

Before we complain...

Congratulations on outstanding and
unprecedented user satisfaction!

Stephen Bailey, Anubhav Jain, Paul
Kent, Adrienne Middleton

NUG Queue committee notes

Stephen Bailey, Anubhav Jain, Paul
Kent, Adrienne Middleton

NUG Queue committee notes



"The Criminal"

"The Athlete"

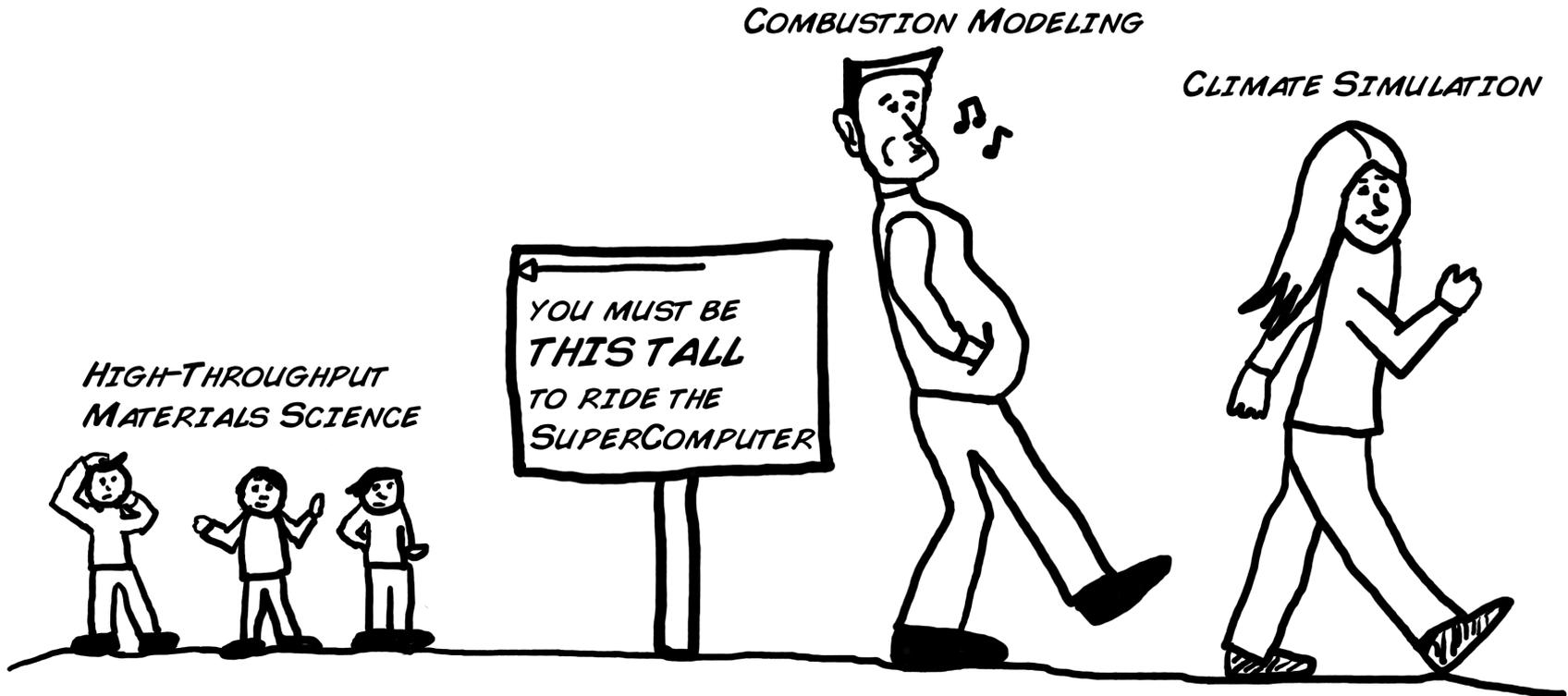
"The Basketcase"

"The Princess"

"The Brain"

