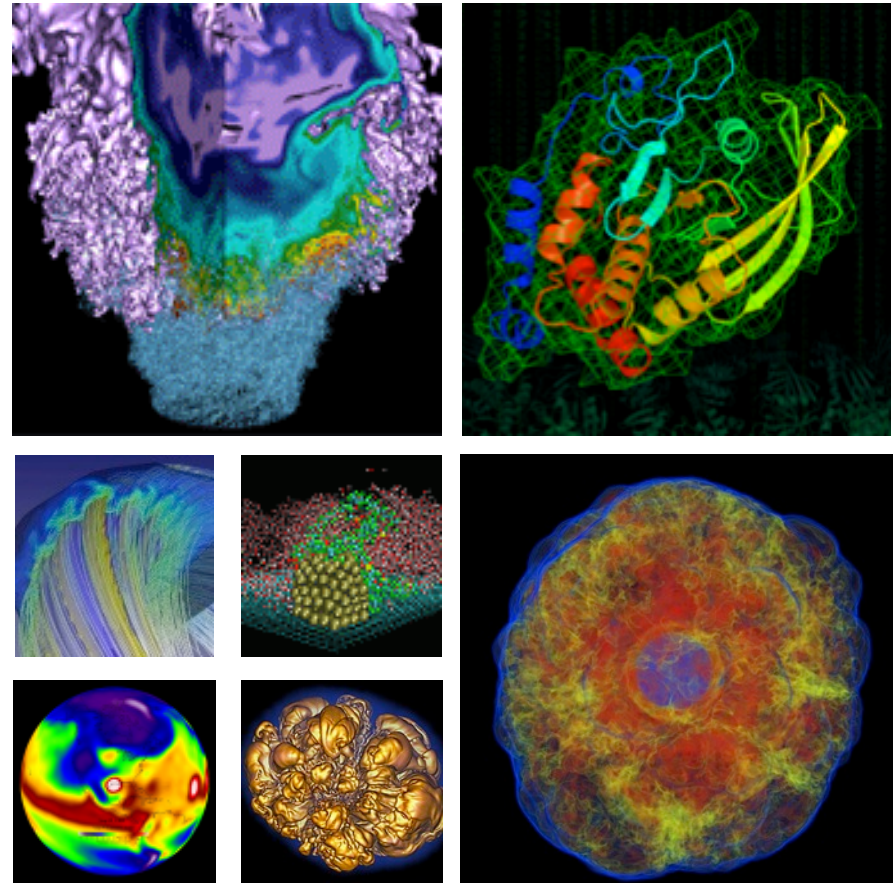
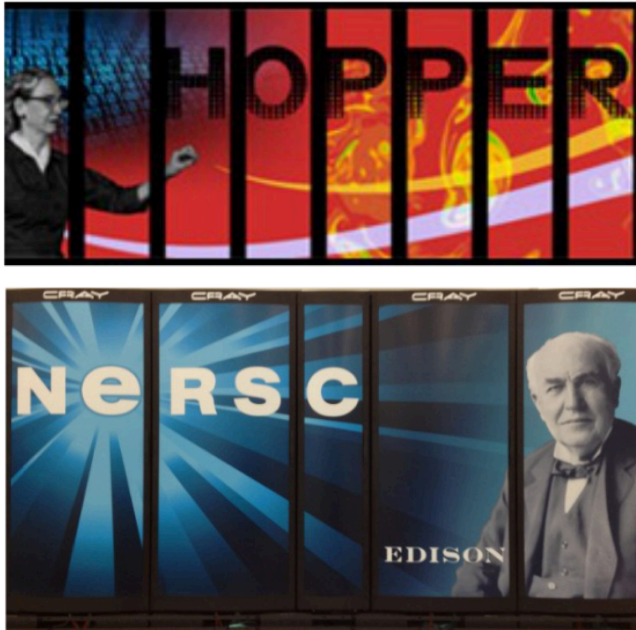


