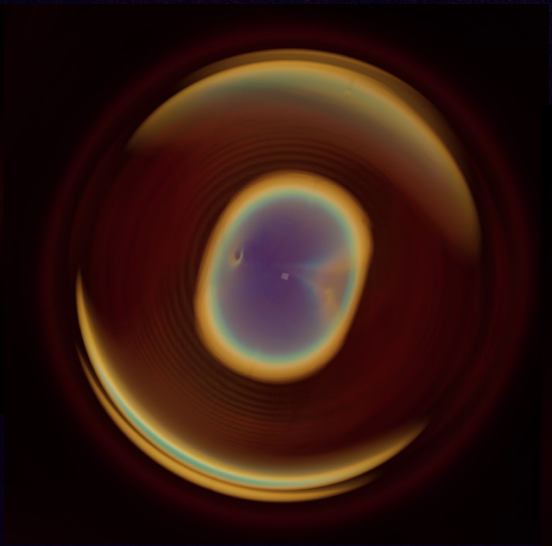
SOLAR CORONA PREDICTION



- Predictive Science, Inc. (California)
- Supporting NASA Solar Dynamics Observatory (SDO)
- Predicted solar corona on S2 during 8/21/17 eclipse

COSMOS GRAVITATIONAL WAVES STUDY





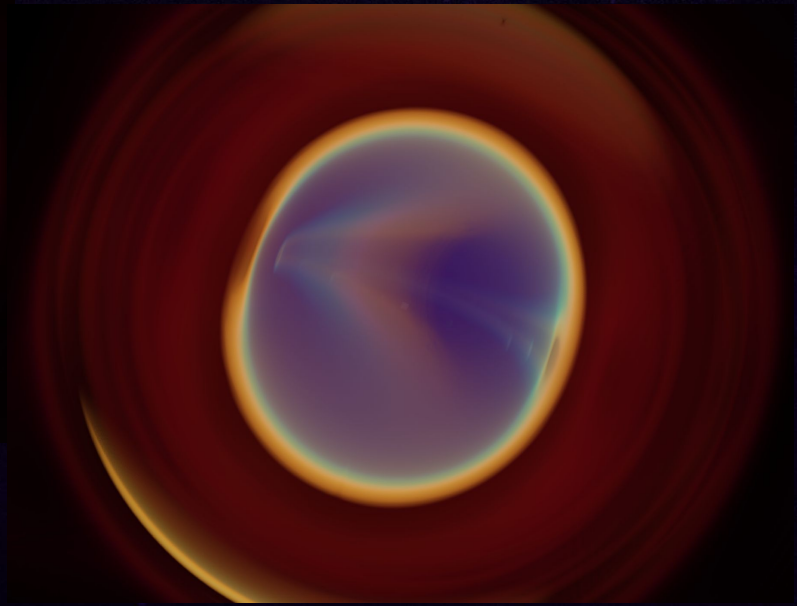


Image Credits:

Greg Abram – TACC

Francesca Samsel – CAT

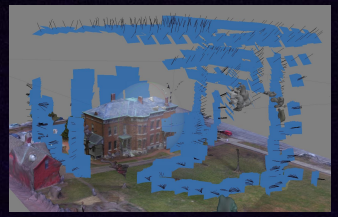
Carson Brownlee - Intel

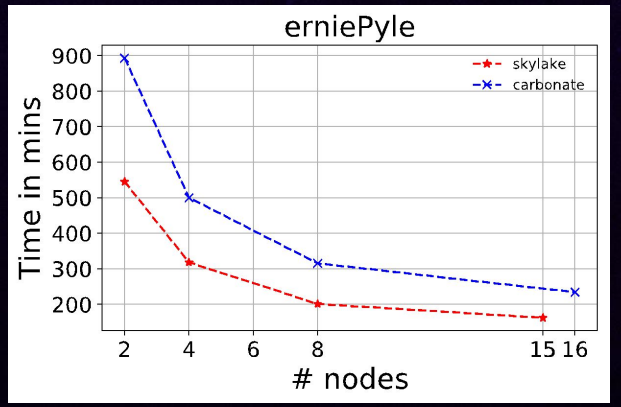
Markus Kunesch, Juha Jäykkä, Pau Figueras, Paul Shellard

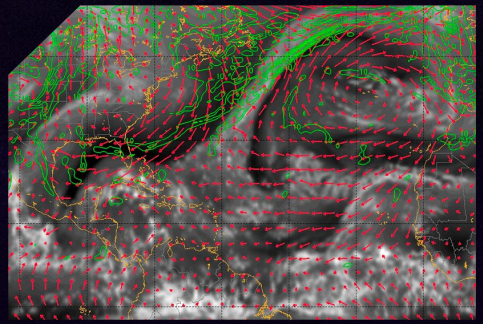
Center for Theoretical Cosmology, University of Cambridge

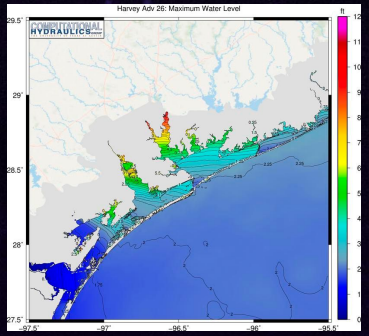
PHOTOGRAMETRY ON KNL

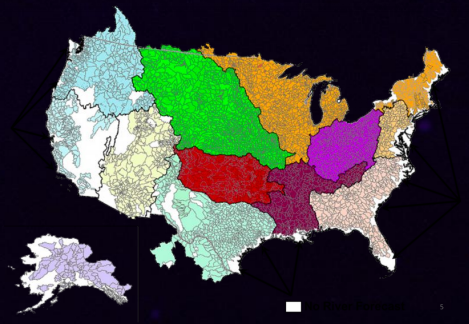
- Effort lead by IU (Wernert, McCombs, Ruan, Tuna)
- Create 3d point cloud & Mesh Model of texture/color map using tiled 2d images
  - Camera panoramas, Drone Survey
  - Future underwater shipwrecks/reefs
- Using Agisoft Photoscan software
  - More speedup from larger datasets
- Exploring OpenSource alternatives
  - Adding MPI layers needed

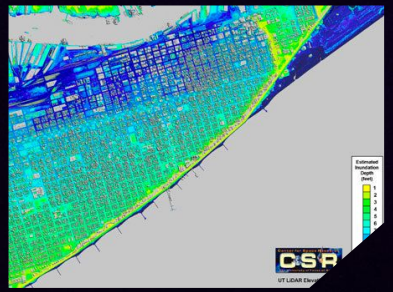










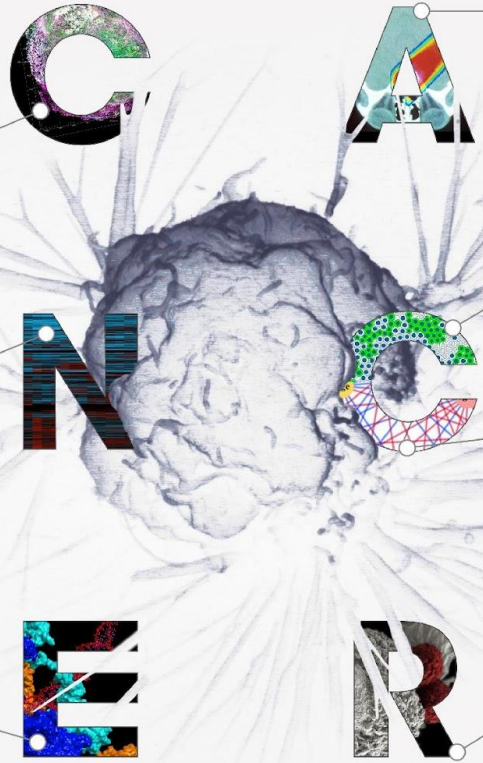


HARVEY

- Next Generation Storm Forecasting (with Penn State)
- Storm Surge Modeling (with Clint Dawson UT Austin)
- Preliminary river flooding and inundation maps (David Maidment UT Austin)
- Remote Image Integration and Assimilation (Center for Space Research, UT Austin)

The Texas Advanced Computing Center accelerates basic and applied cancer research to help save lives.