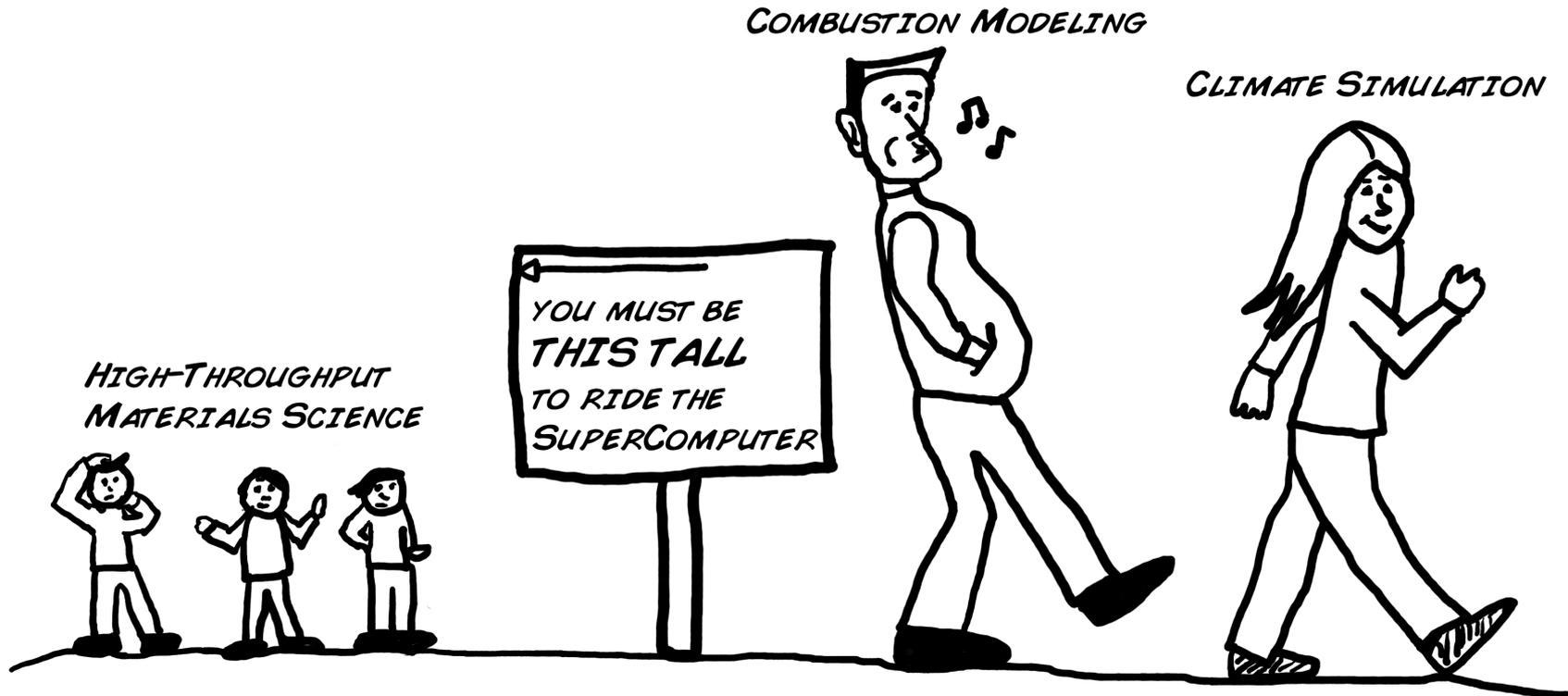
The Breakfast Club

(Francesca's request) -
Should big computers run small calculations?



