So what is this OCR thing?

Tim Mattson
Intel, Parallel Computing Lab

Disclaimer

- The views expressed in this talk are those of the speaker and not his employer.
- If I say something "smart" or worthwhile:
  - Credit goes to the many smart people I work with.
- If I say something stupid…
  - It's my own fault

I work in Intel's research labs. I don't build products. Instead, I get to poke into dark corners and think silly thoughts… just to make sure we don't miss any great ideas.

Hence, my views are by design far "off the roadmap".

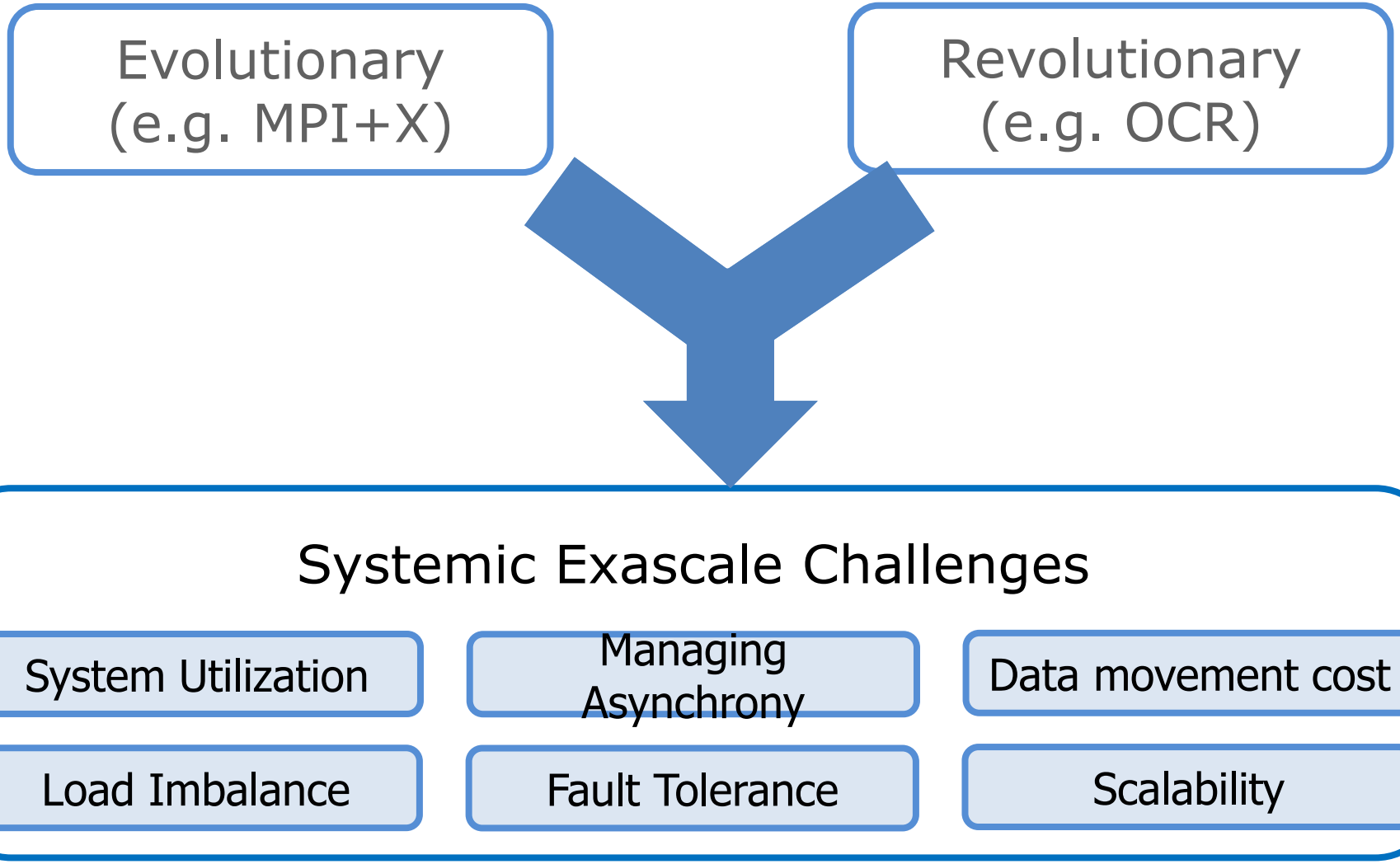
Here is what I'd like to talk about

- Please stop using acronyms and undefined terms in your talks.