# Databases and Data analytic frameworks at NERSC

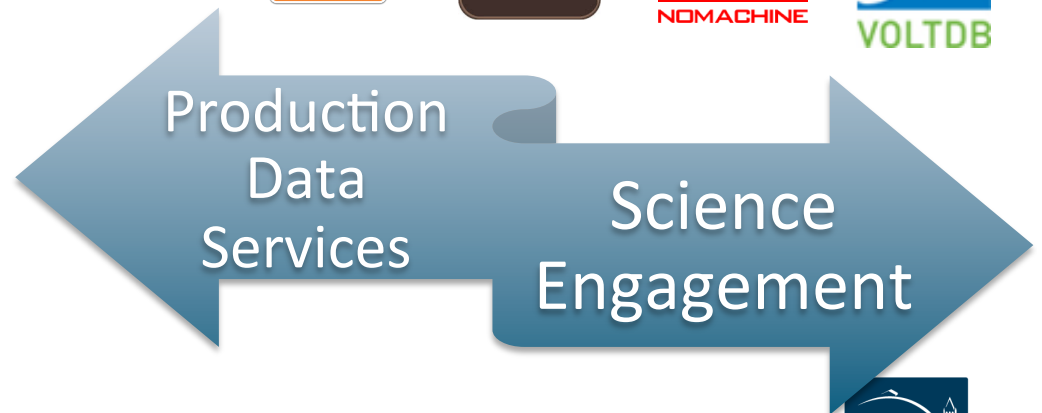


Yushu Yao (yyao@lbl.gov)

# NERSC Data Analytic Services



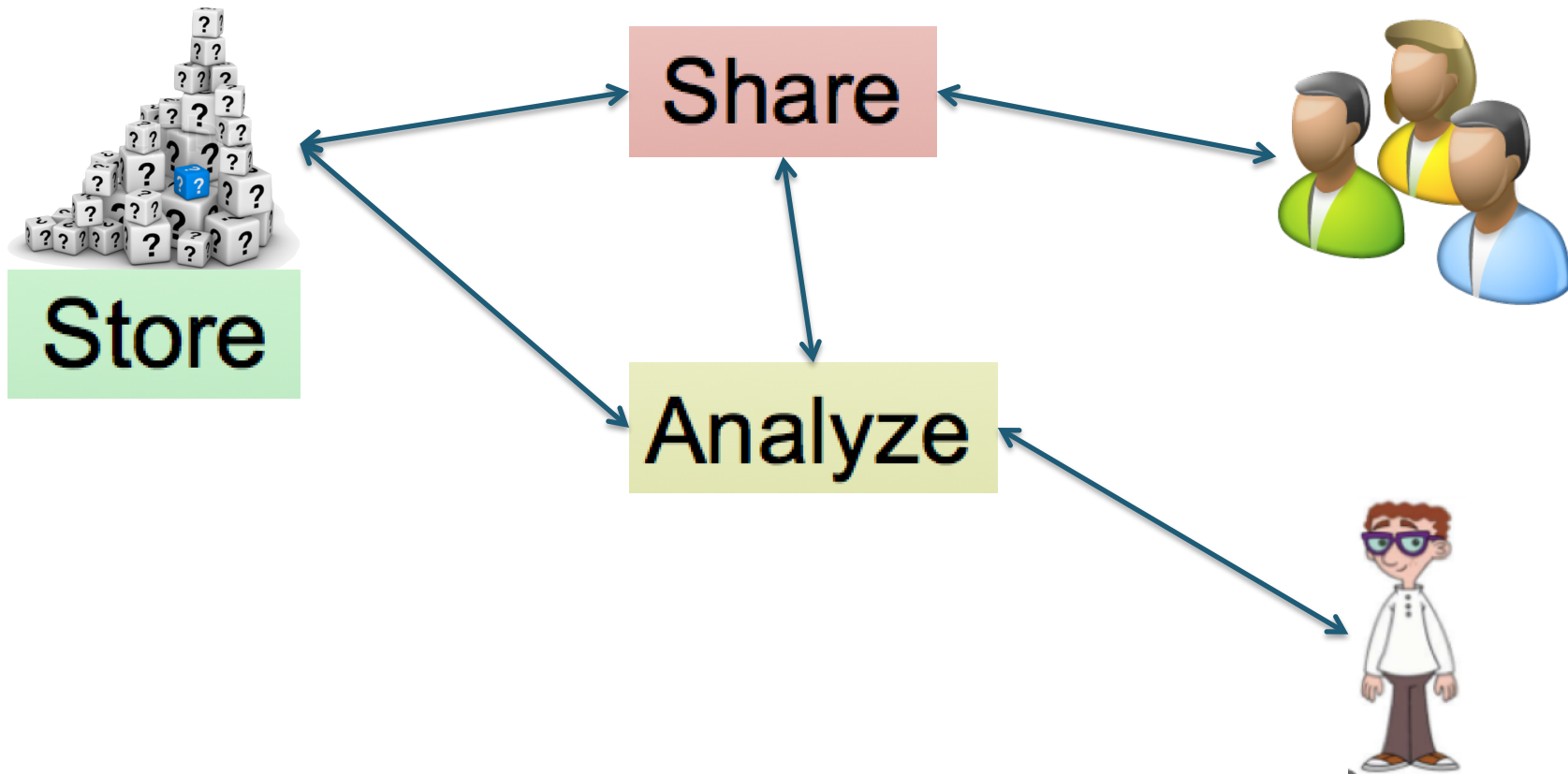
**Big and Diverse Computing Facility**  
6000+ Users, 700+ Projects  
3+ PetaFlops (20+pf more coming)  
50+ PB Storage



