Irregular Applications and their Architectural Challenges

Pradeep K. Dubey
Intel Fellow and Fellow of IEEE

IA^3 - SC12 Workshop
Nov 11, 2012

Emerging Applications and sources of Irregularity

Who Needs Compute

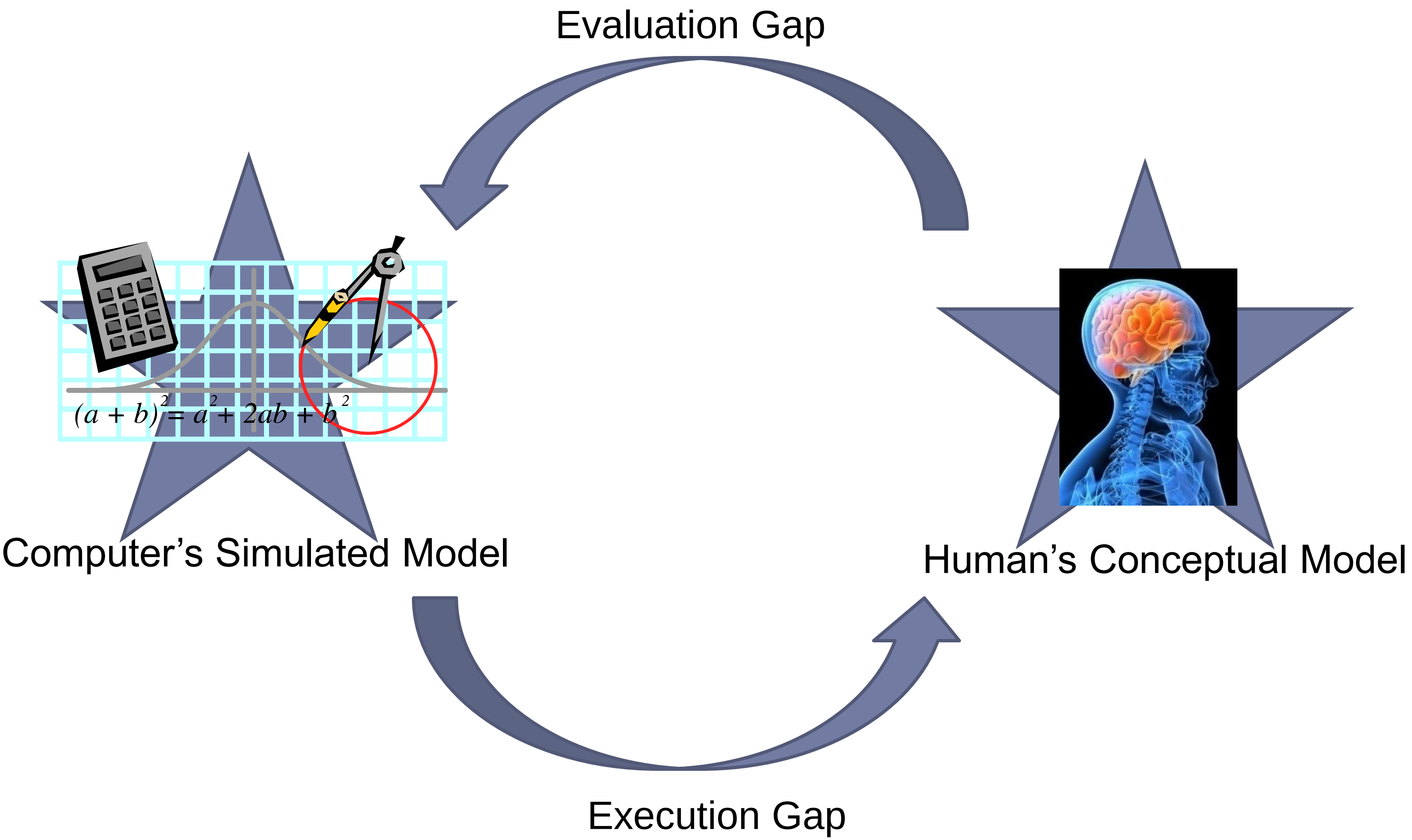
Traditional drivers of compute

- *Norman's Gulf*: Quest for natural human-machine interface
- Entertainment: Unending fascination with virtual and unreal
- The *data deluge*: The problem of drinking out of fire hydrant
- Real-time analytics: *Decision delayed is objective denied*
- *Curious minds want to know* (HPC): Science moves on!