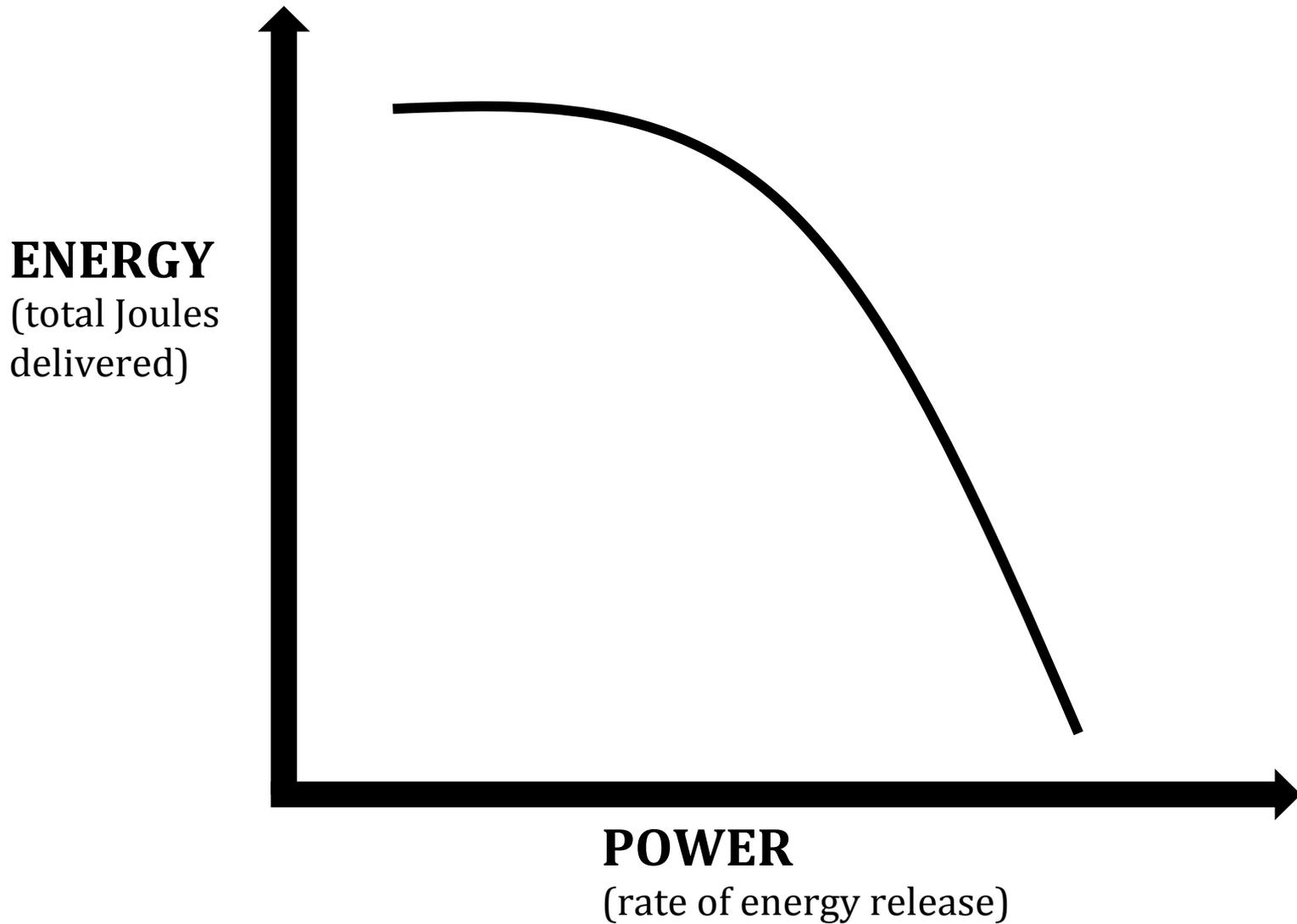
- (a) No, because only massively parallel simulations require large computers
- (b) No, because massively parallel jobs are more important than smaller jobs
- (c) Yes - but small jobs should be second class citizens
- (d) Yes, and small jobs should have equal rights!

(Francesca's request) -
Should big computers run small calculations?

