# Accelerated Materials Design

## through High-throughput First-Principles Calculations and Data Mining

Kristin Persson

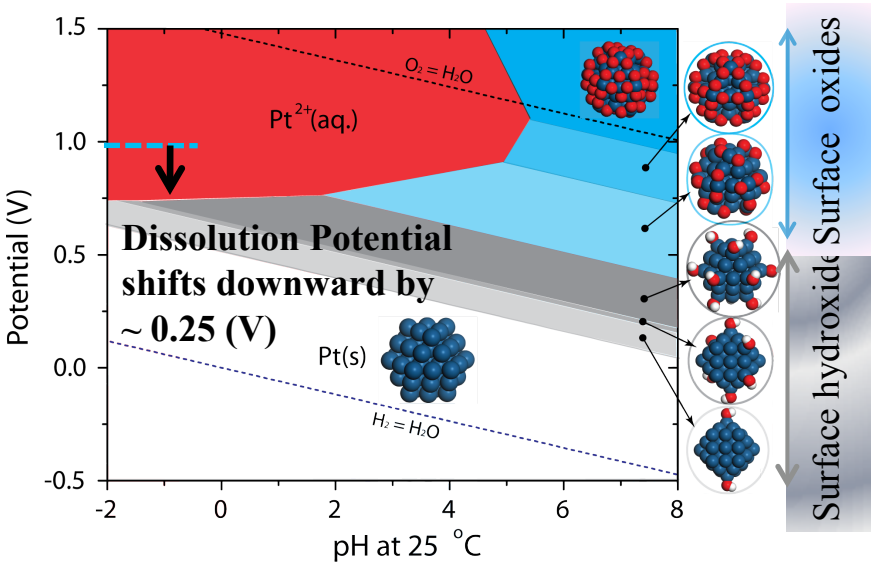
Environmental and Energy Technology Division (LBNL)

# Outline

- Materials Science – to the rescue for a sustainable energy future
- A crash course on density functional theory
- Not a exascale poster child
- Ok – let's say I solve the computing – does data-driven materials design work???
- The Materials Project - Towards the Materials Genome

# Materials can play a critical role

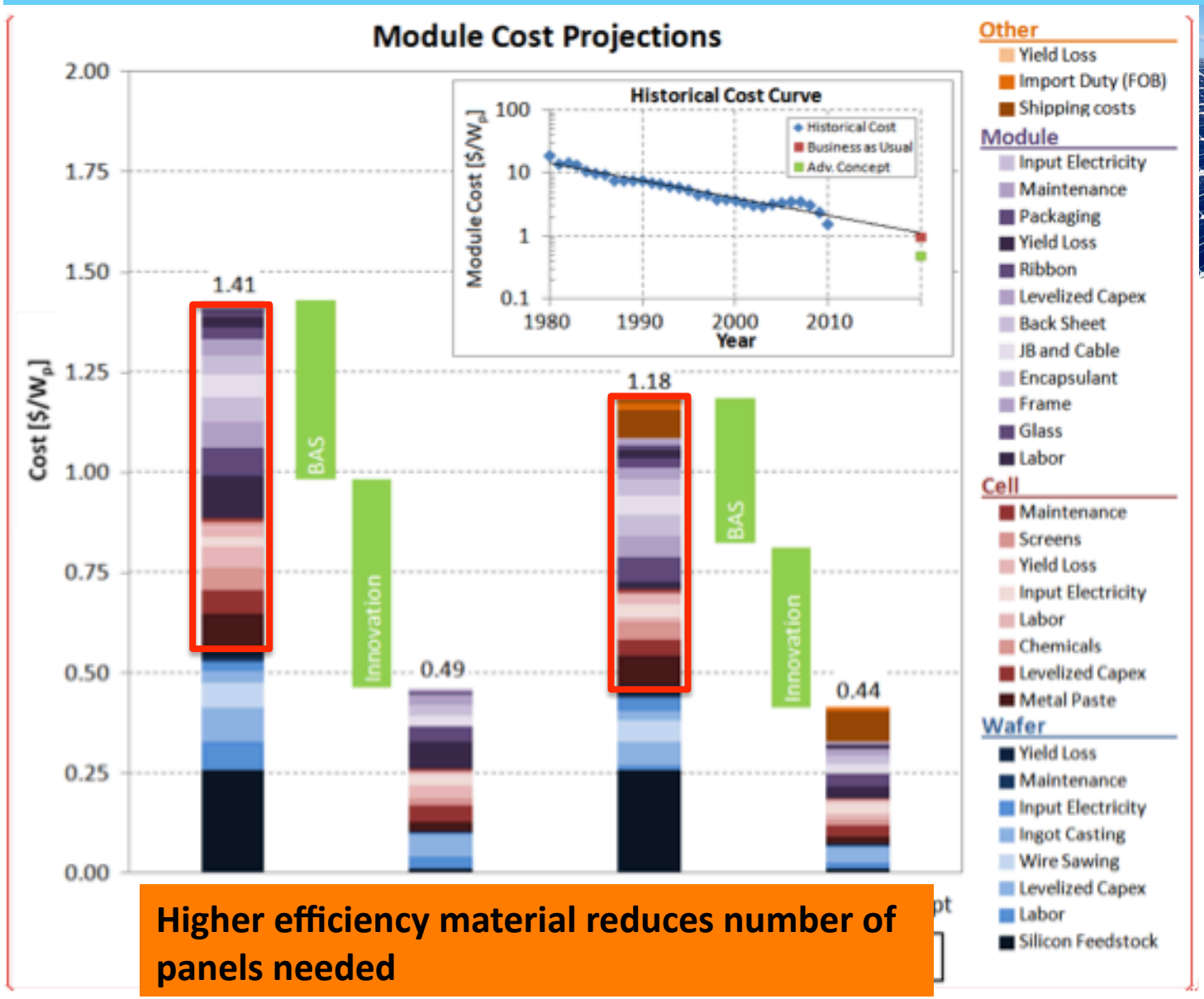
**Fuel Cell Vehicles**


**Intrinsic materials problems**

- Lack of stability of Pt catalysts in acid environments
- Hydrogen Storage

# Photovoltaics




**Higher efficiency material reduces number of panels needed**

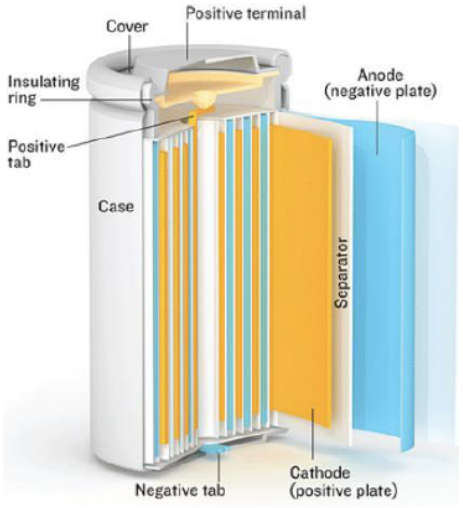
Efficiency of PV material is key as **most of the cost is in making panels**