Recent catalysts of compute

- Changing demographics of computer users
- Massive compute meets massive data
- Connected computing

Norman’s Gulf



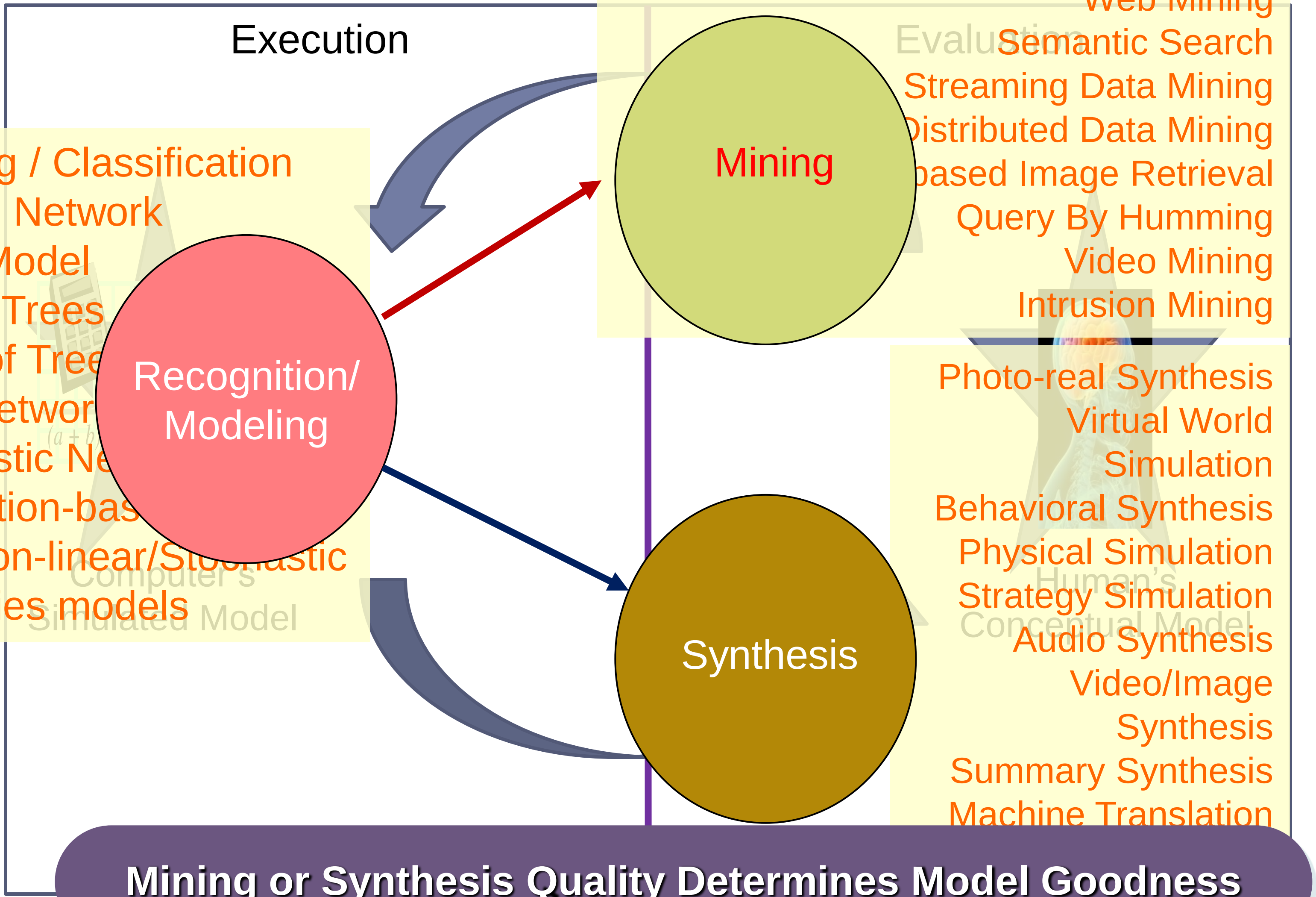
Evaluation Gap

Computer’s Simulated Model

Human’s Conceptual Model

Execution Gap

Decomposing Compute-Intensive Apps



Clustering / Classification
Bayesian Network
Markov Model
Decision Trees
Forests of Tree
Neural Networ
Probabilistic Ne
Optimization-bas
Linear/Non-linear/Stochastic
Time-series models