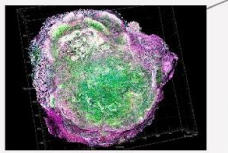
fighting



with supercomputers

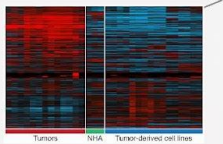
Computer Modeling

Researchers use advanced computing to model tissues, cells and drug interactions, and to design patient-specific treatments and identify new medicines.



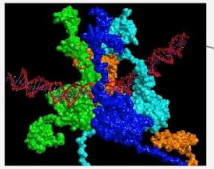
Big Data Analysis

Supercomputers allow researchers to find patterns in genomes and among patient outcomes to pinpoint risks and target treatments.



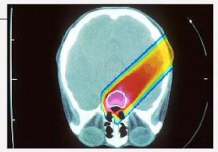
Molecular Dynamics Simulations

Simulating protein and drug interactions at the atomic level enables scientists to understand cancer and design more effective therapies.



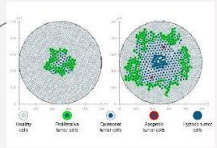
Quantum Calculations

Exploring how proton and x-ray beams interact with DNA on the quantum level helps explain why radiation treatments work and how they can be optimized.



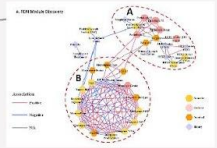
Trial Design

Researchers use TACC's advanced computers to design clinical trials that can determine the combination of dosages that will be most effective.



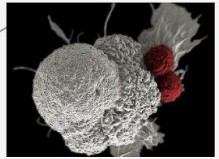
Clinical Planning

Supercomputers can test thousands of potential treatments in advance to help decide which one will work best.



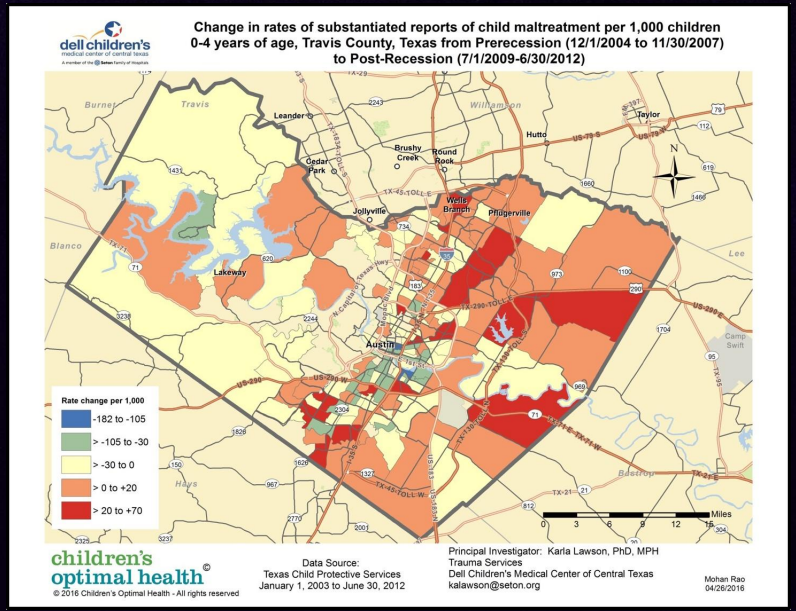
Artificial Intelligence

AI on high-performance computers can uncover relationships among complex cellular networks and reverse-engineer interventions.