# Lithium-ion batteries for electric vehicles

1989

2012


*"No, no, no, no, this sucker's electrical, but I need a nuclear reaction to generate the 1.21 gigawatts of electricity ! "*




**70%-80% of cell cost is materials**

**Higher energy density materials reduces cost per kWh of stored energy**

# Materials Play a Strategic Role Today

**Sept 7, 2010**

Japan arrest Chinese boat captain

**Sept 22, 2010**

China blocks shipments of **Rare Earth Metals** to Japan

**Sept 24, 2010**


Japan releases captain

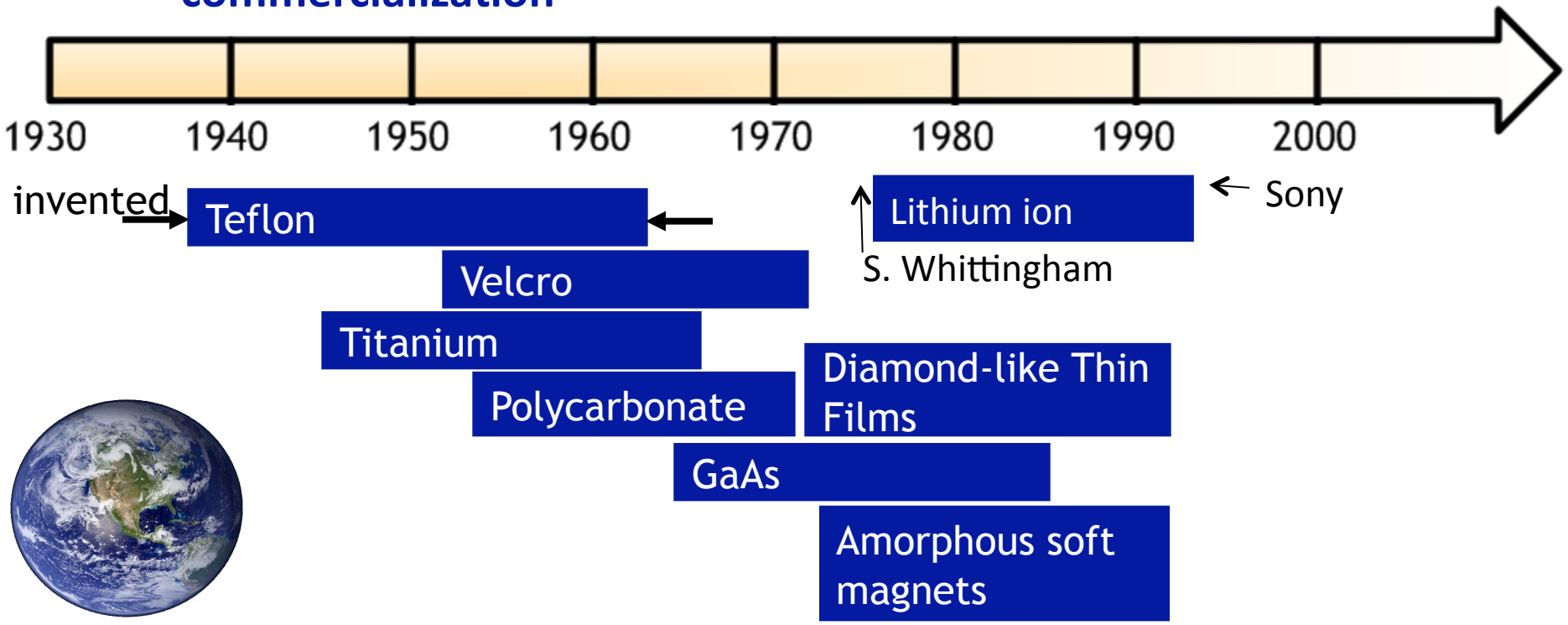
**Japan invests in induction motors….**
**"Toyota Readying Motors That Don't Use Rare Earths…".** *Jan 14, 2011 1:50 PM PT*

# Traditional Materials Discovery Timeline

Solar | Hydrogen | Permanent magnets | Construction (cement, steel, ..)

Thermoelectrics | Energy Storage | …

**18 Years…from the average new materials discovery to commercialization**

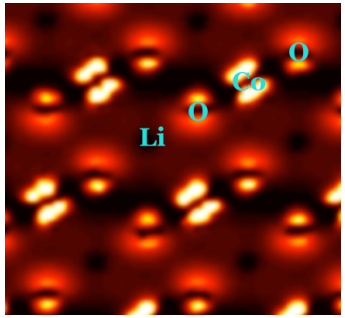


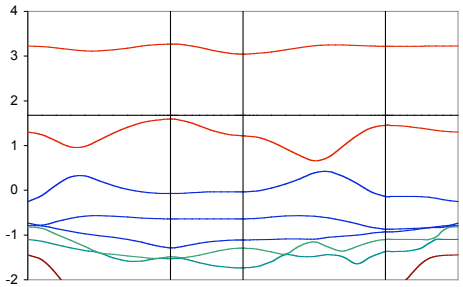



Materials Data from: Eagar, T.; King, M. Technology Review (00401692) **1995**, 98, 42.