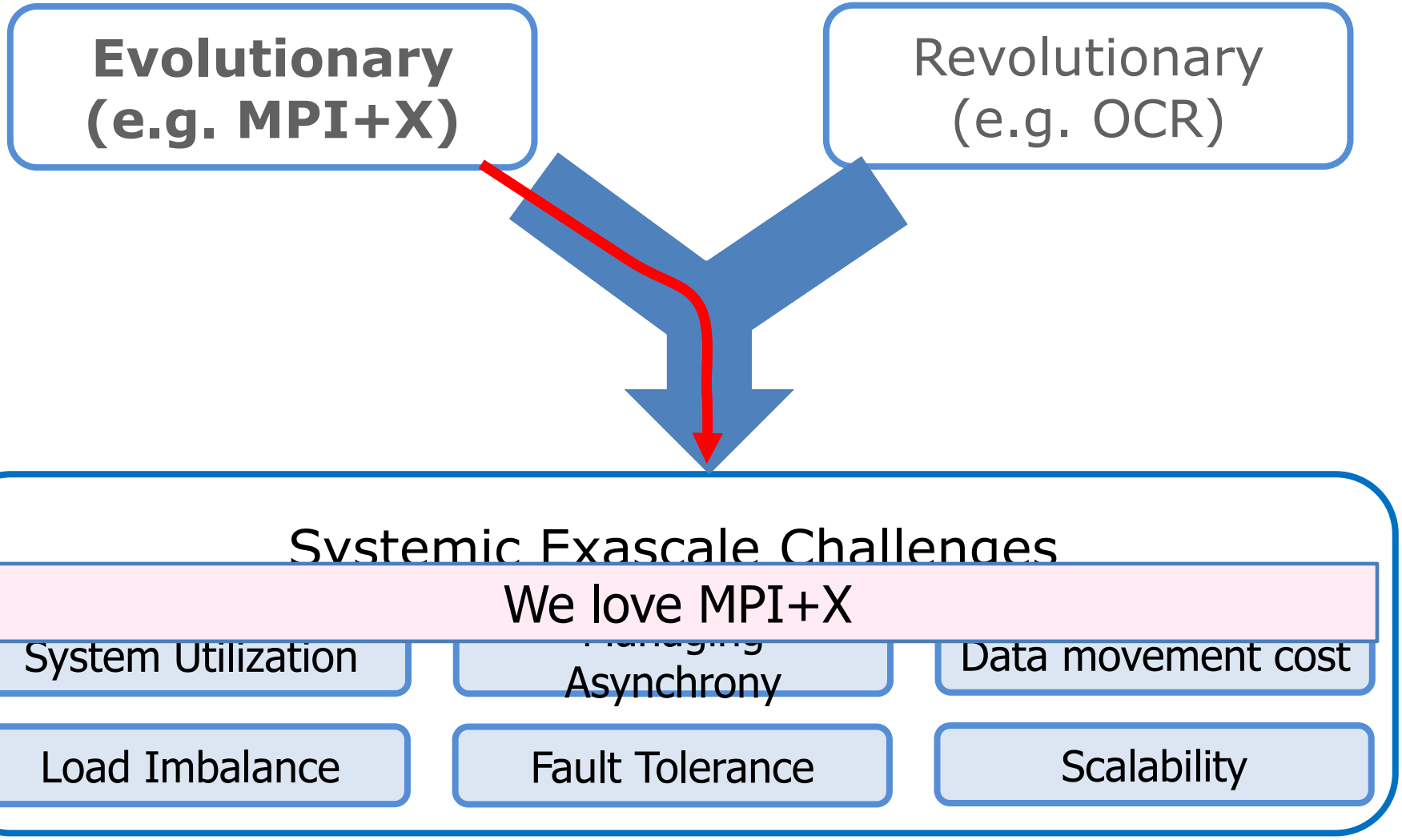
- Rice and Corn are commodities. CPUs and GPUs are NOT.
  - Or maybe I should turn the tables on you …. I should start referring to "commodity national labs" or claim that all universities are interchangeable and "your" department is fundamentally no different from others?