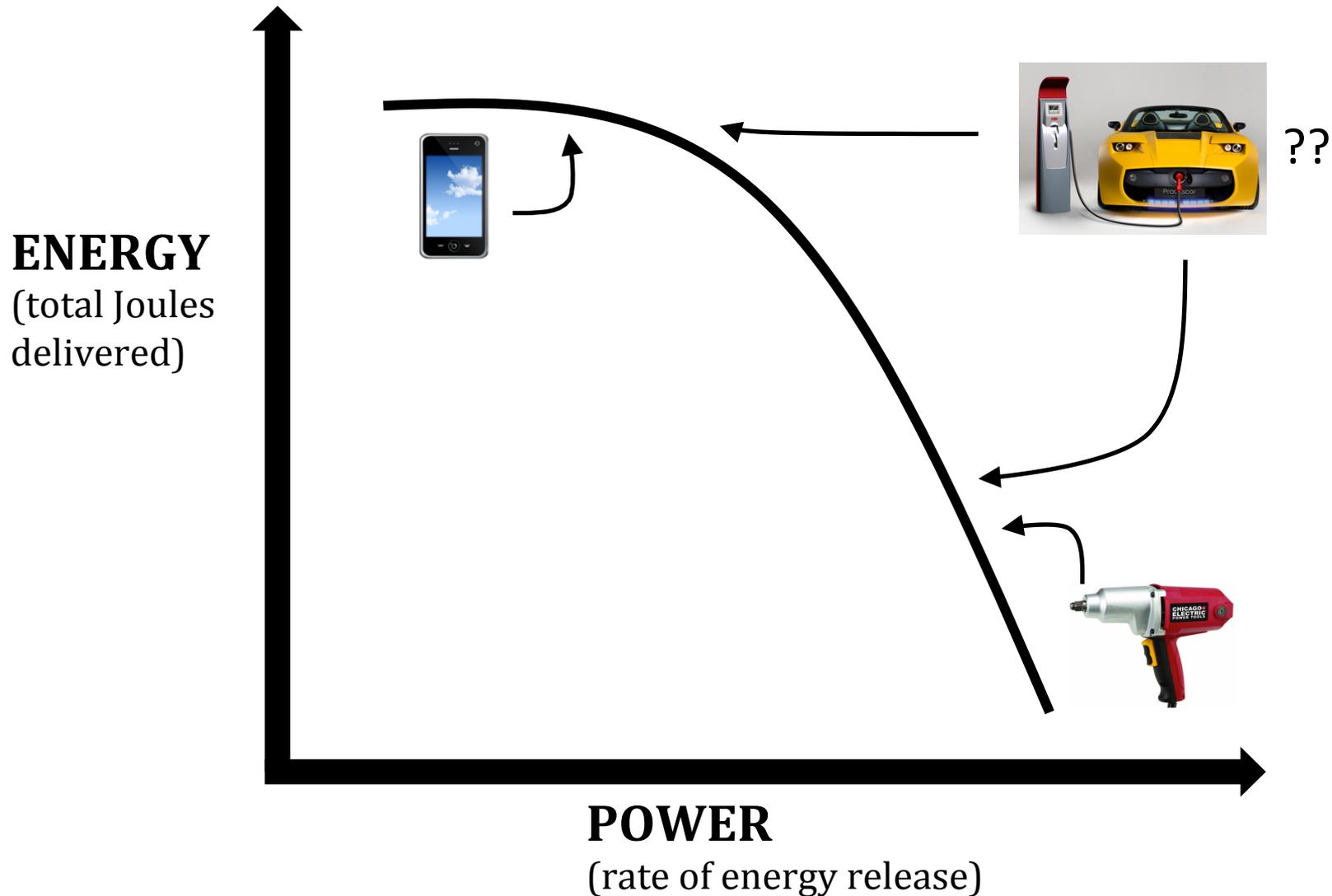


- (a) No, because only massively parallel simulations require large computers
- (b) No, because massively parallel jobs are more important than smaller jobs
- (c) Yes - but small jobs should be second class citizens
- (d) Yes, and small jobs should have equal rights!
- (e) No, we don't have enough capacity for them!**

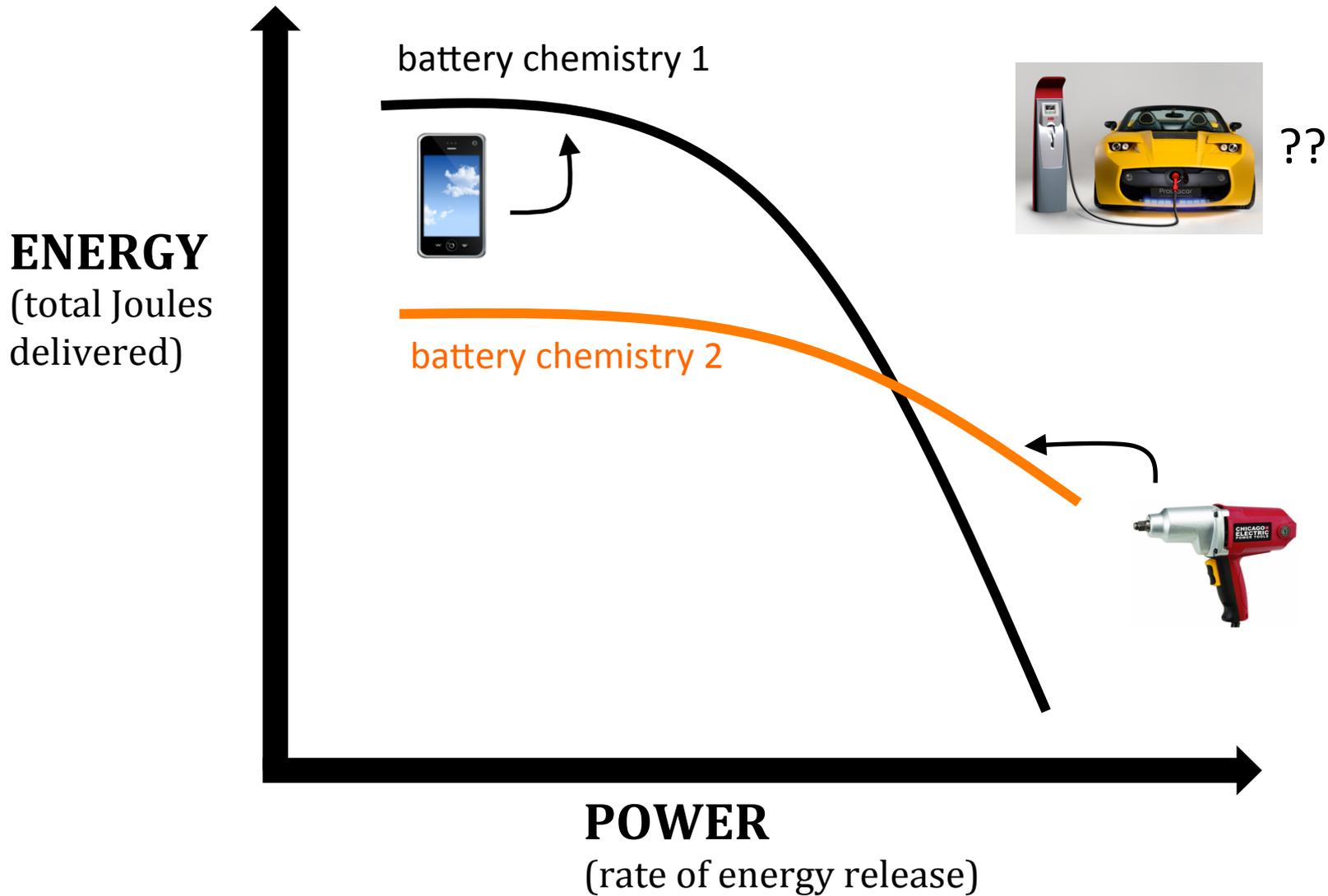
In batteries, a Ragone plot shows the balance between energy & power