TACC
TEXAS ADVANCED COMPUTING CENTER

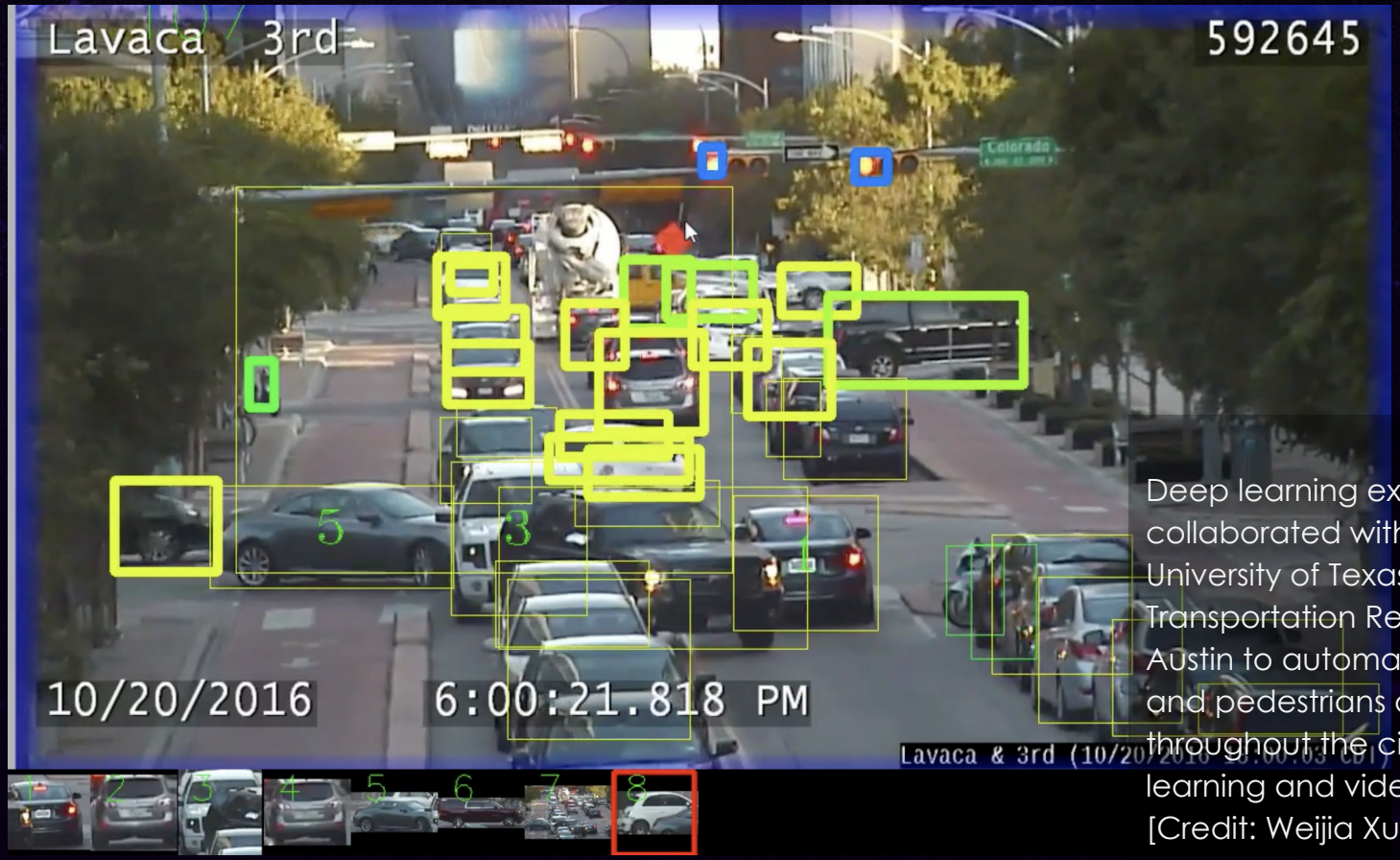
IMPROVE CHILD WELFARE THROUGH ADVANCED COMPUTING



The Texas Advanced Computing Center (TACC) at The University of Texas at Austin has teamed up with Austin, Texas-based non-profit Children's Optimal Health (COH) to provide the technical infrastructure needed to help solve issues in children's health and education throughout Central Texas, including disease, mental health and adverse childhood experiences.

A DANCE WITH ALGORITHMS

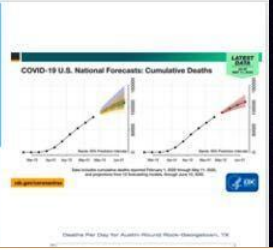
- XSEDE resources help researchers merge art and science to create systems that can understand and produce human-quality movement
- Uses deep learning techniques and relies on the collaboration of graduate students with artists to create new algorithms
- *On a single computer, running our algorithms would take years, on medium-sized resources months, but using XSEDE, we can train some of most complex models within 24 hours.* - Philippe Pasquier, professor and researcher, Simon Fraser University



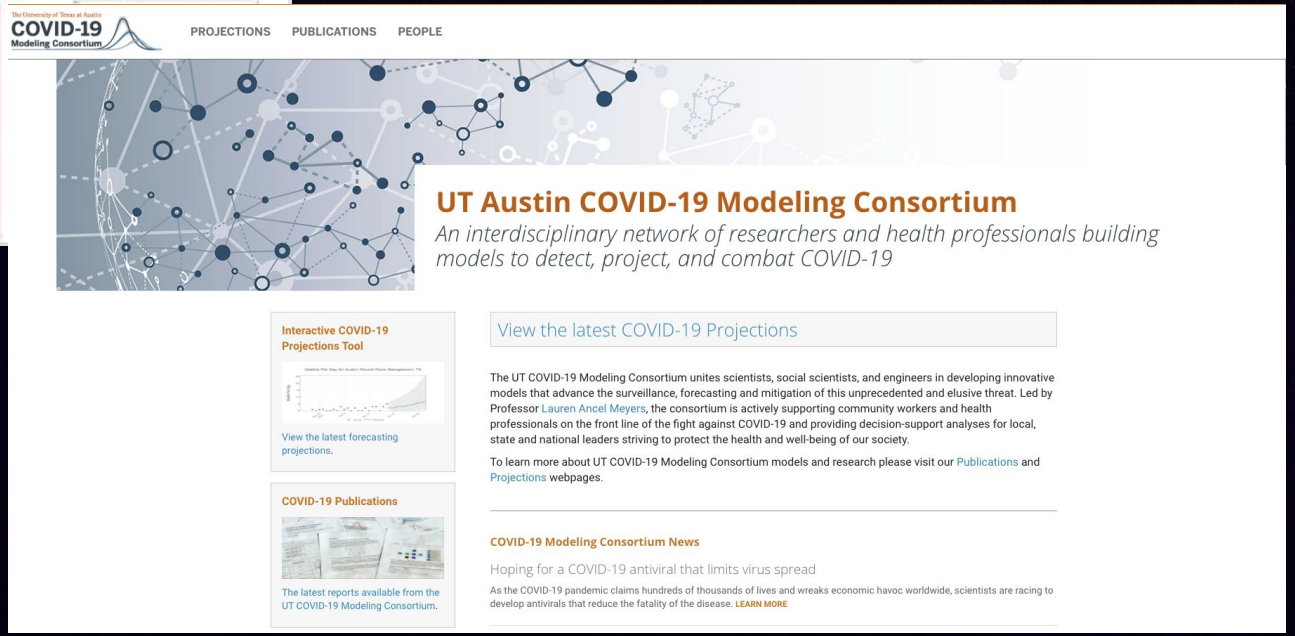
Deep learning experts from TACC collaborated with researchers at the University of Texas Center for Transportation Research and the City of Austin to automatically detect vehicles and pedestrians at critical intersections throughout the city using machine learning and video image analysis. [Credit: Weijia Xu, TACC]

Background: The COVID-19 pandemic required a rapid response to provide forecasting models to key decision makers. The COVID-19 Science Gateway is part of The University of Texas at Austin COVID-19 Modeling Consortium

Objective: Provides decision-makers with key predictive models necessary. Enables researchers to run models easily through the science gateway.



Results: Within a week the team launched an official consortium gateway, allowing people across the country to access projections about mortality rates within different states and metropolitan areas. Used by CDC, White House, local policy makers. Over 400K visits from 180+ countries. Featured on CNN, New York Times. Additional external funding of $389,400 has been identified by the client.



College of Natural Sciences

TEXAS

COVID-19 Modeling Consortium

PROJECTIONS PUBLICATIONS PEOPLE

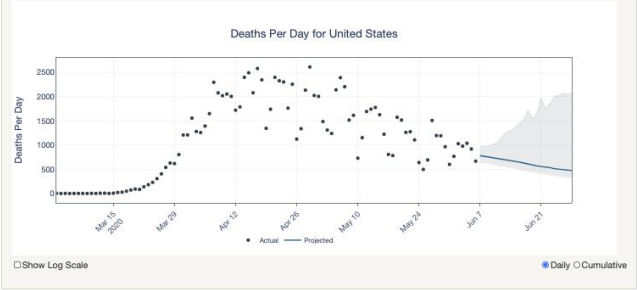
The University of Texas COVID-19 Modeling Consortium

COVID-19 Mortality Projections for US States

These graphs show both the reported and projected number of COVID-19 deaths per day across the US and for individual states. We use local data from mobile-phone GPS traces to quantify the changing impact of social-distancing measures on "flattening the curve."
Code syntax and daily updates of our forecasts are available on our UT-COVID GitHub repository.