# How to compute real world materials properties?

## Quantum Mechanics




"E = 325.67 kJ"

Optimize

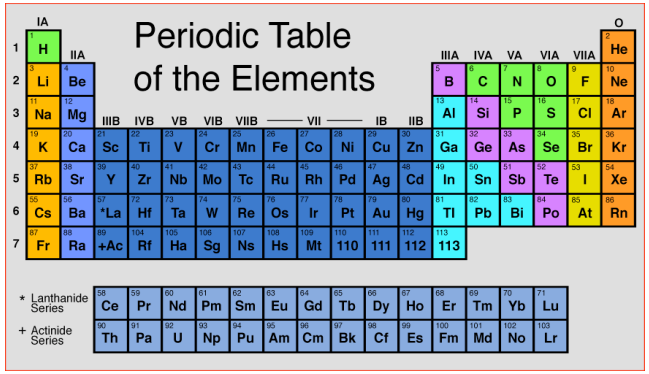
$$\langle \Psi | \text{better battery} | \Psi^* \rangle$$

## Engineering Properties

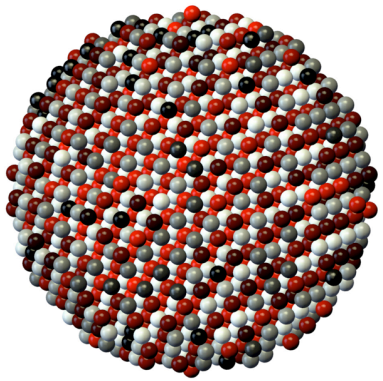
Corrosion, strength, energy density, …

# Computational Materials Science and First-Principles Calculations

# Aim of *ab initio* calculations




$$\hat{T}\Psi_{\mathrm{MB}} + \hat{V}\Psi_{\mathrm{MB}} = -i\frac{d\Psi_{\mathrm{MB}}}{dt},$$



**Atomic Numbers**

↓

**Solve quantum mechanics for the material**

↓

**Predict physical and chemical properties of systems**

# Standard DFT – steady state

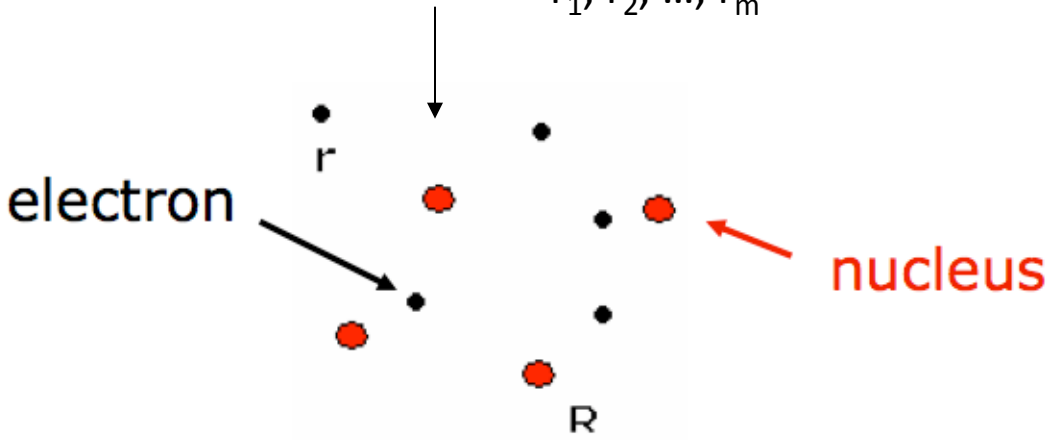
$$\hat{H}\underline{\Psi} = E\underline{\Psi}$$

As you can see, quantum mechanics is “simply” an eigenvalue problem

# Summary of problem to solve

Assume that the nuclei (Mass $M_i$) are at:
$R_1, R_2, \ldots, R_N$

Assume that the electrons (mass $m_e$) are at:
$r_1, r_2, \ldots, r_m$




$$\hat{H}_{N,e}\Psi_{N,e}\left(\{R_I\},\{r_i\}\right) = E_{N,e}\Psi_{N,e}\left(\{R_I\},\{r_i\}\right)$$

where $\hat{H}_{N,e} = \underbrace{\hat{T}_N + \hat{T}_e}_{\text{kinetic energy}} + \hat{V}_{N-N} + \hat{V}_{N-e} + \hat{V}_{e-e}$

**nucleus – nucleus interaction**

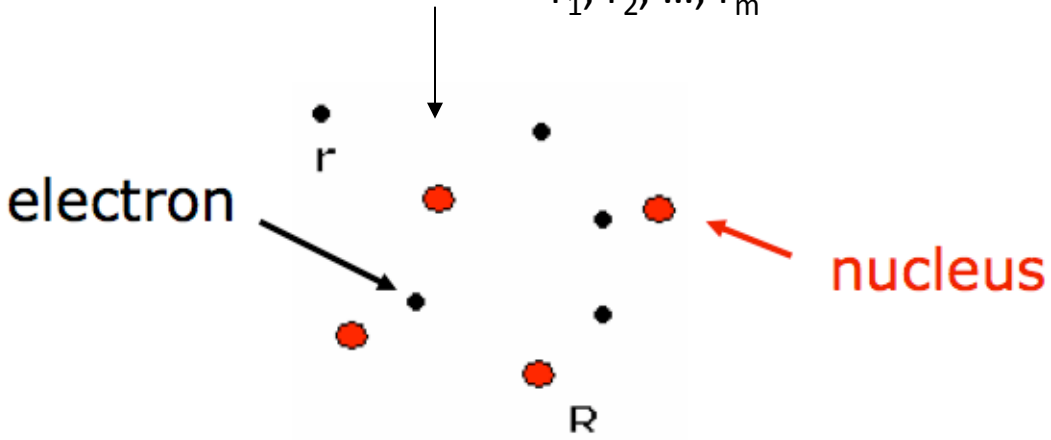
**interaction between electrons**

**nucleus - electron interaction**

# Summary of problem to solve

Assume that the nuclei (Mass $M_i$) are at:
$R_1, R_2, \ldots, R_N$

Assume that the electrons (mass $m_e$) are at:
$r_1, r_2, \ldots, r_m$




$$\hat{H}_{N,e}\Psi_{N,e}\left(\{R_I\},\{r_i\}\right) = E_{N,e}\Psi_{N,e}\left(\{R_I\},\{r_i\}\right)$$

where

$$\hat{H}_{N,e} = \underbrace{\hat{T}_N + \hat{T}_e}_{\text{kinetic energy}} + \hat{V}_{N-N} + \hat{V}_{N-e} + \hat{V}_{e-e}$$

**nucleus – nucleus interaction** (separable)

**nucleus - electron interaction**

Non-separable **interaction between electrons**

# Electrons are difficult!

- The mathematical difficulty of solving the Schrodinger equation increases rapidly with N
- The number of computations scales as $e^N$
- With modern supercomputers we can solve this directly for a very small number of electrons (maybe 4 or 5 electrons)