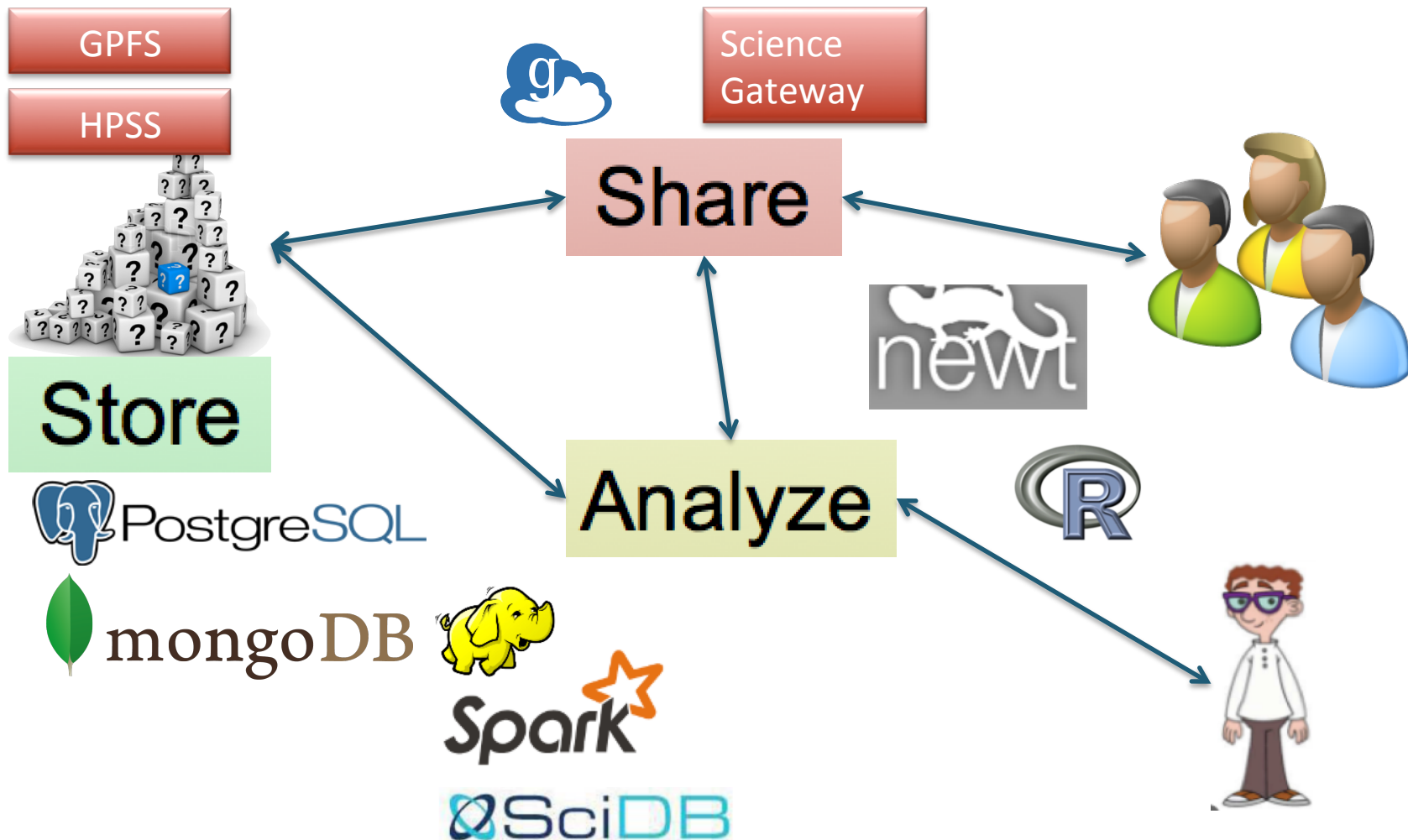
**Usability  
Scale**

**Management  
Analysis  
Share**

# Store/Share/Analyze Data At NERSC



# Store/Share/Analyze Data At NERSC



# Science Gateway Services



- **Publish data on the web**
  - Create a www directory in your project space and put your data on the web
- **Build sophisticated web portals**
  - Build full stack web applications for your science at NERSC using Python/Django, PHP, Ruby on Rails
- **NEWT – the NERSC REST API**
  - Use the NEWT REST HTTP API to access NERSC HPC resources directly from your web apps.
  - Support for Authentication, Jobs, Commands, Files, Persistent Store, NIM, System Information at NERSC

Gateway examples - <http://portal.nersc.gov>

NEWT – <https://newt.nersc.gov>

Science Gateway information <http://www.nersc.gov/users/science-gateways/>

# Example Gateway: Materials Project



<http://www.materialsproject.org>

The screenshot shows the Materials Project website homepage. At the top, there is a navigation bar with links for Home, Apps, Resources, About, and References, along with a user profile for scholia@lbl.gov and a Logout option. The main header features the 'MATERIALS PROJECT' logo and a search bar with a placeholder text 'e.g. explore Fe2O3 or Li-Fe-O pd'. Below the search bar, there are 'Database Statistics' showing 30732 materials, 3044 bandstructures, 405 intercalation batteries, and 15175 conversion batteries. A 'We're hiring!' notice is also present. The main content area is divided into several toolboxes: 'Materials Explorer' (search