Select your area in the dropdown menu below

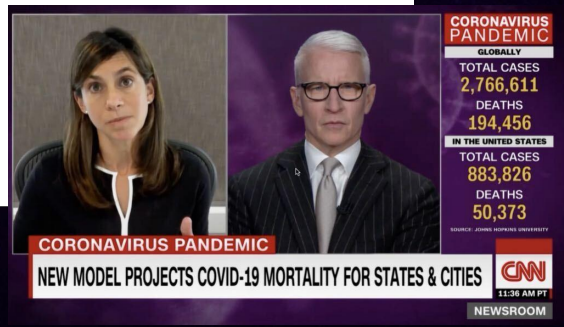
Select State: United States **Select Metro Area:** Metropolitan Statistical Areas



* Click and drag on the plot to zoom-in and double-click to zoom-out completely. The icons at the top right of each graph will provide you with additional options.

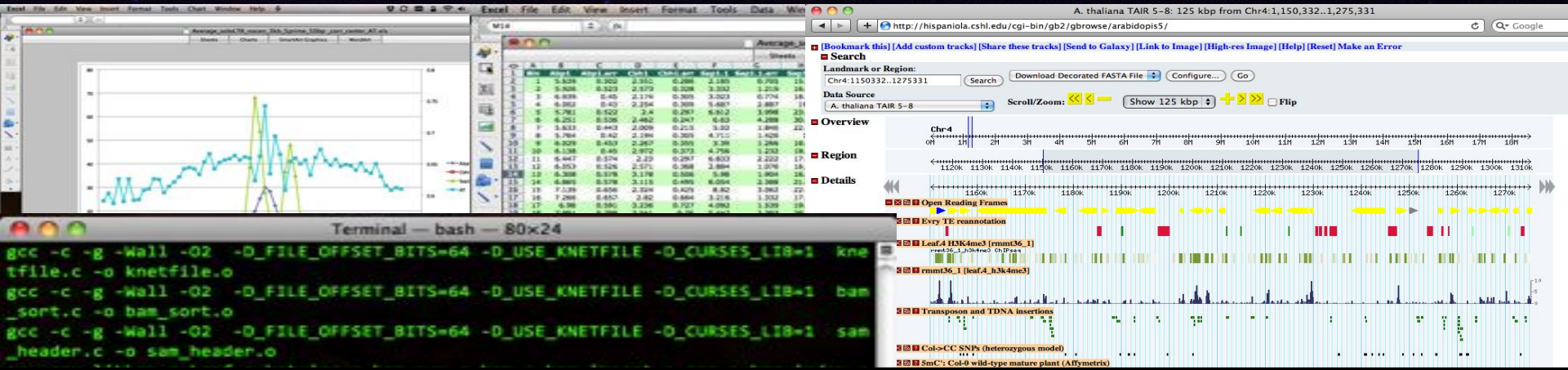
Note: On May 7, 2020 we switched to using data on confirmed and probable COVID-19 deaths to make forecasts, rather than only confirmed deaths.

Key model assumptions: The model estimates the extent of social distancing using geolocation data from mobile phones and assumes that the extent of social distancing does not change during the period of forecasting. On June 2, 2020 we started reporting an ensemble forecast, combining predictions from our original model, based on "curve-fitting," and a new SEIR-based model.

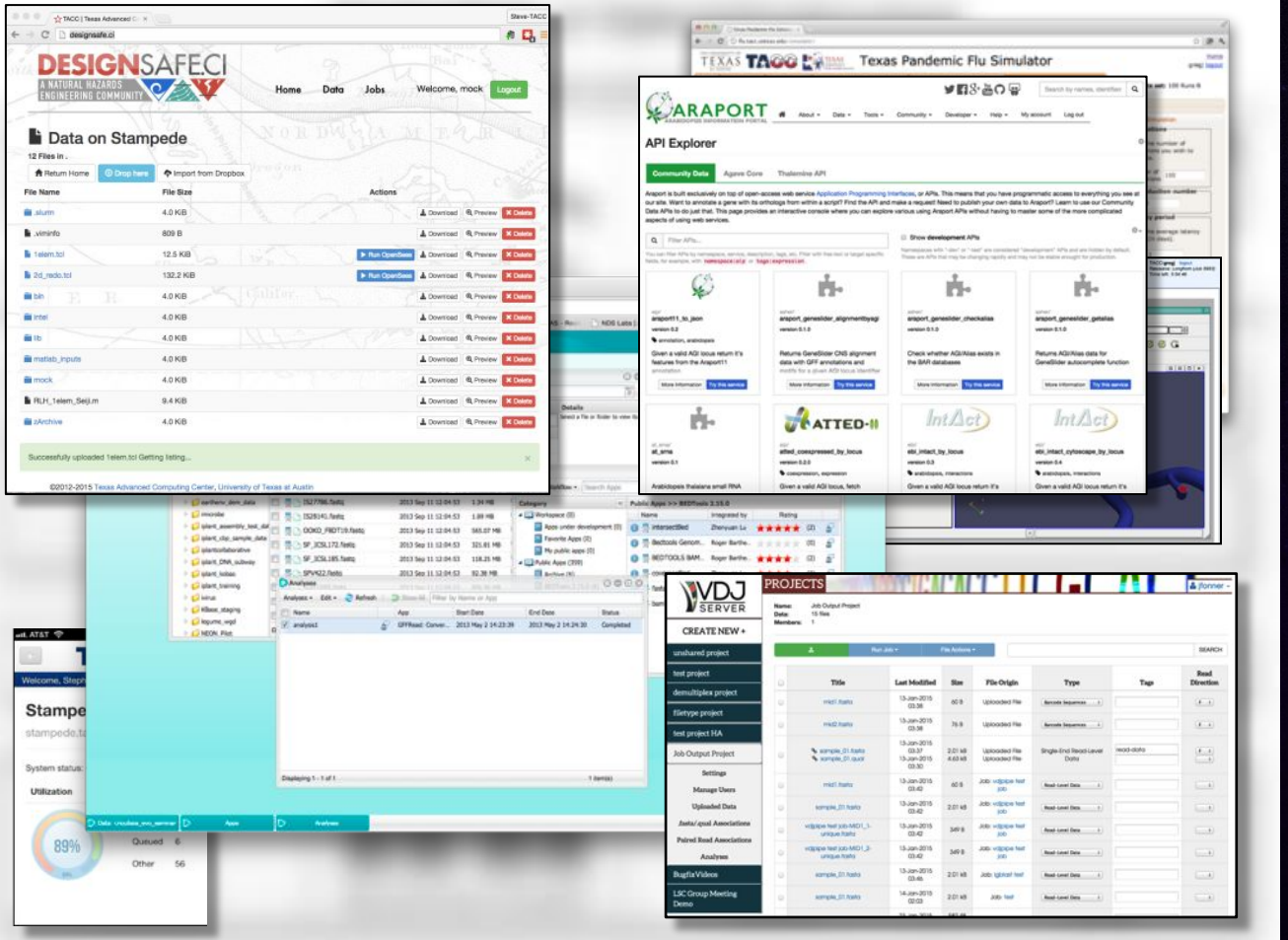


BUILD A MASSIVE STORAGE CLOUD NEXT TO INNOVATIVE, POWERFUL, USABLE COMPUTERS AT THE END OF FAST INTERNET PIPES

HOW DO WE HELP RESEARCHERS WITH SUCH DIVERSE NEEDS AND BACKGROUNDS?