Data Mining
Web Mining
Semantic Search
Streaming Data Mining
Distributed Data Mining
based Image Retrieval
Query By Humming
Video Mining
Intrusion Mining

Photo-real Synthesis
Virtual World
Simulation
Behavioral Synthesis
Physical Simulation
Strategy Simulation
Audio Synthesis
Video/Image
Synthesis
Summary Synthesis
Machine Translation

Mining or Synthesis Quality Determines Model Goodness

Interactive RMS Loop

Recognition

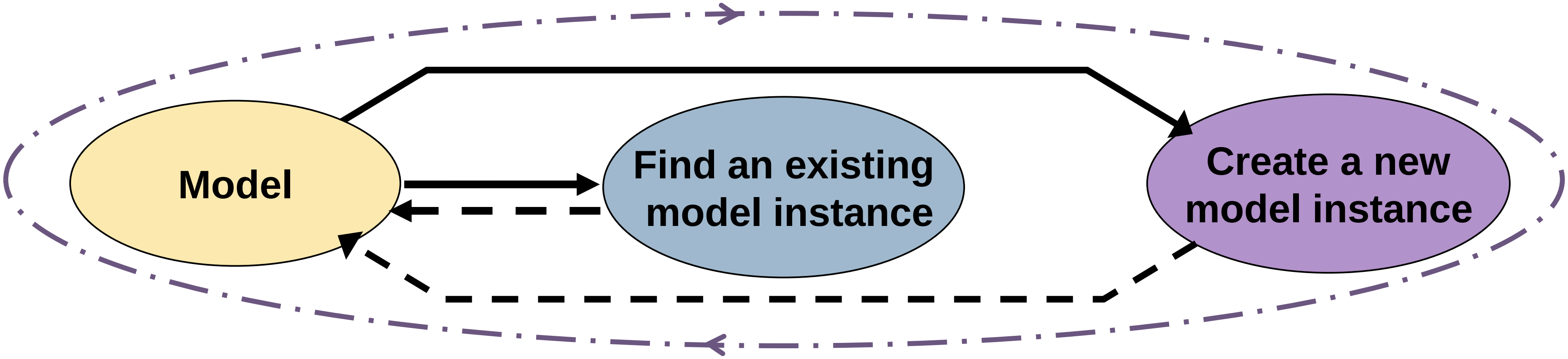
What is …?

Mining

Is it …?

