

**TORSTEN HOEFLER**

DEPARTMENT OF COMPUTER SCIENCE, ETH ZÜRICH

# HPC vs. Irregular Applications

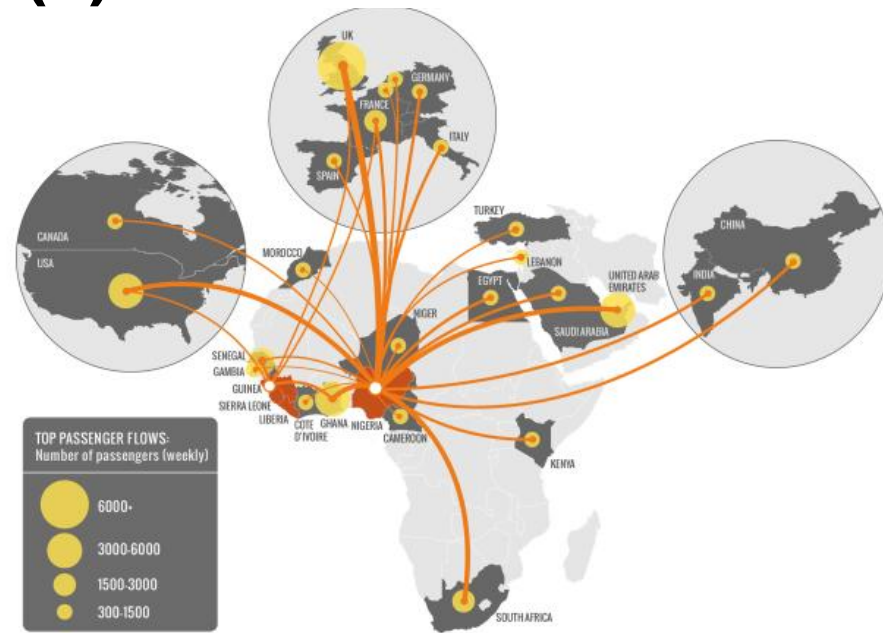


# Killer Irregular Application(s)?

- **Not sure if there is a killer yet ...**
  - Unless we talk about Ebola simulation(s)

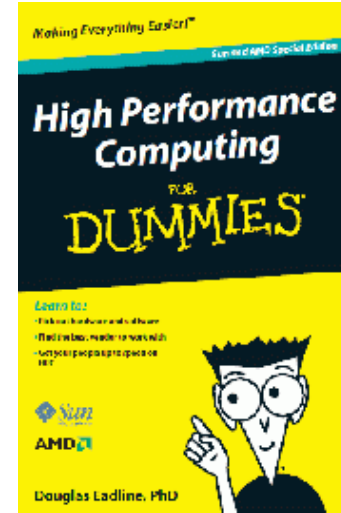
- **Graph databases in particular:**

- Shape search is hopeless for anything complex
- Complexity is double exponential  
[The Complexity of Evaluating Path Expressions in SPARQL]
- Is the analytics part of the database?  
*Or is it an in-memory graph computation?*



# Should we Make Programming Easy?

- Yes, always!
- Can we do it without loosing performance?
  - Tough question, may not be possible in the near future
  - We don't even quite all agree on the right abstraction:
    - Vertex-centric? [Pregel]*
    - Edge-centric? [GraphLab]*
    - Traversal-centric? (e.g., BFS) [PGBL]*
    - Building blocks? [Galois]*
    - Linear-algebra-centric? [Kepner et al., GraphBLAS]*
    - Communication-centric? [Active Pebbles]*
- How to make this easy?
  - Common denominator: data-centric\* (defer to Keshav)



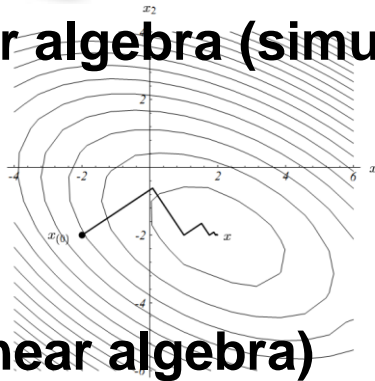
(\* Unless you're from Germany)

# Is it really only memory?

- It's always about balance [Kung'86]
- But what is the balance ... ranging from dense linear algebra



- through sparse linear algebra (simulating physical objects)



- to irregular graph (linear algebra)



# Accelerators – yeah or nay?

- Depends again on the balance

PCI EXPRESS®

- GPUs: fine until you have to use more than one!
- Xeon Phi: a bit more challenging (even on one)
- Graph-specific (Convey) – great! (but expensive)
- So for **large-scale assuming bad separators**: nay!
- There is hope: NVLINK



# Graph Processors? FPGAs?

- Why not? It's interesting!

- A new trend:

- Example: NoSQL 😊

**NO!**



Google goes back to the future with SQL F1 database

'Can you have a truly scalable database without going NoSQL? Yes!'

By Jack Clark, 30 Aug 2013



5,831 followers



- Non-von Neumann? Automata CPUs?

*So far only string matching examples (which also vectorize & parallelize very well)*

# Standardization? Library? Benchmarks?

## ■ Standard Interfaces

- Can we please agree on an abstraction first?
- cf. BLAS, MPI, ...

## ■ Is it GraphBLAS?

- Can it capture all graph algorithms efficiently?  
*Seems to be an open research topic*
- cf. BLAS forms a complete basis for all of linear algebra

## ■ Standard Benchmarks

- Important for many reasons - need to be chosen wisely
- **Goodhart's law**: If a benchmark becomes an optimization target than it loses its value as benchmark!
- My advice: HPC shall not ignore the datacenter folks (and vice versa)  
*Graph500 vs. LDBC?*



vs.



# Energy Efficiency?

- **Most important (Onur will tell us more)**
  - Must drive architecture innovations
  - But current DRAM protocols are really suboptimal!  
*Think bout RAS/CAS!*  
*Start a **No-DDR** movement?*
- **Watch the Green Graph 500**
  - Records progress over time  
*Same benchmark as Graph500*



See Graph500 BoF: Tue, 12:15pm, 286

MTEPS/W	June '13	Nov. '13	June '14
Small Data	64.1	153.2	445.9
Big Data	5.4	6.7	59.1



# Push vs. Pull?

- Data to computation or computation to data?
- Look at the execution DAG, assign costs, solve for balance
  - → done!

Q.E.D.