for materials), 'Lithium Battery Explorer' (find candidate materials), 'Crystal Toolkit' (convert CIF and VASP files), 'Phase Diagram App' (computational phase diagrams), 'Reaction Calculator' (calculate enthalpy), and 'Structure Predictor' (predict new compounds). At the bottom, there are sections for 'Tutorials' (Materials Project Tube) and 'Press Highlights' (The New York Times), and a 'Latest News' section for 'pymatgen v2.5.2'.

- Web Server Hosted at NERSC
- Jobs Submitted to NERSC Systems as needed
- Data Stored in different DB and Storage Systems
- **You can do this too**

# Traditional SQL Database Services

---



- **PostgreSQL and MySQL**
- **Good For:**
  - Structured, Relational Data
  - Mid-Size,  $\leq$ several GB in total
  - Transactional Operations



# MongoDB

---



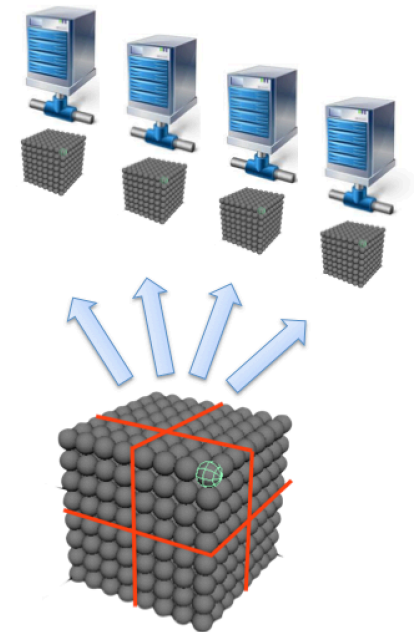
- **Key-value pair / Text database**
- **Good For:**
  - Un-Structured, Text Data
  - Mid-Size to Large, e.g. 10 GB of Text
  - E.g: for metadata that has ever changing schema

To request a MongoDB/PostgreSql/MySQL database:

<https://www.nersc.gov/users/science-gateways/science-gateway-databases/>

# SciDB For High Usability Big Data Analytic

- **Why?** It's painful to manage and analyze terabytes of data. Need a unified solution that's easy to use.
- **What?** SciDB is a parallel database for array-structured data, great for **Terabytes** of:
  - Time series, spectrums, imaging, etc
- The greatest benefit of SciDB is:
  - **Usability:** Use HPC hardware without learning parallel programming and parallel I/O.