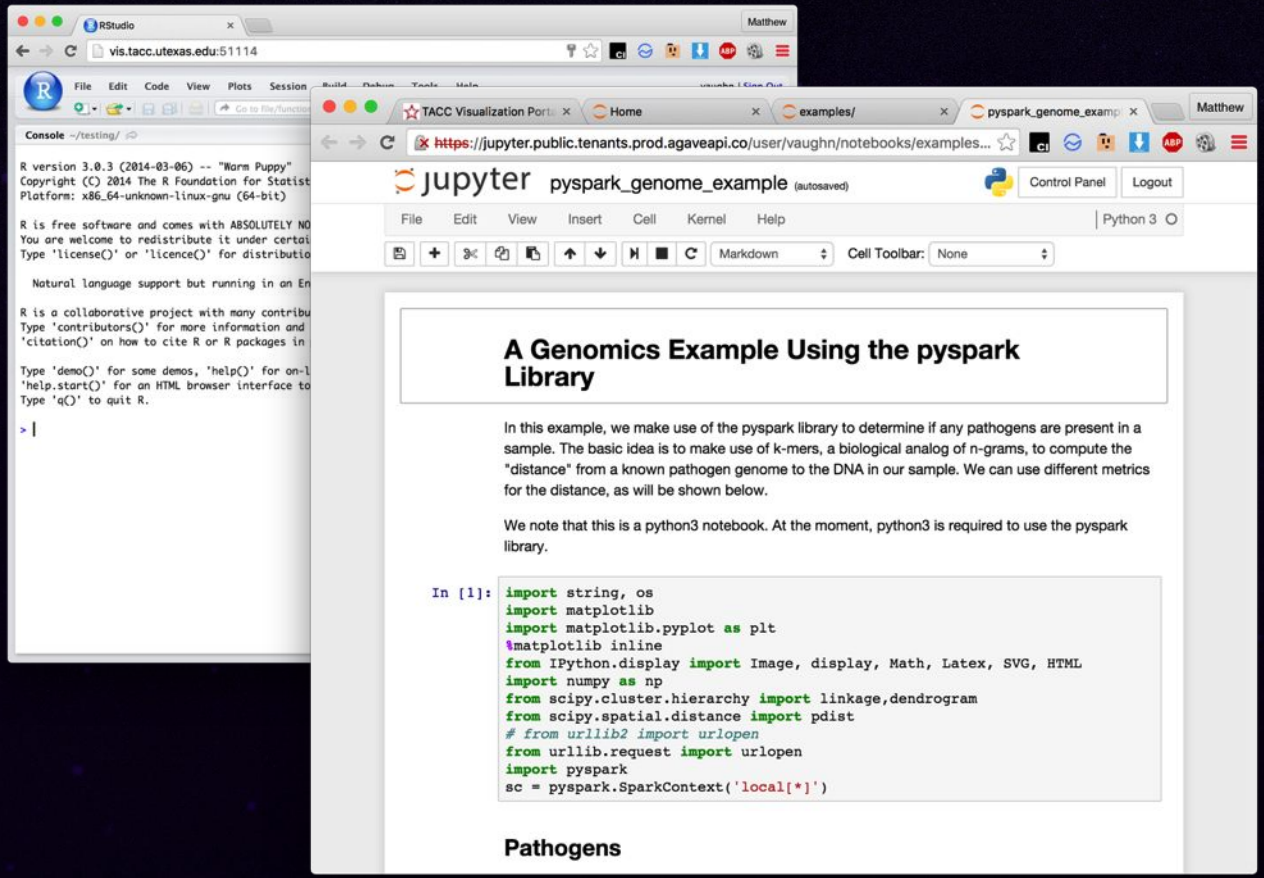
MANY DOMAIN SCIENTISTS ARE NOT EXPERTS AT COMPUTING TECHNOLOGY.
CREATE PURPOSE-BUILT, HIGHLY INTUITIVE INTERFACES



Point-and-click interfaces

- Data management, sharing, and analysis
- Publishing reproducible analysis workflows
- Discovery of new or updated tools and data
- Interactive visualization of results

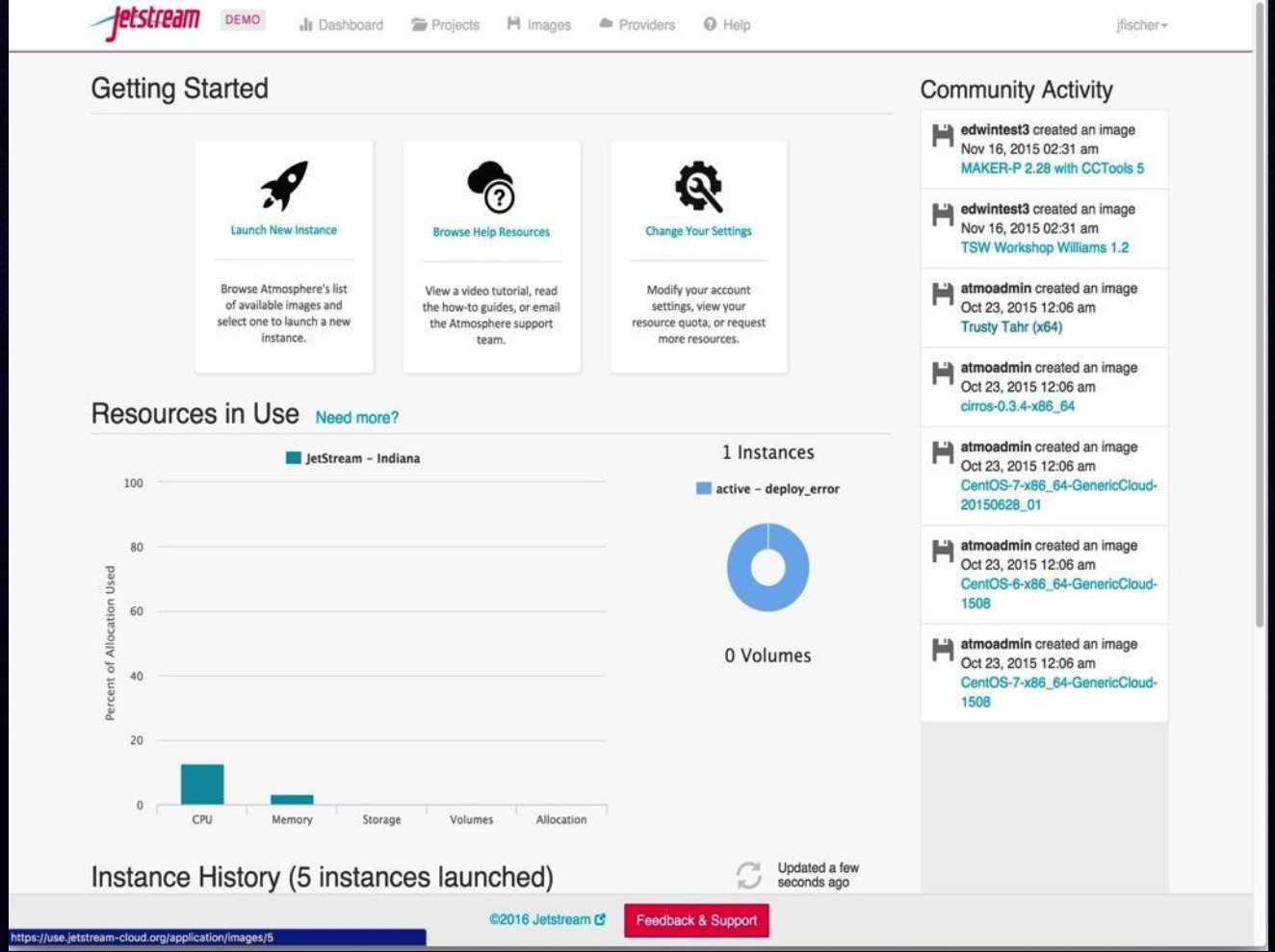
Backed by world-class computing and data capacity