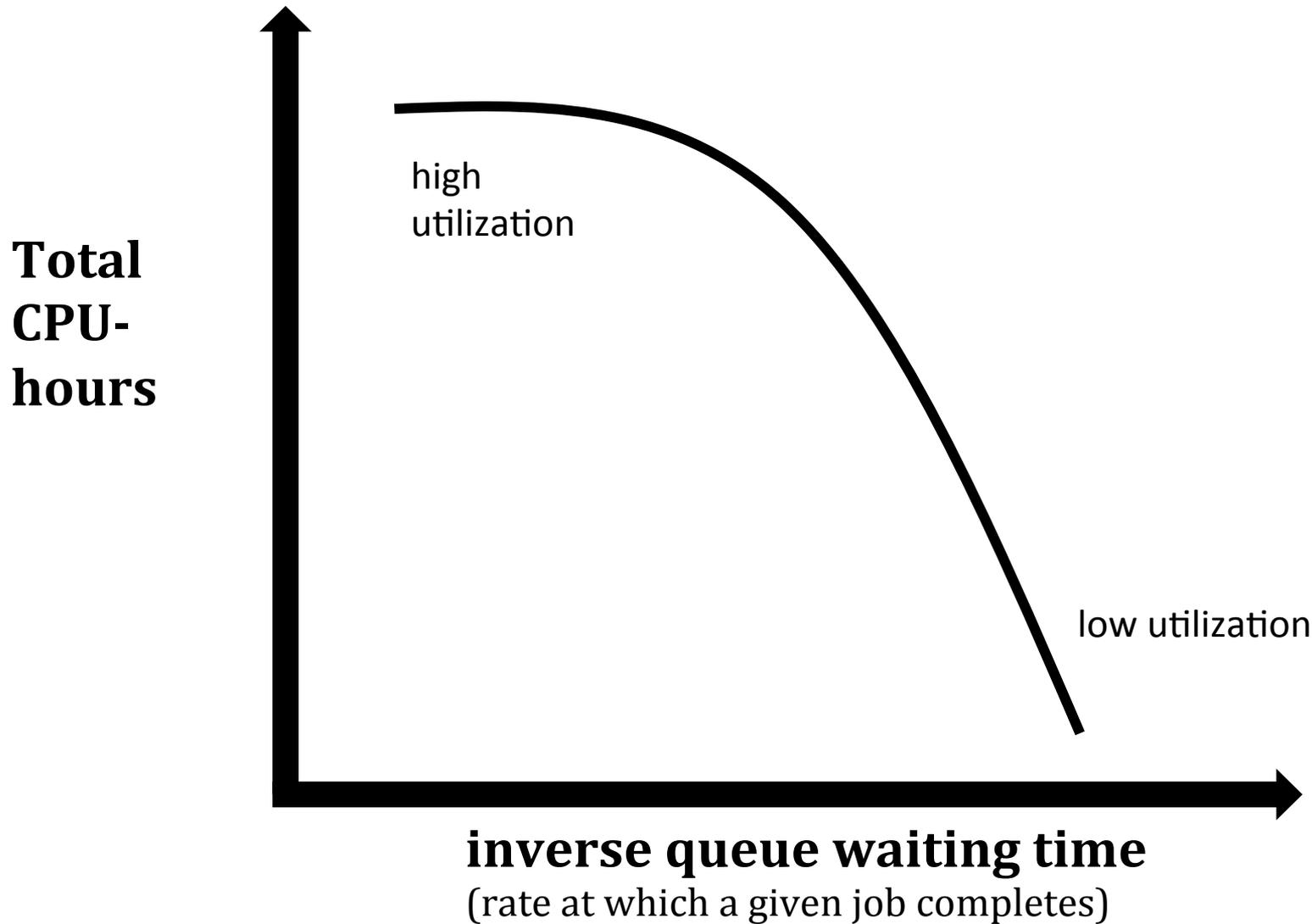
The usable energy depends on application's power requirements