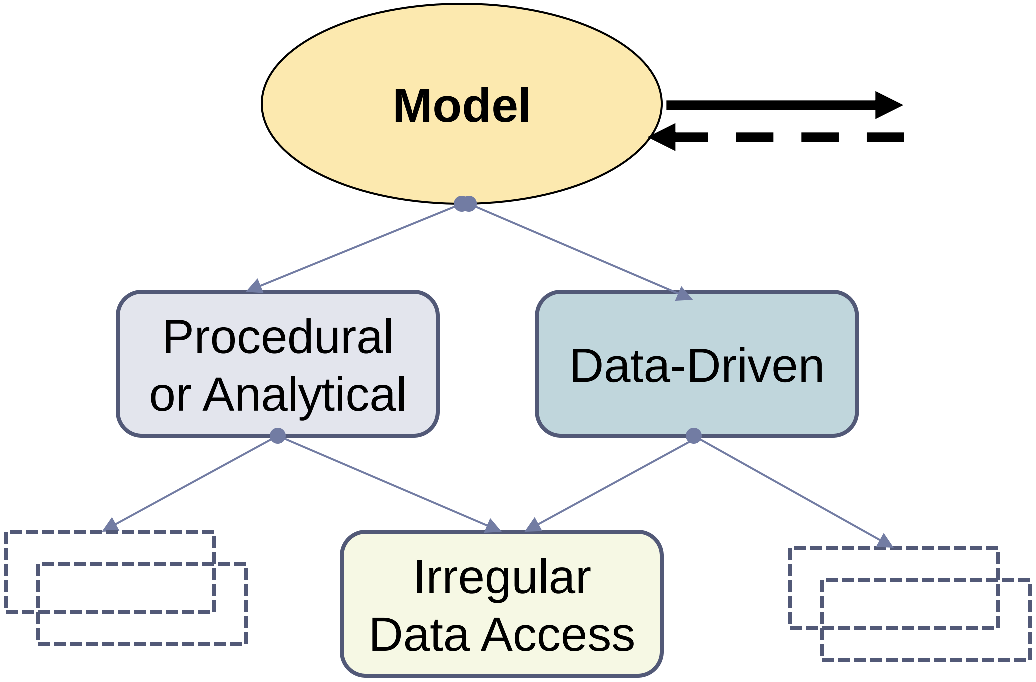
Synthesis

What if …?



Most RMS apps are about enabling interactive (real-time) RMS Loop (iRMS)

Models: Good, Bad, and Ugly?