*Materials contain of the order of $10^{26}$ electrons*

# Quantum power: *Density Functional Theory*

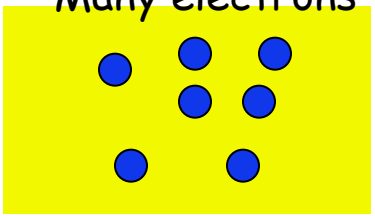
Kinetic Energy

Interaction with nucleus

Interaction between electrons

$$H = \sum_{i=1}^{N_e} \nabla_i^2 + \sum_{i=1}^{N_e} V_{nuclear}(r_i) + \frac{1}{2} \sum_{i}^{N_e} \sum_{j \neq i}^{N_e} \frac{1}{|r_j - r_i|}$$

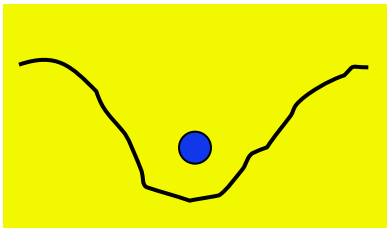
Many electrons



Replace e-e interaction by average potential

$$H = \sum_{i=1}^{N_e} \nabla_i^2 + \sum_{i=1}^{N_e} V_{nuclear}(r_i) + \sum_{i=1}^{N_e} V_{effective}(r_i)$$

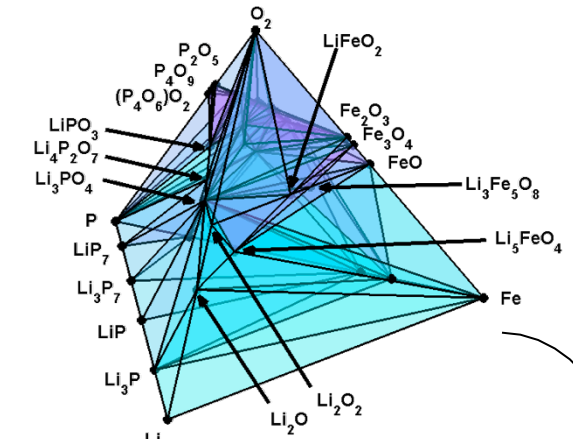
One electron

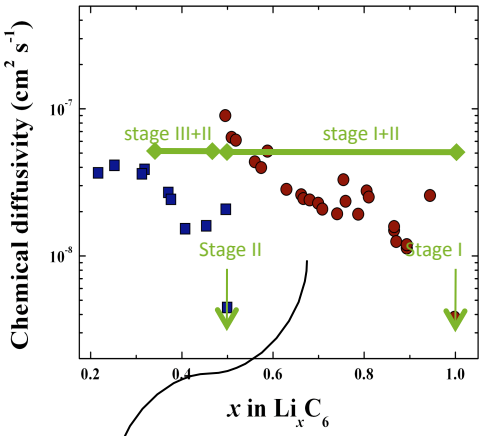


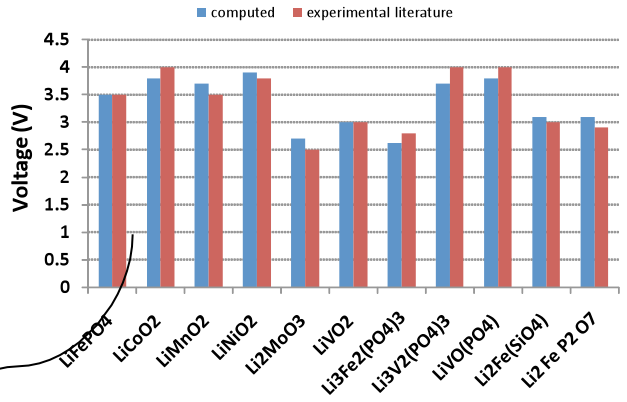
Approximated in the Local Density Approximation (LDA) or Generalized Gradient Approximation (GGA) to Density Functional Theory (DFT)

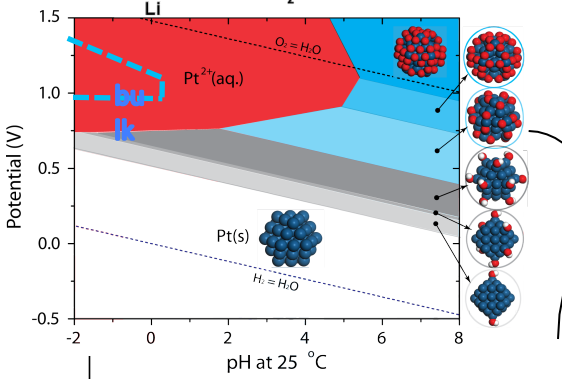
$V_{eff}$ = average electrostatic potential from other electrons + exchange effect (Pauli principle) + correlation effects