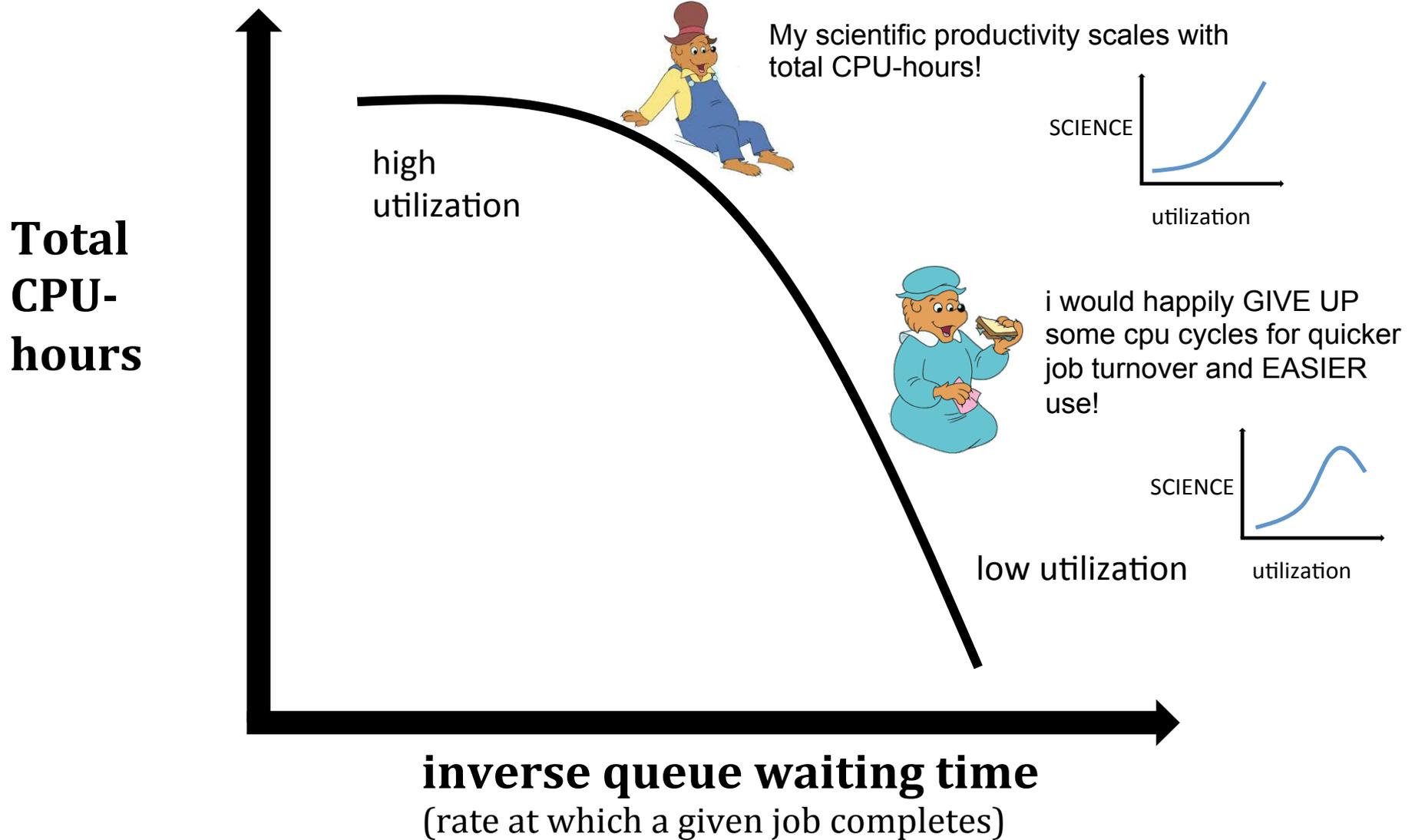
In batteries, different chemistries handle different use cases