Model

Procedural or Analytical

Data-Driven

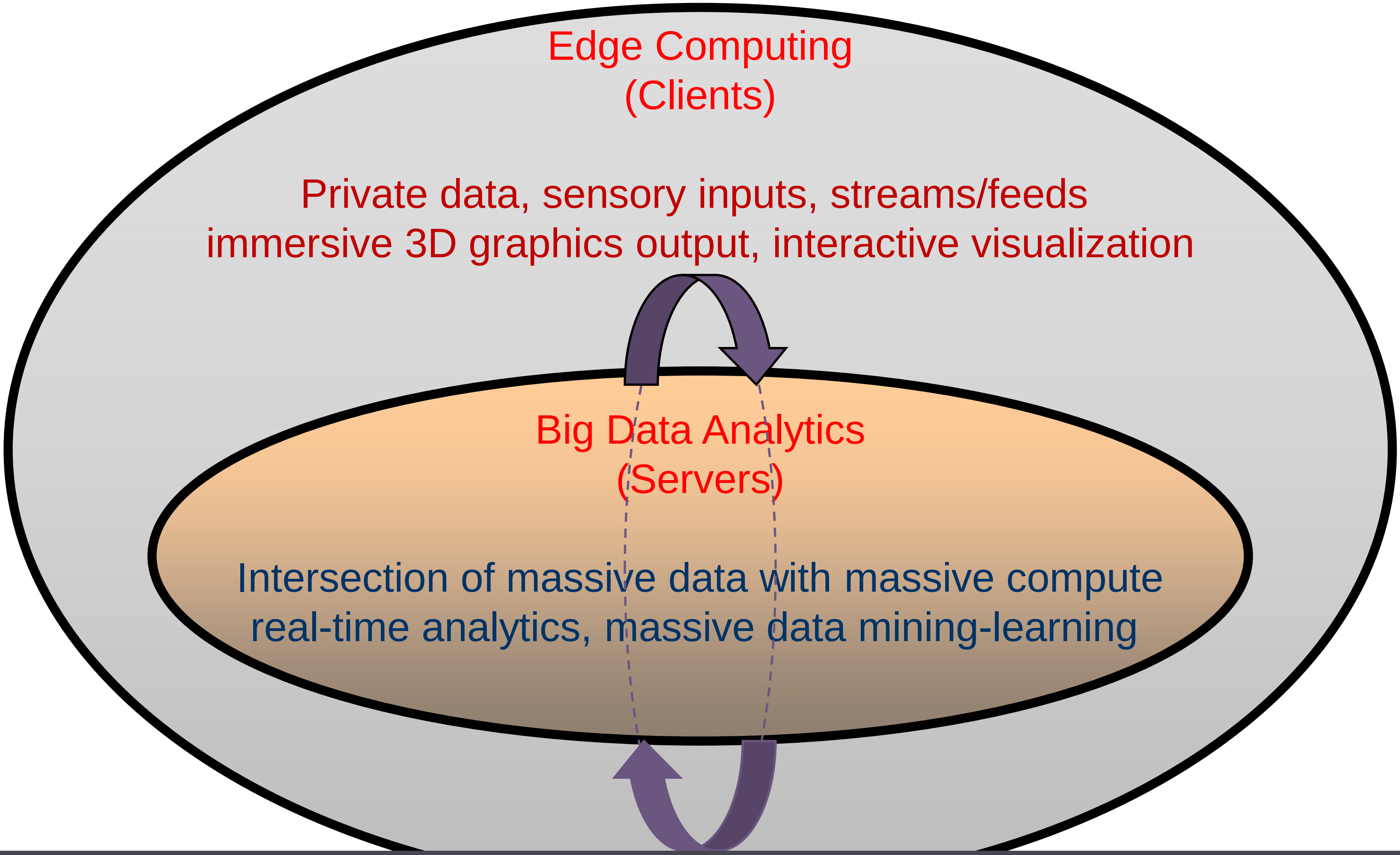
Irregular Data Access

Such as: Unstructured Grids, Graphs …

Big Data: New Catalyst for Irregular Apps

- Data-driven models are now tractable and usable
  - We are not limited to analytical models any more
  - No need to rely on *heuristics* alone for unknown models
  - Massive data offers new algorithmic opportunities
    - Many traditional compute problems worth revisiting
- Web connectivity significantly speeds up model-training
- Real-time connectivity enables continuous model refinement
  - Poor model is an acceptable starting point
  - Classification accuracy improves over time

Compute Platform Abstraction



Edge Computing
(Clients)

Private data, sensory inputs, streams/feeds
immersive 3D graphics output, interactive visualization

Big Data Analytics
(Servers)

Intersection of massive data with massive compute
real-time analytics, massive data mining-learning

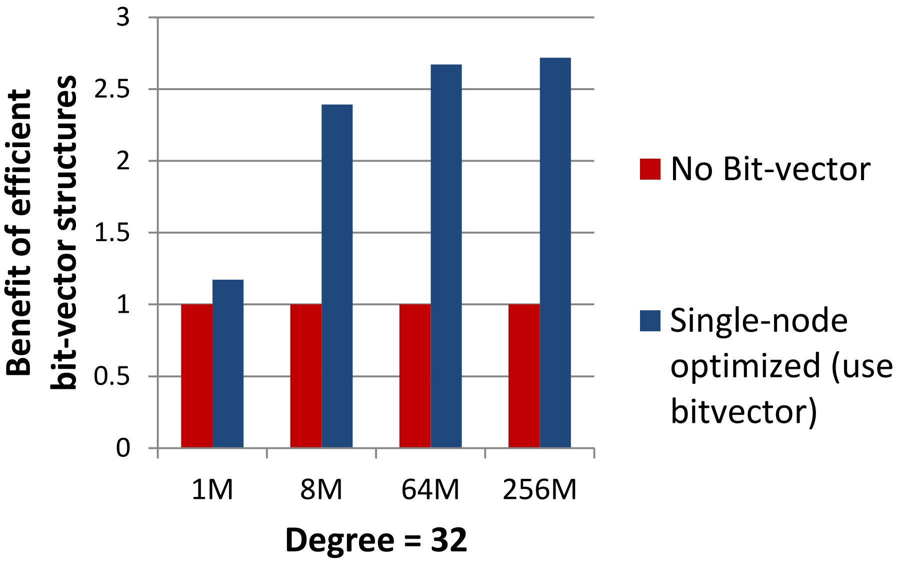
Architectural Implications Are Radical!