SciDB  
Distribute a  
big array on  
many nodes

# Spark for In-Memory Map-Reduce

---

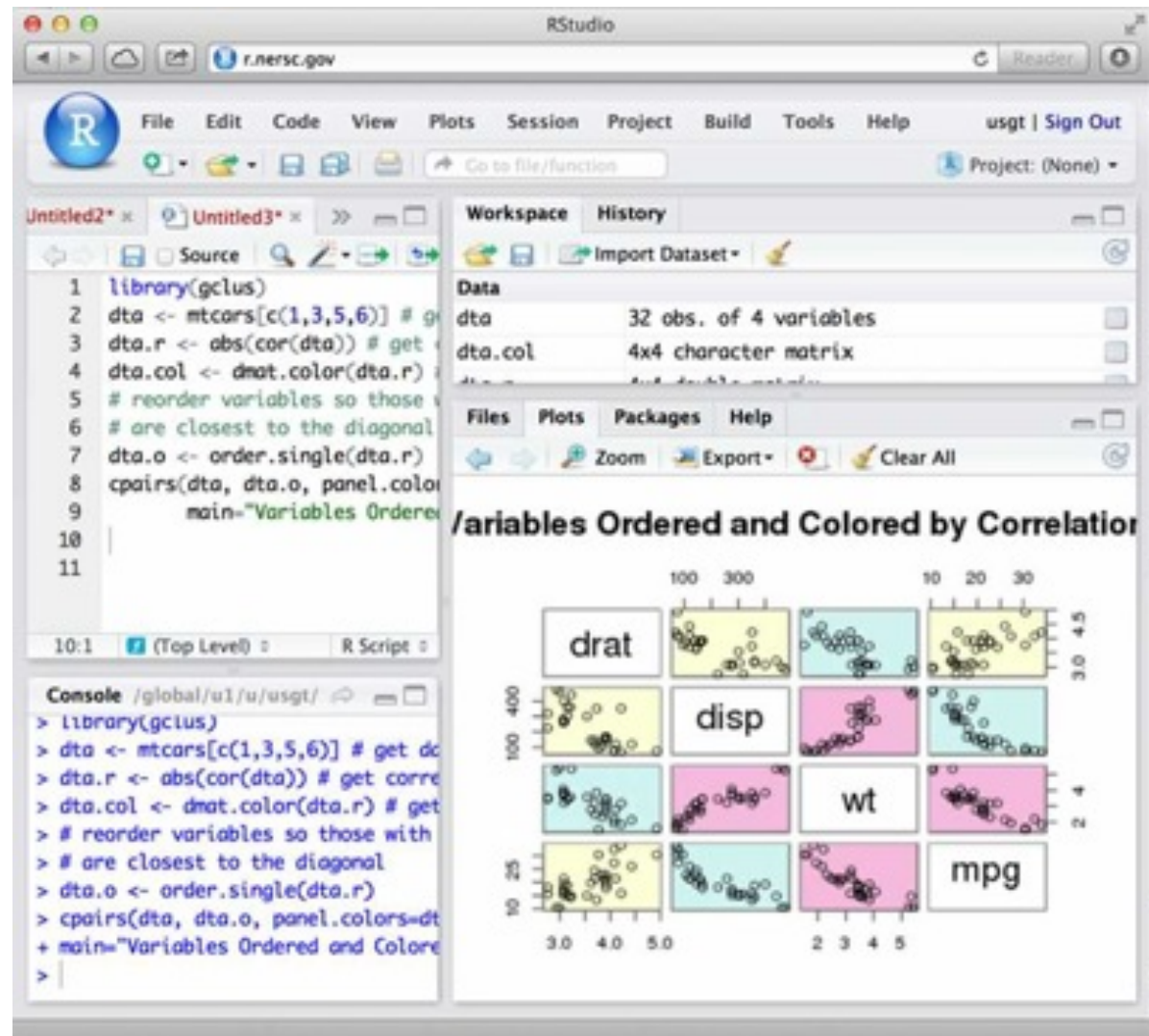


<https://www.nersc.gov/users/software/data-visualization-and-analytics/spark-distributed-analytic-framework>