Hosted SaaS

- JupyterHub notebooks
- Rstudio
- Web-based VNC

Also, backed by world-class computing and data capacity



Easy to use Cloud Computing

- Atmosphere (Cyverse)
- Jetstream (IU,UA,TACC)
- Chameleon (UC,TACC)

Cloud consoles are aimed at sysadmins and unintutive.

We're changing that with improved UX and support

- APIs are still available
- No cost to end user

GIVE EXPERTS ACCESS TO EVERY SINGLE ONE OF YOUR BUILDING BLOCKS.
WEB SERVICE APIs EVERYWHERE. AUGMENT WITH PROFESSIONAL TOOLING.

Compute

Collaborate

Je'aime Powell, jpowell@tacc.utexas.edu
Charlie Dey, charlie@tacc.utexas.edu

Jupyter

https://designsafe-ci.org

https://jupyter.designsafe-ci.org

https://www.designsafe-ci.org/rw/user-guides/tools-applications/jupyter/

What are Jupyter Notebooks?

A web-based, interactive computing tool for capturing the whole computation process: developing, documenting, and executing code, as well as communicating the results.

How do Jupyter Notebooks Work?

An open notebook has exactly one interactive session connected to a kernel which will execute code sent by the user and communicate back results. This kernel remains active if the web browser window is closed, and reopening the same notebook from the dashboard will reconnect the web application to the same kernel.

What's this mean?

Notebooks are an interface to kernel, the kernel executes your code and outputs back to you through the notebook. The kernel is essentially our programming language we wish to interface with.

Jupyter Notebooks, Structure

- Code Cells

  Code cells allow you to enter and run code
  Run a code cell using Shift-Enter

- Markdown Cells

  Text can be added to Jupyter Notebooks using Markdown cells. Markdown is a popular markup language that is a superset of HTML.

Jupyter Notebooks, Structure

- Markdown Cells

You can add headings:

```
# Heading 1
# Heading 2
## Heading 2.1
## Heading 2.2
```

You can add lists

```
1. First ordered list item
2. Another item
⋅⋅* Unordered sub-list.
1. Actual numbers don't matter, just that it's a number
⋅⋅1. Ordered sub-list
4. And another item.
```

Jupyter Notebooks, Structure

- Markdown Cells

```
pure HTML
  <dl>
    <dt>Definition list</dt>
    <dd>Is something people use sometimes.</dd>

    <dt>Markdown in HTML</dt>
    <dd>Does *not* work **very** well. Use HTML <em>tags</em>.</dd>
  </dl>

And even, Latex!

    $e^{i\pi} + 1 = 0$
```

Jupyter Notebooks, Workflow

Typically, you will work on a computational problem in pieces, organizing related ideas into cells and moving forward once previous parts work correctly. This is much more convenient for interactive exploration than breaking up a computation into scripts that must be executed together, as was previously necessary, especially if parts of them take a long time to run.

Jupyter Notebooks, Workflow

Let a traditional paper lab notebook be your guide:

Each notebook keeps a historical (and dated) record of the analysis as it's being explored.

The notebook is not meant to be anything other than a place for experimentation and development.

Notebooks can be split when they get too long.

Notebooks can be split by topic, if it makes sense.

Jupyter Notebooks, Shortcuts

- `Shift-Enter`: **run cell**
  - Execute the current cell, show output (if any), and jump to the next cell below. If `Shift-Enter` is invoked on the last cell, a new code cell will also be created. Note that in the notebook, typing `Enter` on its own *never* forces execution, but rather just inserts a new line in the current cell. `Shift-Enter` is equivalent to clicking the `Cell | Run` menu item.

Jupyter Notebooks, Shortcuts

- `Ctrl-Enter`: **run cell in-place**
  - Execute the current cell as if it were in “terminal mode”, where any output is shown, but the cursor *remains* in the current cell. The cell’s entire contents are selected after execution, so you can just start typing and only the new input will be in the cell. This is convenient for doing quick experiments in place, or for querying things like filesystem content, without needing to create additional cells that you may not want to be saved in the notebook.

Jupyter Notebooks, Shortcuts

- **`Alt-Enter`: run cell, insert below**
  - Executes the current cell, shows the output, and inserts a *new* cell between the current cell and the cell below (if one exists). (shortcut for the sequence `Shift-Enter`,`Ctrl-m a`. (`Ctrl-m a` adds a new cell above the current one.))
- **`Esc` and `Enter`: Command mode and edit mode**
  - In command mode, you can easily navigate around the notebook using keyboard shortcuts. In edit mode, you can edit text in cells.

TACC
TEXAS ADVANCED COMPUTING CENTER
WWW.TACC.UTEXAS.EDU

TEXAS
The University of Texas at Austin

BigData-Hack-2022 Day 1

Let's do this!

PRESENTED BY:

Computational Thinking Objectives

The student will …

- Learn the about the concept of “computational thinking”
- Practice algorithm implementation through abstraction
- Learn about the concept of pseudo code
- Apply computational thinking to the equation for a straight line

Back when mathematicians were computers and computers were calculators...

- Initially all programming was dedicated to translating math formulas.
- The work lead to the language FORmula TRANslation.

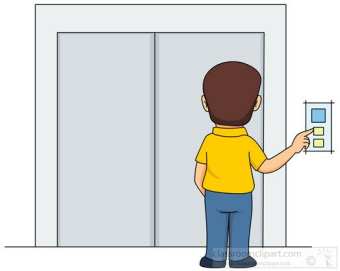
“Computational Thinking is the translation of ideas into computer code” ~Victor Eijkhout

Mathematical Thinking

- Number of people an elevator takes per day
- Speed (velocity) of an elevator
- Distribution of people in an elevator

Computational Thinking

- If there are X # of people expected to use elevators, how many should be installed?
- If someone at floor 0 presses the call button and there are available cars on floors 5 and 9, which car should respond?



The Process of Forming Logic (Think teaching a 3 year-old)

How would you tell a three(3) year old family member to get your keys out of the drawer in your room ?