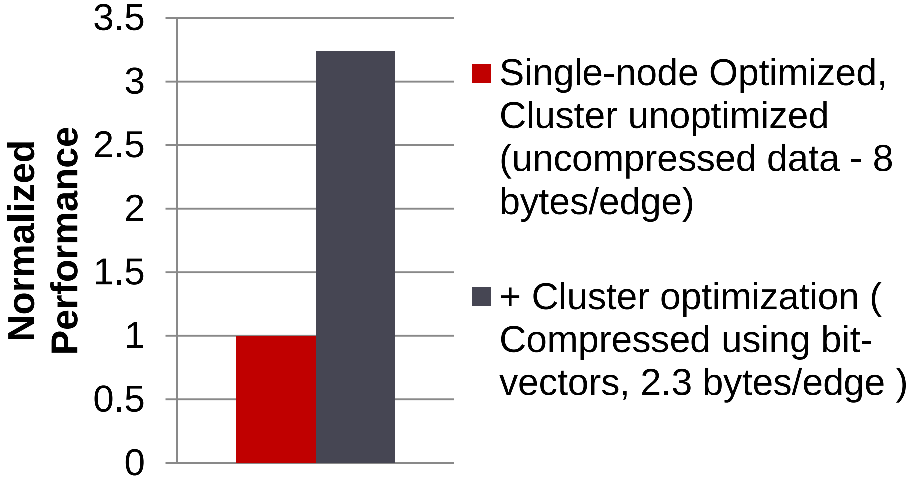
Sources of Irregularity

- Weak scaling or problem set growth
  - Growing gap of compute density wrt with memory capacity/bw growth
- More complex model – larger working set
- Lower complexity, less regular algorithm
  - Real-time response at the expense of accuracy of your response

Architectual Challenges and Success Stories So Far …

Large-scale Graph processing: Restructuring for regularity





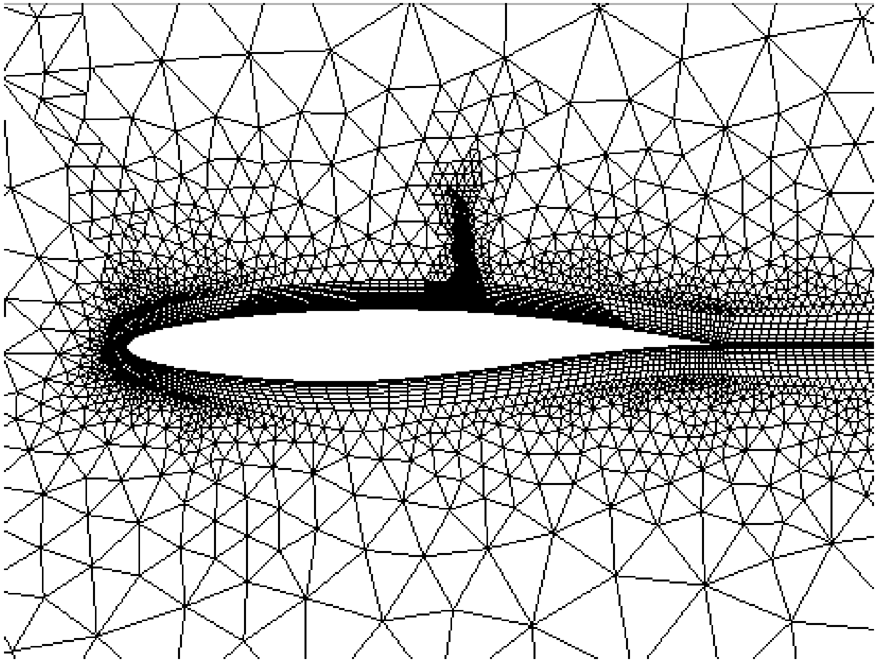
- Using bit-vector improves single-node performance by 2.6x *