- Applications developers … whenever you use a programming model that is not standard, vendor-neutral, and available on a wide range of platforms…. You reward bad behavior and harm your own interests!!!
  - Standards only work if users demand them and support them

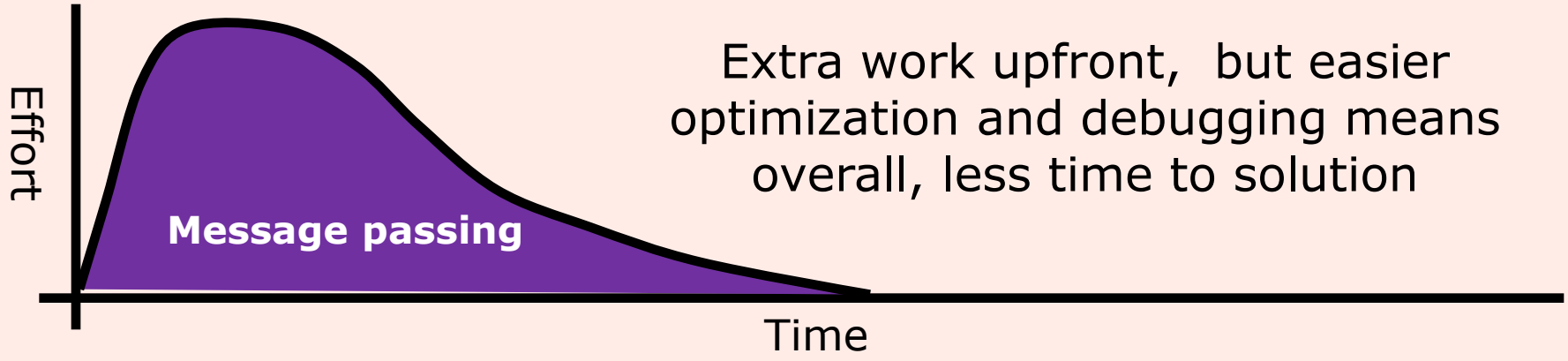
2 pathways to Exascale Runtime Research



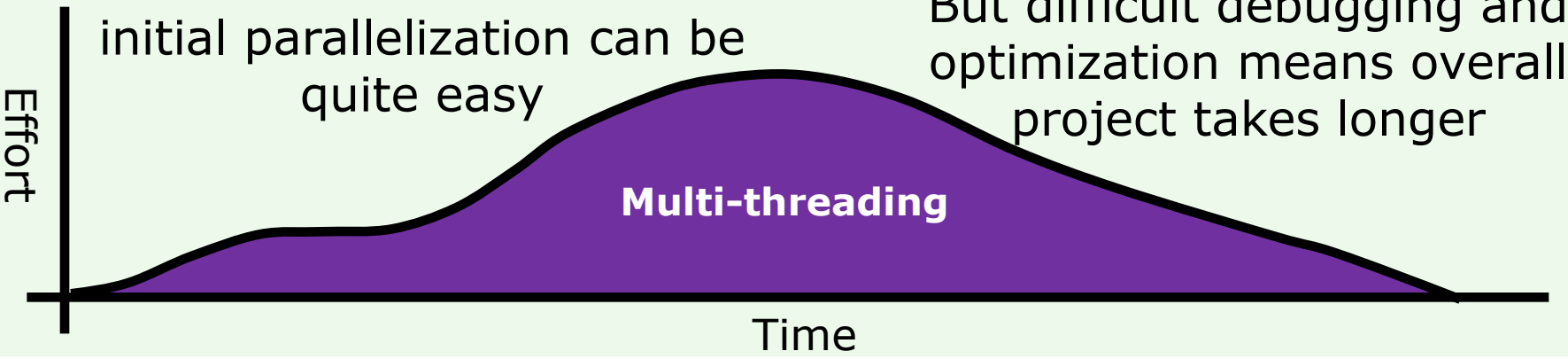
2 pathways to Exascale Runtime Research



I mean it … we really LOVE MPI



Extra work upfront, but easier optimization and debugging means overall, less time to solution



But difficult debugging and optimization means overall project takes longer

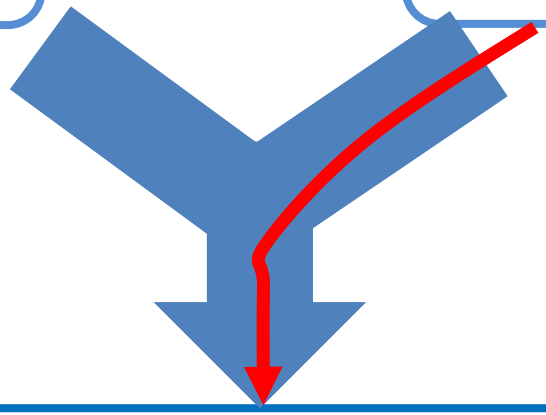
Proving that a shared address space program using semaphores is race free is an NP-complete problem*

*P. N. Klein, H. Lu, and R. H. B. Netzer, Detecting Race Conditions in Parallel Programs that Use Semaphores, Algorithmica, vol. 35 pp. 321–345, 2003

2 pathways to Exascale Runtime Research

Evolutionary
(e.g. MPI+X)