# Many properties can be computed

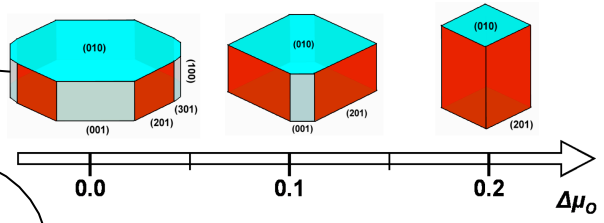


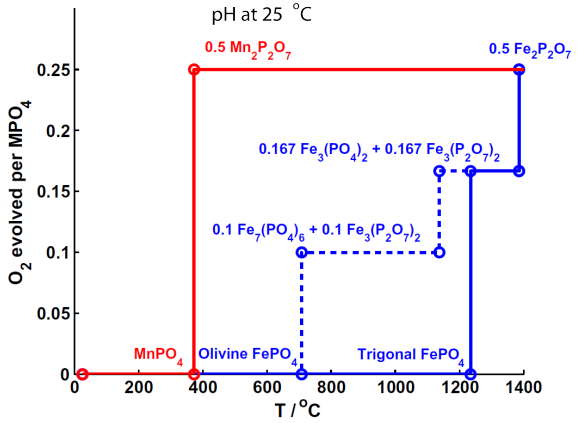



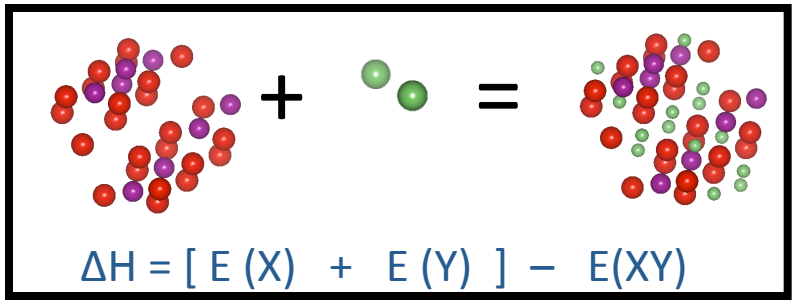



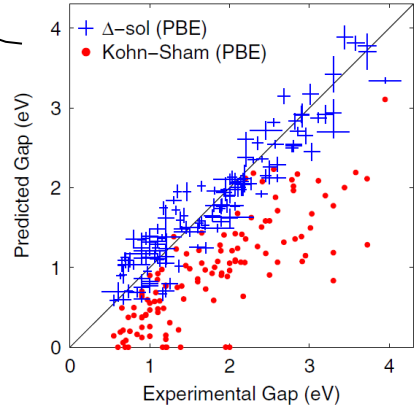




Photovoltaics, Thermoelectrics,
Energy Storage, Hydrogen,
Catalysts, $CO_2$ capture….

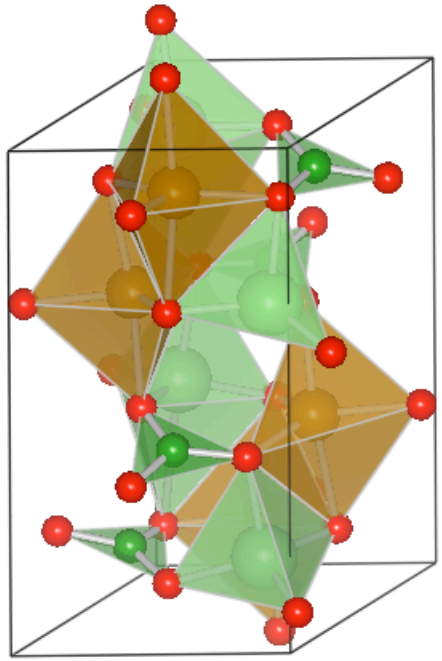


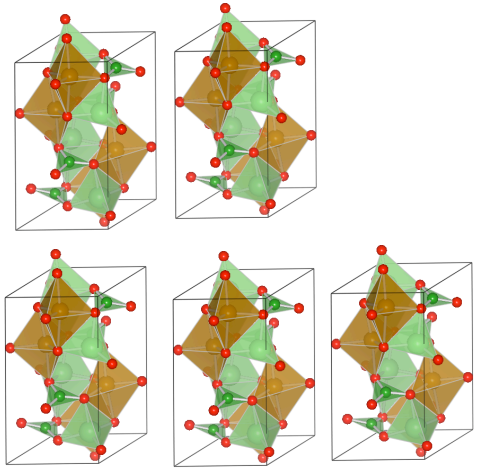
# Computations are scalable (or are they?)

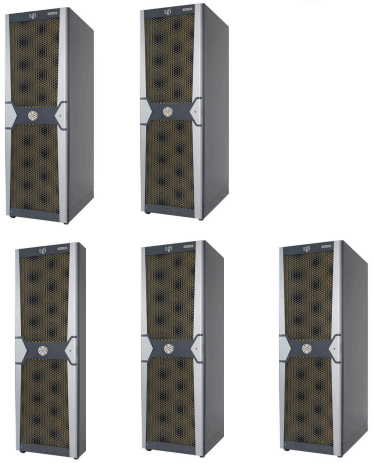


+

+

$$i\hbar \frac{d\Psi(\{r_i\};t)}{dt} = \hat{H}\,\Psi(\{r_i\};t)$$





**Total energy**
**Optimized structure**
Magnetic ground state
Charge density
Band structure / DOS
… etc

# High Throughput Scientific Computing

# Bummer – 'exascale' not working for DFT… ☹

DFT codes are trivially parallelizable over k-points

BUT after every reciprocal k point calculation – all the energies (information) have to be assembled to calculate charge density and total energy… happens hundreds of times per calculation.

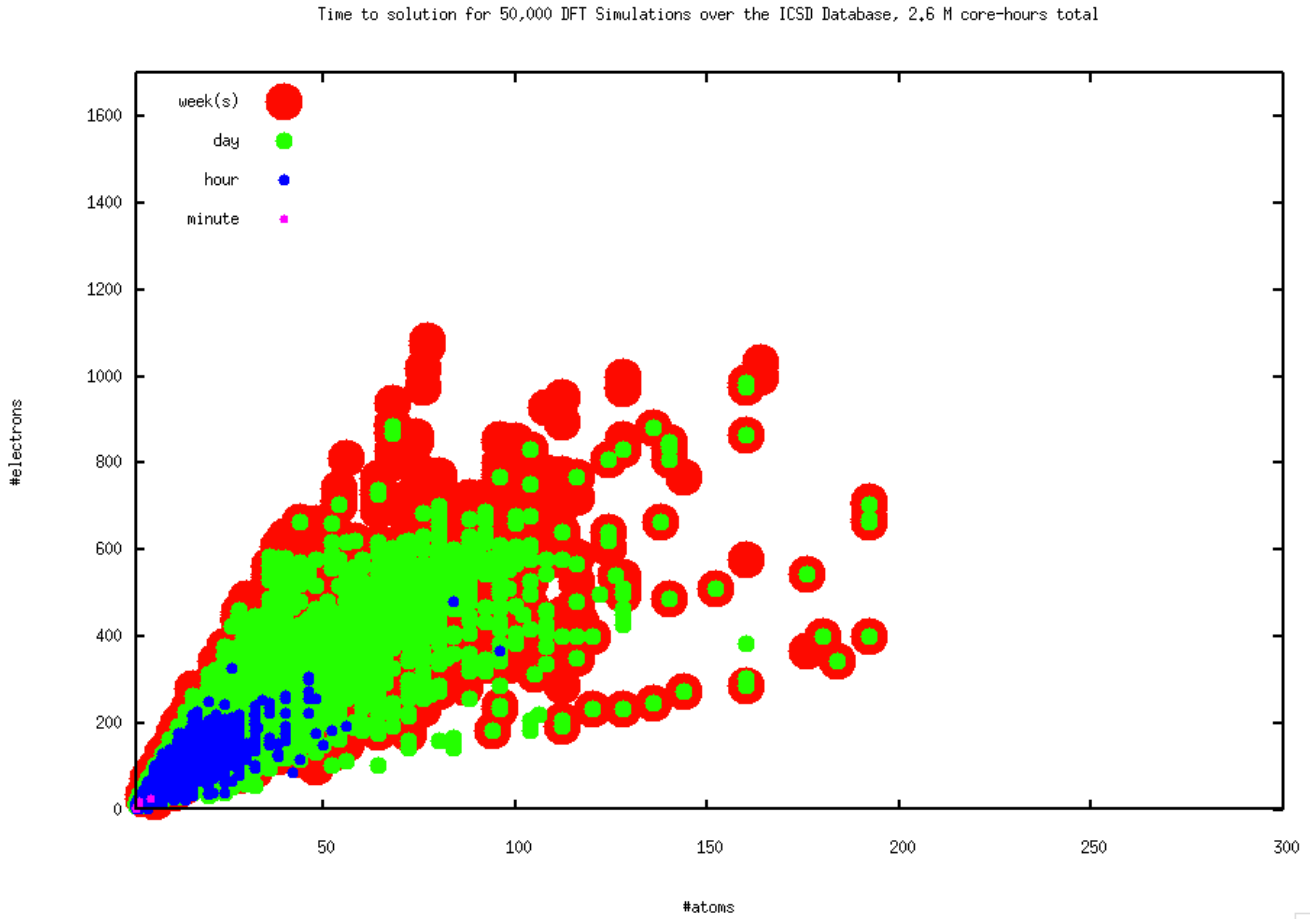
too much communication between nodes!