- Bit-vector can be communicated across node instead of sending full neighbors and parent data
- 3.2X performance improvement on cluster *

* Fast and Efficient Graph Traversal Algorithm for CPUs : Maximizing Single-Node Efficiency", Jatin Chhugani; Nadathur Satish; Changkyu Kim; Jason Sewall; Pradeep Dubey, IPDPS'12.

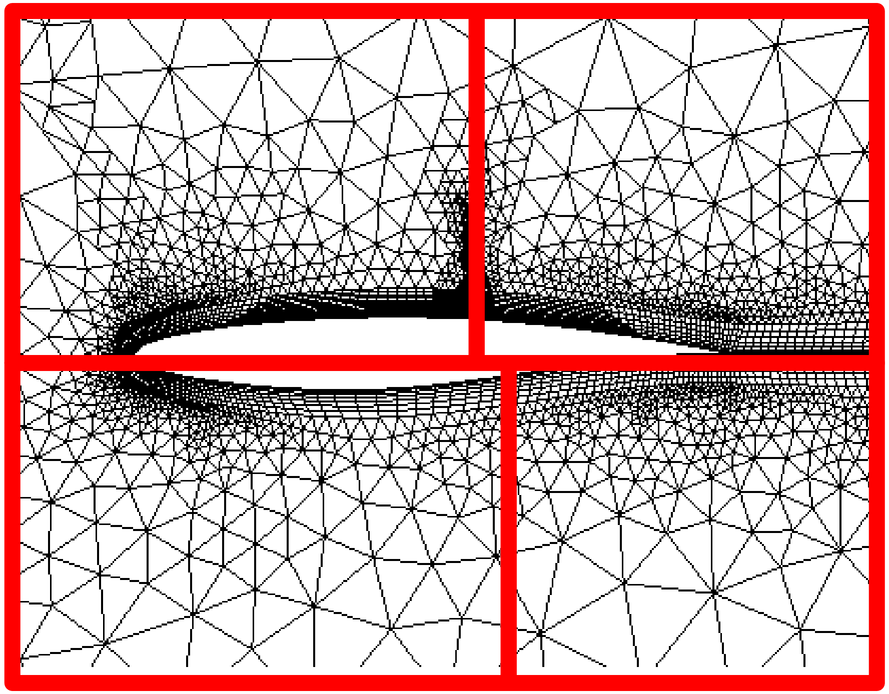
Large Unstructured Grids

- Unstructured grid problems common in HPC (aerospace, CAD, material, etc.)
- Load balanced among MPI nodes by dividing spatially
- Challenge: within node, ‘vertices’ of grid still accessed irregularly, frustrating prefetching and cache reuse
- Worst-case: *no* reuse + main memory latency hit for each vertex access!