Revolutionary (e.g. OCR)



Systemic Exascale Challenges

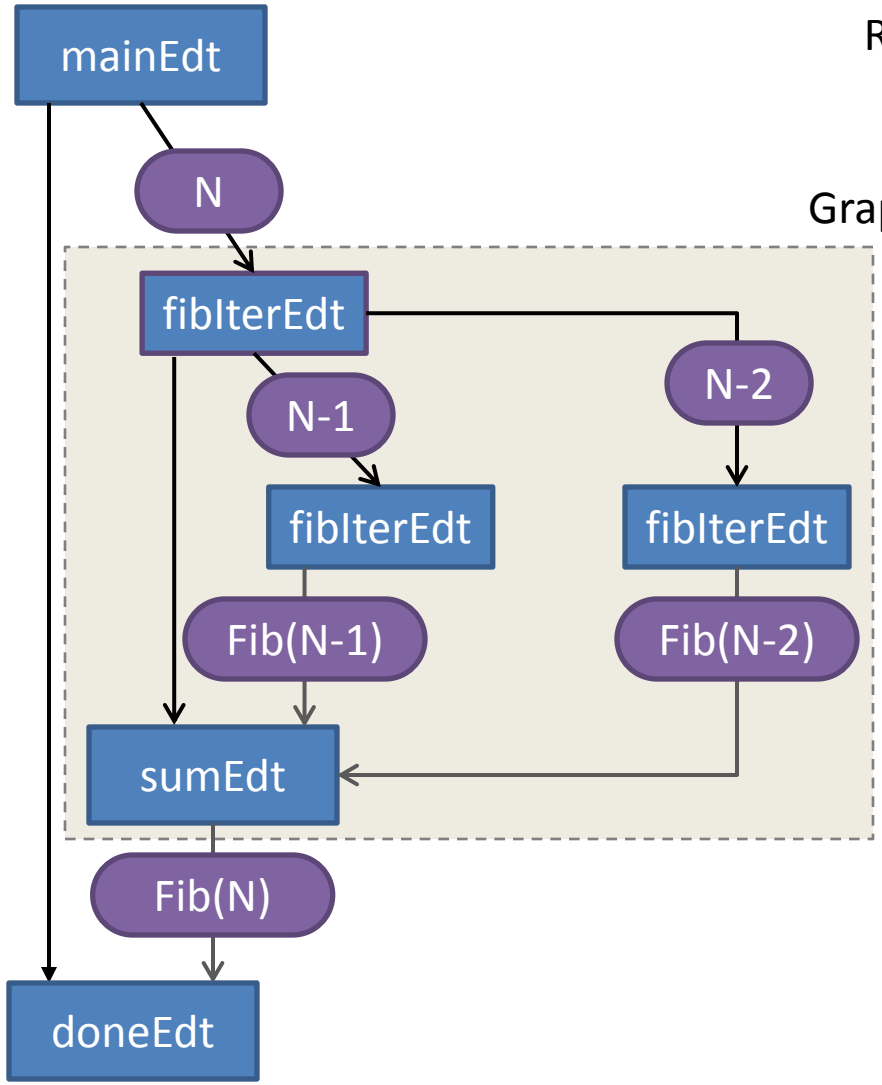
System Utilization	Managing Asynchrony	Data movement cost
Load Imbalance	Fault Tolerance	Scalability

OCR: A task-based runtime system for ExaScale research

- **O**pen **C**ommunity **R**untime
  - OCR is a research runtime to address the needs of next-generation HPC platforms
  - Developed collaboratively with multiple partners (mainly Rice University, Reservoir Labs and Intel)

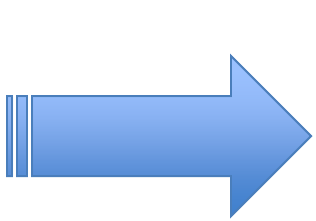
- Goal: Separation of concerns
  - Help programmer express the algorithm and its parallelism independent of the underlying platform
  - Optionally pass “hints” to minimize data movement and optimize performance/watt (work in progress)

Event Driven Tasking model



Program as a directed Acyclic Graph (DAG)

Runtime maps the constructed Directed Acyclic Graph onto architecture



EDT — A non-blocking unit of work. Runnable once all dependences are satisfied.

Datablock — Data shared between EDTs

Dependence: Source EDT satisfies one of destination's dependences

Conclusion

- OCR: Open Source release and community development model soon (early summer?).
  - Please join us and help create the “next MPI”.
  - Contact william.j.feirereisen@intel.com for more info
- CPUs and GPUs are not commodities
- Please stop killing kittens