More sophisticated parallellization schemes exist, but fact remains – **no DFT code scales better than 30-40 nodes**

**So what do we do? We run one material per node…no intercommunication needed and large # nodes can be requested**

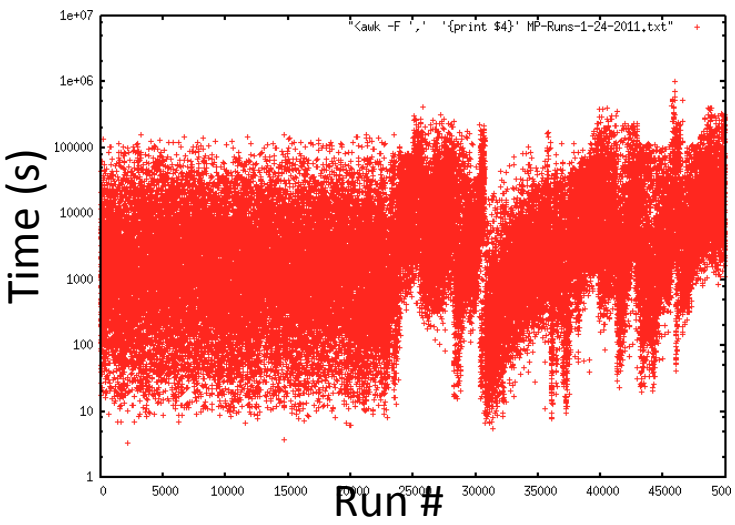
# HTC Principles

- Our computations have rather unpredictable runtimes
- Think swarms of workers – one is very slow, another one terrible efficient…

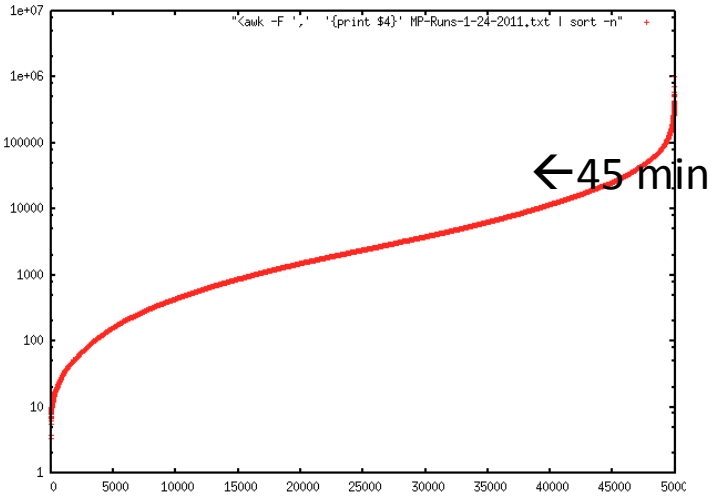


# Unpredictable runtimes

- 50K DFT runs lasting from 10s to 10 days




sort →




- Must ask for ~upper bound walltime in batch script → terrible for the unfinished jobs
- Re-starting…

# High-Throughput Computing

- Requires scheduler or run-time tools to facilitate running large numbers of jobs with variable duration time
- Inverse of typical large-scale simulations of inter-connected tasks (climate, astronomy, …)