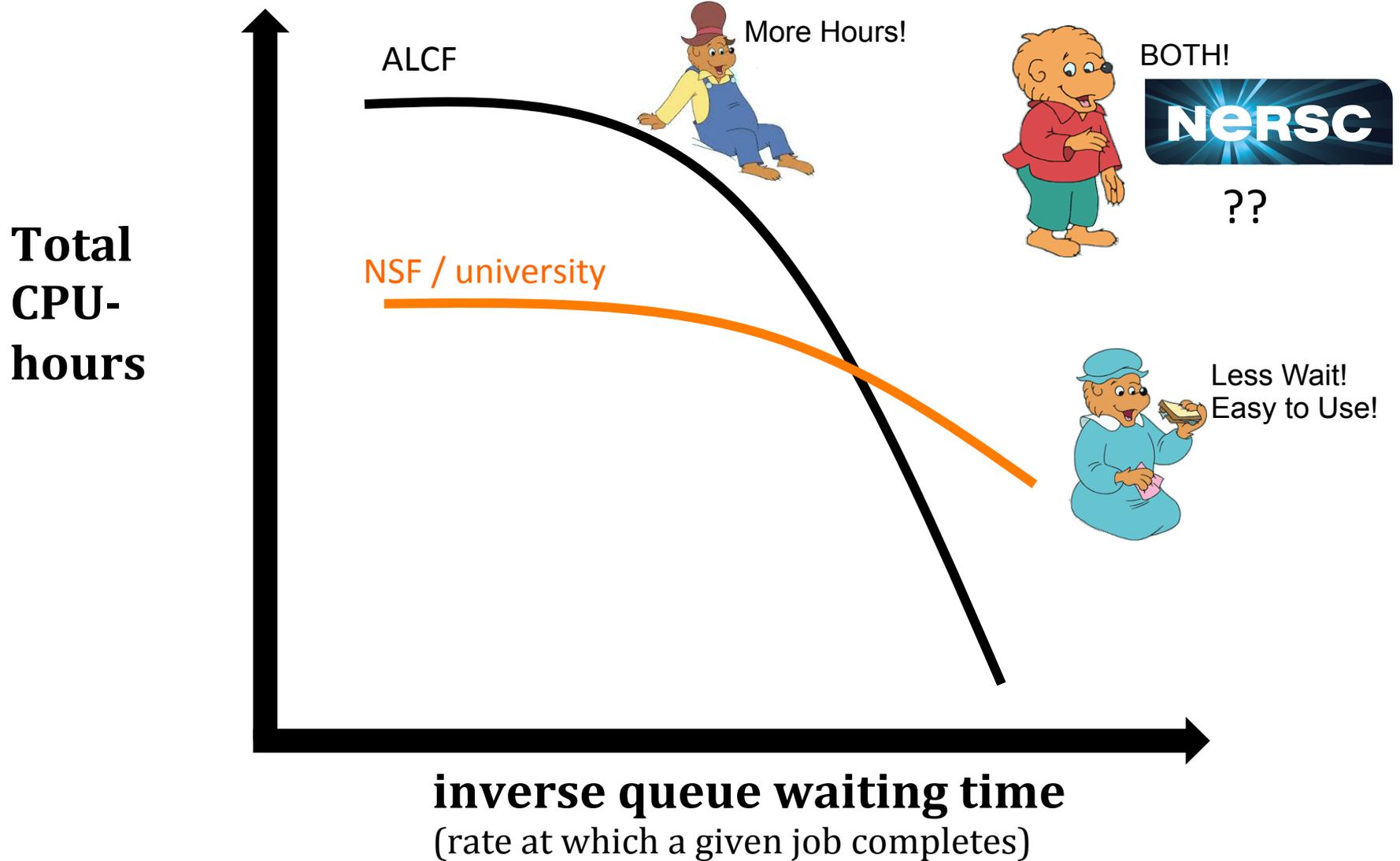
There is a similar situation with general-purpose supercomputing centers



As NERSC is aware, there are different situations here as well



And there are different types of supercomputing centers



Can NERSC ever make people happy as a mixed-use center?

- No

Trying harder...

Part 1: Better information

- Part of the problem comes from not knowing how long it will take for a job to run
 - **human stress:** “Will I have these results in time for group meeting? My big conference?”
 - **real inefficiency:** “I cannot know whether to try experiment A or B until this job finishes, and I won’t even know when the job will start. So we cannot schedule that meeting discussing the experiment yet... I’ll send another email when the job starts...”
 - **queues dilemma:** “There are 5 different strategies (queues, machines) I can use to get this piece of work done. Which is the best strategy?”

What do people want?

- They want “**showstart**”, but:
 - no one knows that it exists
 - those that do complain of its inaccuracy
- Although showstart can never be fully accurate, is there a possibility of doing better?
 - MLBase?
 - Summer project?
 - Official project?

What else do people want?

- What is the optimal way to submit a flexible job with regard to processors + walltime?
 - Something like “heatmap” + “showstart” combined
 - e.g., should I submit 3 jobs that are 6 hour walltime, or 1 job that is 18 hour walltime? What amount of processors?
- People would like a tool that can quickly tell them the best way to get a flexible unit of work accomplished
- In other words, how does job shape affect total throughput?

Trying harder:

Part 2 – winning a zero sum game

- Let's take the case where NERSC is adamant to maintain 99.9999% utilization
- With no extra cycles, job priority is a zero-sum game
 - Improving priority for one job must decrease priority for another
- Clearly there is no way to win - ***right?***



But we can exploit that people want their zero sum in different ways!