Requirements, Logic, Algorithms, and Parameters

Requirements - what elements are needed before the job can be taken on

Logic - a system or set of principles underlying the arrangements of elements in a computer or electronic device so as to perform a specified task [an order in which to do a task]

Algorithm - a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer [logic + calculations = algorithm]

Parameters - a limit or boundary that defines the scope of a particular process or activity [limits set on an algorithm = parameters]

Making a PB&J Sandwich

Requirements - what elements are needed before the job can be taken on

Parameters - a limit or boundary that defines the scope of a particular process or activity [limits set on an algorithm = parameters]

Logic - a system or set of principles underlying the arrangements of elements in a computer or electronic device so as to perform a specified task [an order in which to do a task]

Algorithm - a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer [logic + calculations = algorithm]

Finding a definition in a dictionary

Requirements - what elements are needed before the job can be taken on

Parameters - a limit or boundary that defines the scope of a particular process or activity [limits set on an algorithm = parameters]

Logic - a system or set of principles underlying the arrangements of elements in a computer or electronic device so as to perform a specified task [an order in which to do a task]

Algorithm - a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer [logic + calculations = algorithm]

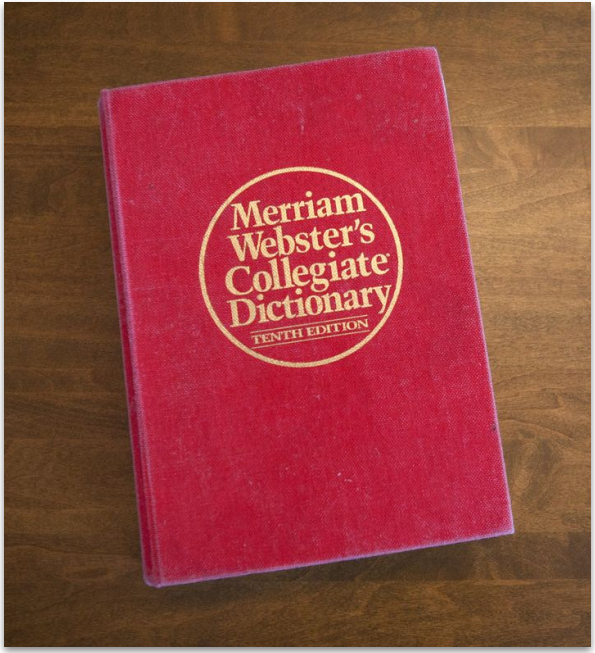
Given (Input):

- Dictionary
- Word (string)

Find (Output):

- Definition

Define your algorithm



Sorting

Requirements - what elements are needed before the job can be taken on

Parameters - a limit or boundary that defines the scope of a particular process or activity [limits set on an algorithm = parameters]

Logic - a system or set of principles underlying the arrangements of elements in a computer or electronic device so as to perform a specified task [an order in which to do a task]

Algorithm - a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer [logic + calculations = algorithm]

Given:

A bag of potatoes

Problem:

Sort the bag of potatoes from smallest to largest

Algorithm:????

Think outside the "Box"

Requirements - what elements are needed before the job can be taken on

Parameters - a limit or boundary that defines the scope of a particular process or activity [limits set on an algorithm = parameters]

Logic - a system or set of principles underlying the arrangements of elements in a computer or electronic device so as to perform a specified task [an order in which to do a task]

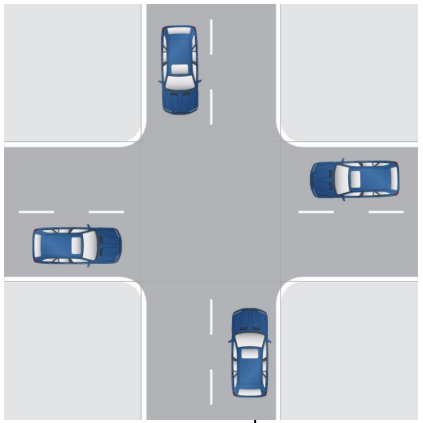
Algorithm - a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer [logic + calculations = algorithm]

Problem:

4 automated cars come to an intersection at the same time.

Who goes first?

Algorithm: ???



What decisions did you make?

Requirements - what elements are needed before the job can be taken on

Parameters - a limit or boundary that defines the scope of a particular process or activity [limits set on an algorithm = parameters]

Logic - a system or set of principles underlying the arrangements of elements in a computer or electronic device so as to perform a specified task [an order in which to do a task]

Algorithm - a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer [logic + calculations = algorithm]

What was the last meal you ate?

What were the defining parameters on why you chose that meal?

Where do we go from here?

Look at each problem you are going to tackle, and figure out the requirements - what is needed to solve? Figure out the logic on how to solve it, and apply the algorithm.

Think about a Scientific Process

Let's meet Joe.

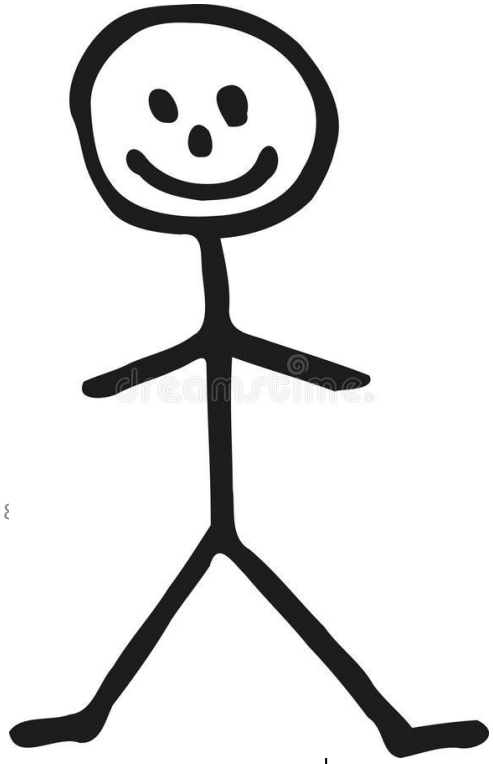
Joe might get sick.

Joe will be sick for 5 days.

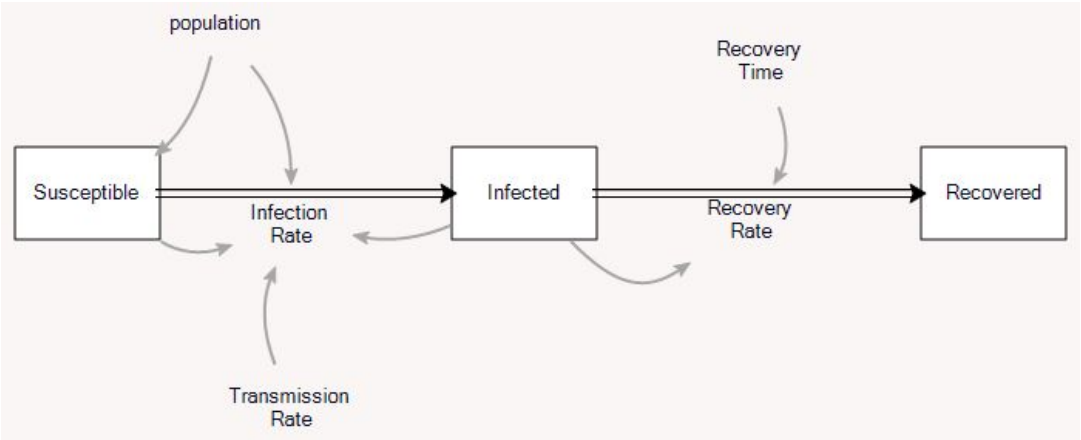
After 5 days, Joe gets better.