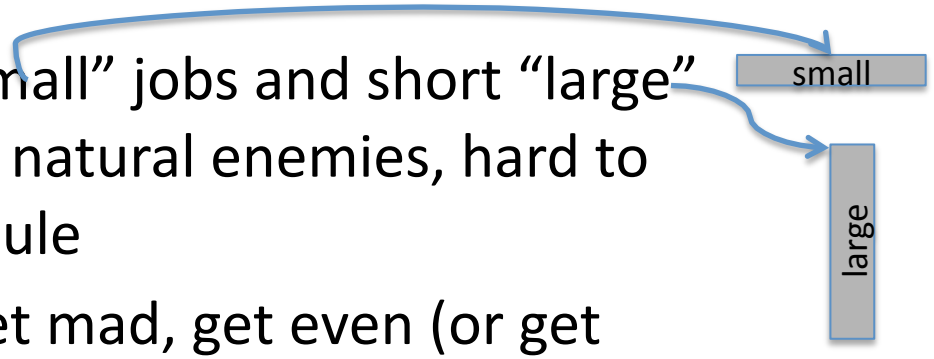
# HTC Principles

**HTC comes down to two concepts:**

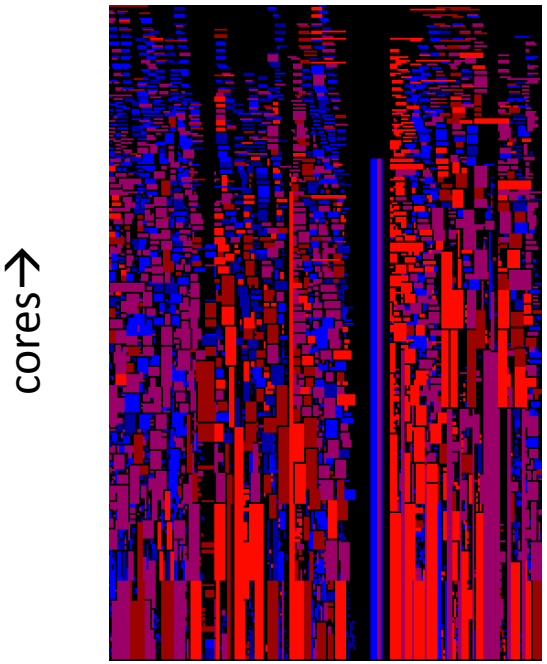
- Concurrency: how many cores?
  - Assuming tasks are independent then core-hours needed for N jobs is just $T = \Sigma_i t_i$
  - What if T is _really_ big? → need HPC
- Policy : for how long and in what groupings?
  - Local resource : do what you want within concurrency limits, require T be small
  - Shared resource : map workflow to policy
    - N _really_ big may present issues

# Shared Resources: Some Challenges are Unavoidable

- Long “small” jobs and short “large” jobs are natural enemies, hard to coschedule
- Don’t get mad, get even (or get things done)
- Get the throughput you want
- Read the queue policies