- Many projects have certain periods of time and certain jobs that require rapid turnover, and others that are not so urgent. NERSC knows this so there is...
- **Special queue**
 - 2X charge factor controversial way to burn hours
 - shape limits of special queue means it doesn't work for many requests
- **Boosting (every project's dirty little secret)**
 - arbitrary – depends on NERSC's whim. Do you have a good NERSC representative for your project?
 - slow – message NERSC, NERSC messages appropriate people, discussion, implementation.
 - wasteful – NERSC's time and energy in approving and implementing, often on a short time-scale requiring they drop other things

Is there a better way to boost?

- Paul Kent's priority economics
 - *every* queue has a “low” and “high” priority version
 - if you submit to “low”, you earn credits in exchange for increased waiting times
 - if you submit to “high”, you burn credits in exchange for rapid turnover
- Each project manages its own priority cycles without affecting the overall priority of others

How does “priority economics” compare?

- Compared to special queue:
 - No 2X charge factor
 - no “magic shape” of jobs that can use it
- Compared to boosting:
 - arbitrary and requires NERSC representative? NO
 - slow and requires NERSC deliberation for each case? NO
 - wasteful? NO, it manages itself after initial setup

...and boosts can still be used, but at least minimized...

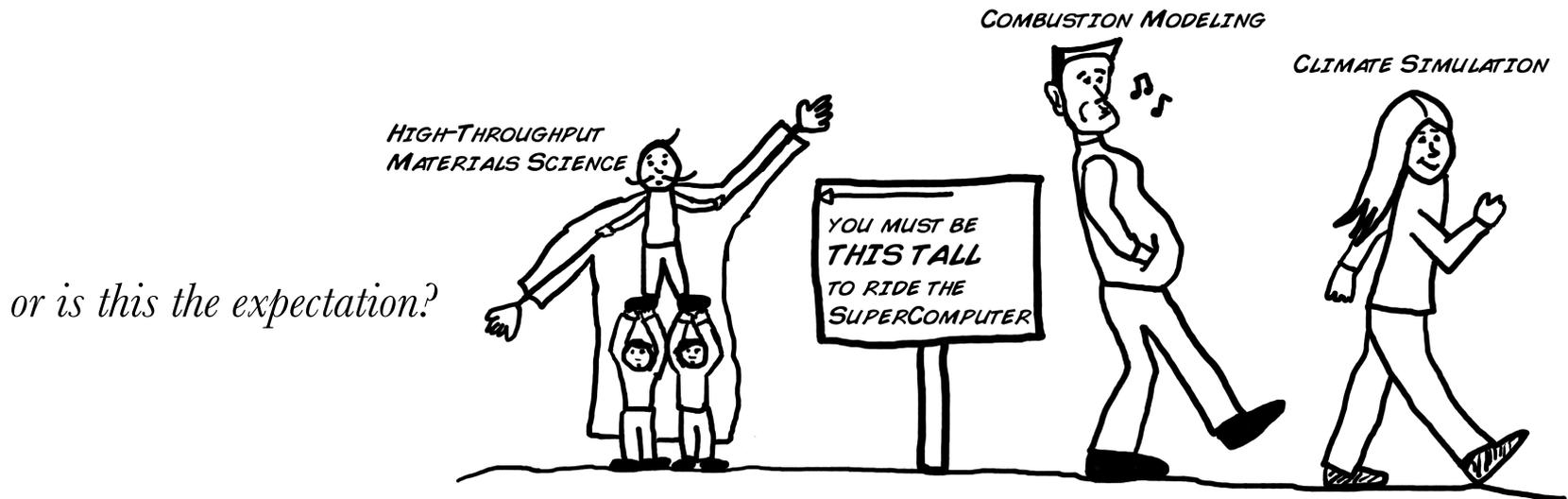
But need to make sure it works (i.e. low jobs don't get stuck forever without extra laws), and isn't another example of an overoptimistic economic model!

What if I think the the economic model is impossible/stupid/known to fail?

- OK
- But really NERSC should figure out a way to let people manage “priority rushes” for themselves. Otherwise
 - the most aggressive and well-represented projects are getting the most priority (just under the table)
 - NERSC’s human resources are spent deliberating and implementing queue priorities, whereas a good system should really be able to solve 95% of problems

Misc questions

- Why don't dependent jobs age?
 - for running workflows, or long-running jobs
- Is thruput queue on Edison under consideration?



if these topics are interesting to you...

“are these suggestions in the realm of reality?”

“are we discussing things NERSC has discussed before?”

Also, we need more *typical* NERSC users to join the committee!