- **Very efficient in-memory map-reduce data analytic framework.**

# RStudio Service(Beta) – r.nersc.gov

Any NERSC user can go to [r.nersc.gov](http://r.nersc.gov) and login with NIM Username/Password



# iPython Notebook Service (Coming Soon)

IP[y]: Notebook    Lecture\_3\_Scipy (autosaved)    Logout

File   Edit   View   Insert   Cell   Kernel   Help

Markdown    Cell Toolbar: None    Env: np18py27-1.9

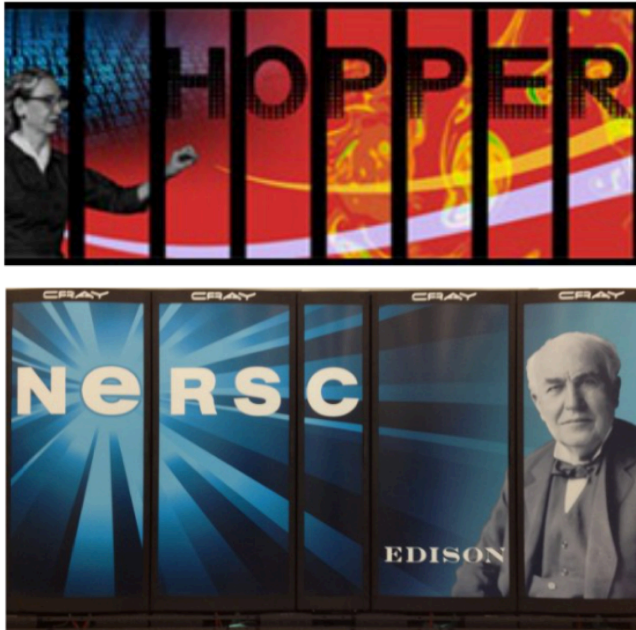
```
In [6]: x = linspace(0, 10, 100)

fig, ax = subplots()
for n in range(4):
    ax.plot(x, jn(n, x), label=r"$J_{%d}(x)$" % n)
ax.legend();
```

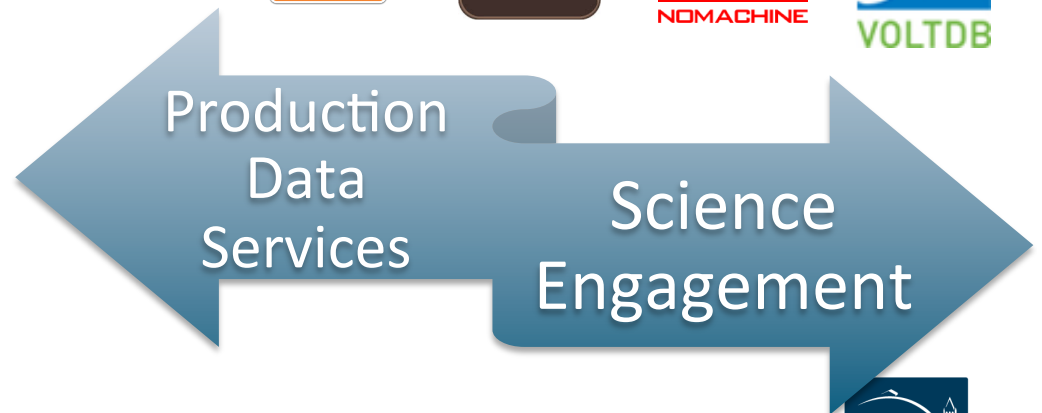
```
In [7]: # zeros of Bessel functions
n = 0 # order
m = 4 # number of roots to compute
jn_zeros(n, m)
```

Out[7]: array([ 2.40482556, 5.52007811, 8.65372791, 11.79153444])

# NERSC Data Analytic Services



**Big and Diverse Computing Facility**  
6000+ Users, 700+ Projects  
3+ PetaFlops (20+pf more coming)  
50+ PB Storage



# Links



- **Science Gateways:**
  - <http://www.nersc.gov/users/science-gateways/>
- **Database Request Form**
  - <https://www.nersc.gov/users/science-gateways/science-database-request-form/>
- **SciDB**
  - <http://www.nersc.gov/users/computational-systems/testbeds/scidb/>
- **Rstudio**
  - <http://r.nersc.gov>
- **Spark**
  - <https://www.nersc.gov/users/software/data-visualization-and-analytics/spark-distributed-analytic-framework/>