Real job data from NERSC

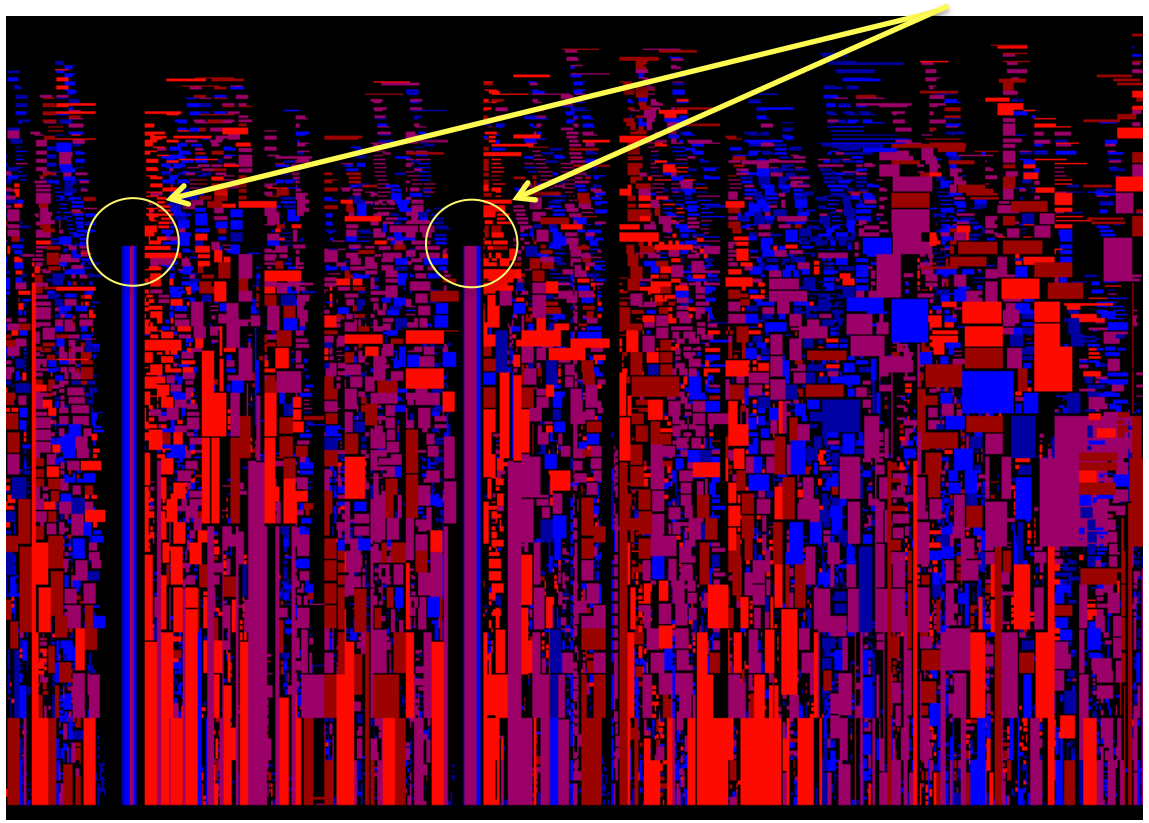



long queue wait

short queue wait

# Queueing: long+small vs. short+tall

On a shared machine, big jobs “drain” the queue



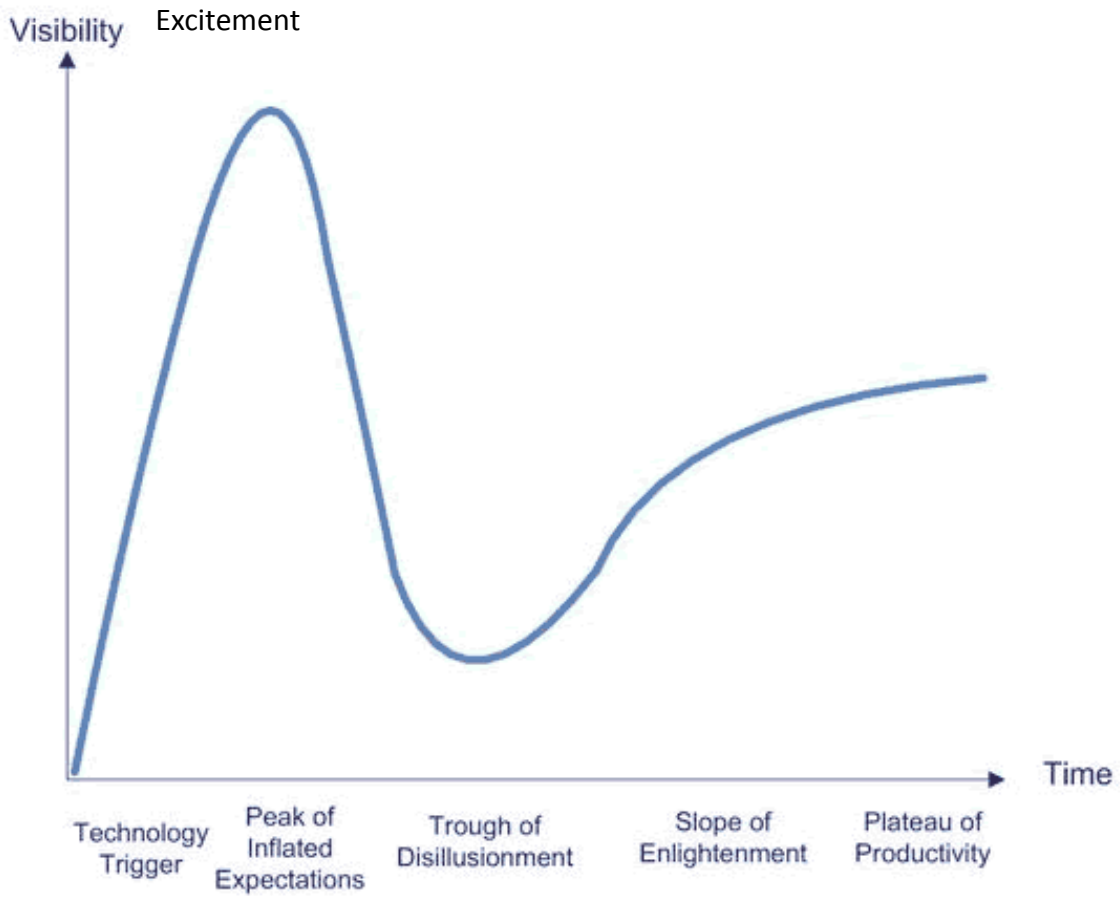

long queue wait

short queue wait

# HTC on HPC: Modern Day

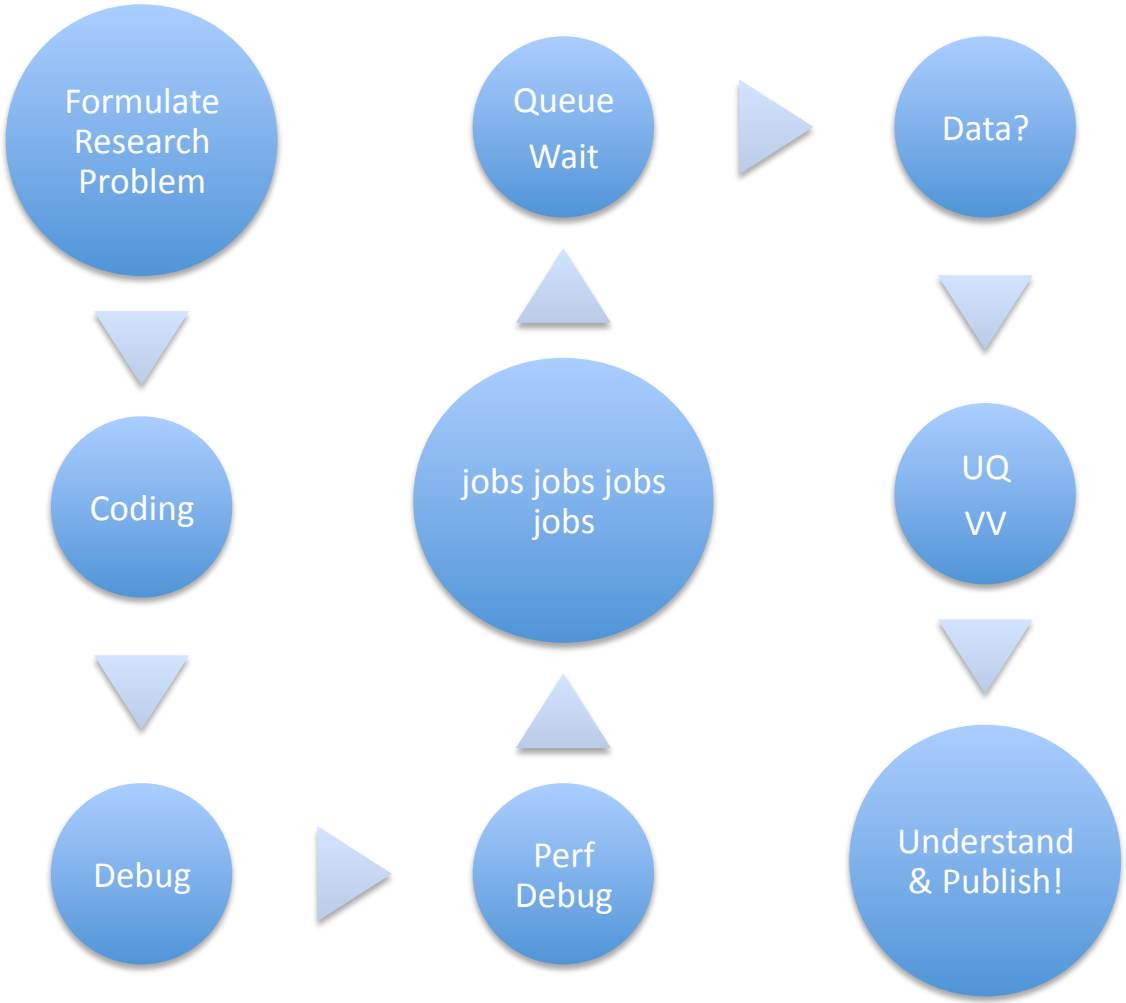
- If you need to run lots of jobs and can do that with good performance then do what works.
- If the number of jobs does not match with the job scheduling policies then you may consider consolidation.
- Optimization often requires consolidating tasks,or jobs to act in synchrony.

# So you have access to 150K cores, now what?




Gartner Hype Cycle : graphic representation of the maturity and adoption of technologies and applications

# Performance is more than a single number




- Plan where to put effort
- Optimization in one area can de-optimize another
- Timings come from timers and also from your calendar, time spent coding
- Sometimes a slower algorithm is simpler to verify correctness

# A few notes on queue optimization

Consider your schedule

- Charge factor
  - regular vs. low
- Scavenger queues
- Xfer queues
  - Downshift concurrency

Consider the queue constraints

- Run limit
- Queue limit
- Wall limit
  - Soft (can you checkpoint?)