Once Joe gets better, Joe can no longer get sick.

How would we "code" Joe?



The SIR Model



Task 1 - Code Joe

Variables to hold data

Mathematical Operations to do math :)

Conditionals to make decisions

Loops to repeat our process

Functions/Subroutines to reuse code

Objects or Classes to define our "things"

Let's meet Joe.

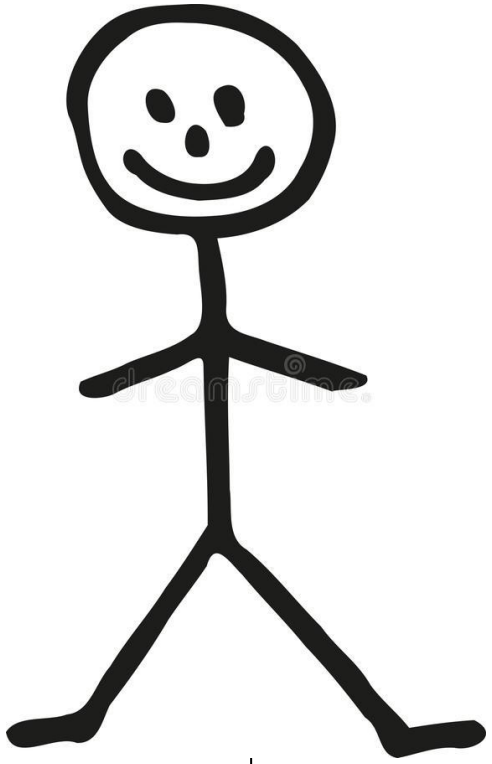
Joe might get sick.

Joe will be sick for 5 days.

After 5 days, Joe gets better.

Once Joe gets better, Joe can no longer get sick.

Let's "code" Joe.



Introduction to Python

Hello World!
Data types
Variables
Arithmetic operations
Relational operations
Input/Output
Control Flow

Do not forget:

Indentation matters!

Python

```
print(“Hello World!”)
```

Let's type that line of code into a Code Cell, and hit Shift-Enter:

```
Hello World!
```

Python

```
print(5)
print(1+1)
```

Let's add the above into another Code Cell, and hit Shift-Enter

```
5
2
```

Python - Variables

You will need to store data into variables
You can use those variables later on
You can perform operations with those variables
Variables are declared with a **name**, followed by '=' and a **value**

An integer, string,…
When declaring a variable, **capitalization** is important:
'A' <> 'a'

Python - Variables

in a code cell:

```
five = 5
one = 1
twodot = 2.0
print (five)
print (one + one)
message = “This is a string”
print (message)
```

Notice: We're not "typing" our variables, we're just setting them and allowing Python to type them for us.

Python – Data Types

in a code cell:

```
integer_variable = 100
floating_point_variable = 100.0
string_variable = “Name”
```

Notice: We're not "typing" our variables, we're just setting them and allowing Python to type them for us.

Python – Data Types

Variables have a type

You can check the type of a variable by using the type() function:

```
print (type(integer_variable))
```

It is also possible to change the type of some basic types:

`str(int/float)`: converts an integer/float to a string
`int(str)`: converts a string to an integer
`float(str)`: converts a string to a float

Be careful: you can only convert data that actually makes sense to be transformed

Python – Arithmetic Operations

+	Addition	1 + 1 = 2
-	Subtraction	5 – 3 = 2
/	Division	4 / 2 = 2
%	Modulo	5 % 2 = 1
*	Multiplication	5 * 2 = 10
//	Floor division	5 // 2 = 2
**	To the power of	2 ** 3 = 8

Python – Arithmetic Operations

Some experiments:

```
print (5/2)
print (5.0/2)
print ("hello" + "world")
print ("some" + 1)
print ("number" * 5)
print (3+5*2)
```

Python – Arithmetic Operations

Some more experiments:

```
number1 = 5.0/2
number2 = 5/2
```

what **`type()`** are they?

```
type(number1)
type(number2)
```

now, convert number2 to an integer:

```
int(number2)
```

Python – Making the output prettier

Let put the following into a new Code Cell:

```
print ("The number that you wrote was : ", numIn)
print ("The number that you wrote was : %d" % numIn)

print ("the string you entered was: ", stringIn)
print ("the string you entered was: %s" % stringIn)
```

Want to make it prettier?

\n for a new line

\t to insert a tab

```
print (" your string: %s\n your number: %d" %(stringIn, numIn))
```

for floating points, use %f

Python – Control Flow

So far we have been writing instruction after instruction where every instruction is executed

What happens if we want to have instructions that are only executed if a given condition is true?

Python – if/else/elif

Let's look at some example of booleans.

type the following into a code cell

```
a = 2
b = 5

print (a>b)
print (a<b)
print (a == b)
print (a != b)
print (b>a or a==b)
print (b<a and a==b)
```

Python – if/else/elif

The if/else construction allows you to define conditions in your program

(Don’t forget your indentation!!)

```
if conditionA:
    statementA
elif conditionB:
    statementB
else:
    statementD
this line will always be executed (after the if/else)
```

Python – if/else/elif

The if/else construction allows you to define conditions in your program

(Indentation is IMPORTANT!)

```
if conditionA:
    statementA
elif conditionB:
    statementB
else:
    statementD
this line will always be executed (after the if/else)
```

conditions are a datatype known as booleans, they can only be true or false

Python – if/else/elif

A simple example

```
simple_input = input(“Please enter a number: “)
if (int(simple_input)>10):
    print ("You entered a number greater than 10")
else:
    print ("you entered a number less than 10")
```

Python – if/else/elif

You can also nest if statements together:

```
if (condition1):
    statement1
    if (condition2):
        statement2
    else:
        if (condition3):
            statement3 # when is this statement executed?
else:  # which ‘if’ does this ‘else’ belong to?
    statement4  # when is this statement executed?
```

Python – For Loops

When we need to iterate, execute the same set of instructions over and over again… we need to loop! and introducing range()

(Indentation is IMPORTANT!)

```
for x in range(0, 3):
    print ("Let's go %d" % x)
```

Python – For Loops, nested loops

When we need to iterate, execute the same set of instructions over and over again… we need to loop! and introducing range()

```
for x in range(0, 3):
    for y in range(0,5):
       print ("Let's go %d %d" % (x,y))
```

Python – While Loops

Sometimes we need to loop while a condition is true...

(remember to indent!)

```
i = 0                # Initialization
while (i < 10):      # Condition
    print (i)        # do_something
    i = i + 1        # Why do we need this?
```

Exercise:

using a while loop, find the prime numbers less than 1000

Task 1 - Code Joe

Variables to hold data

Mathematical Operations to do math :)

Conditionals to make decisions

Loops to repeat our process

Functions/Subroutines to reuse code

Objects or Classes to define our "things"

Let's meet Joe.

Joe might get sick.

Joe will be sick for 5 days.

After 5 days, Joe gets better.

Once Joe gets better, Joe can no longer get sick.

Let's "code" Joe.