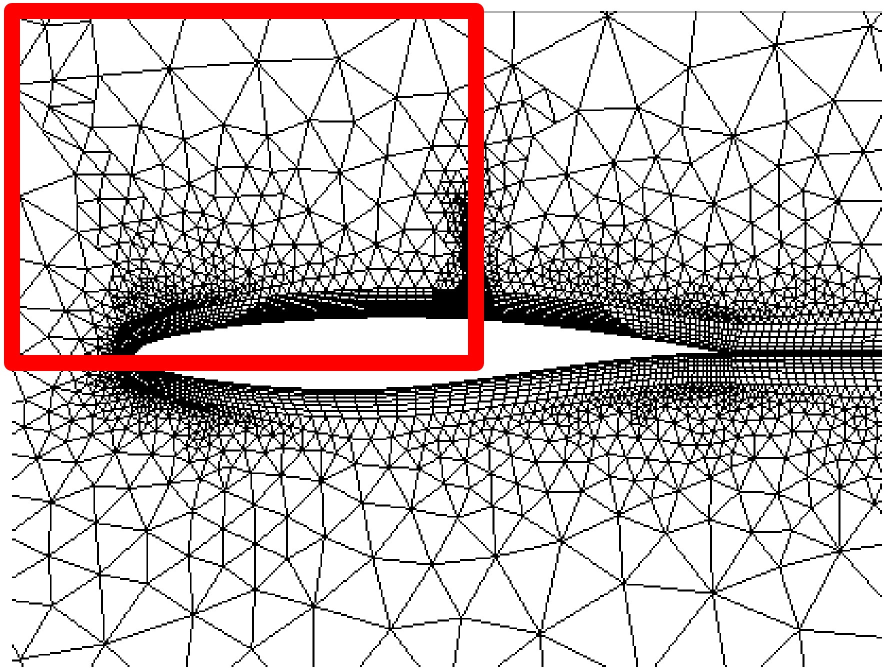
Think local: subdividing & blocking unstructured grids

- Solution: There is still structure in the ‘unstructured grid’ – keep decomposing *within nodes*
- At the *core* level, subdivide/reorder compute to maximize locality and reuse



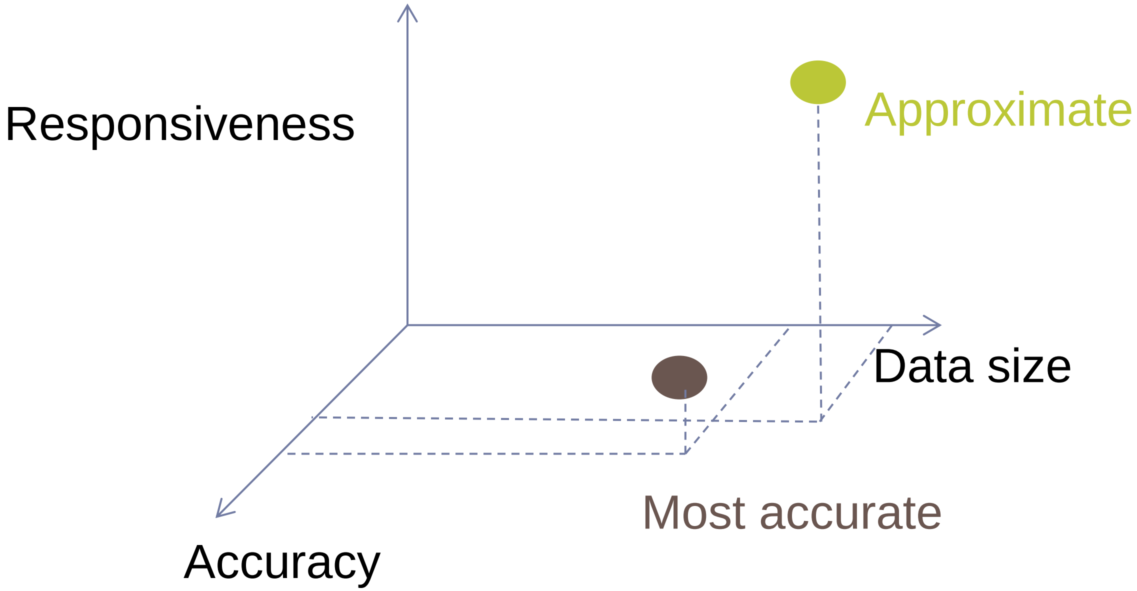
Subdividing & blocking continued …

- For many irregular problems, such blocking is *feasible* - and *necessary* with dense computing platforms!
- Decomposition size & shape based on memory hierarchy and surface-to-volume ratio.
- With prefetching, latency hidden and bandwidth reduced by average vertex degree of grid.



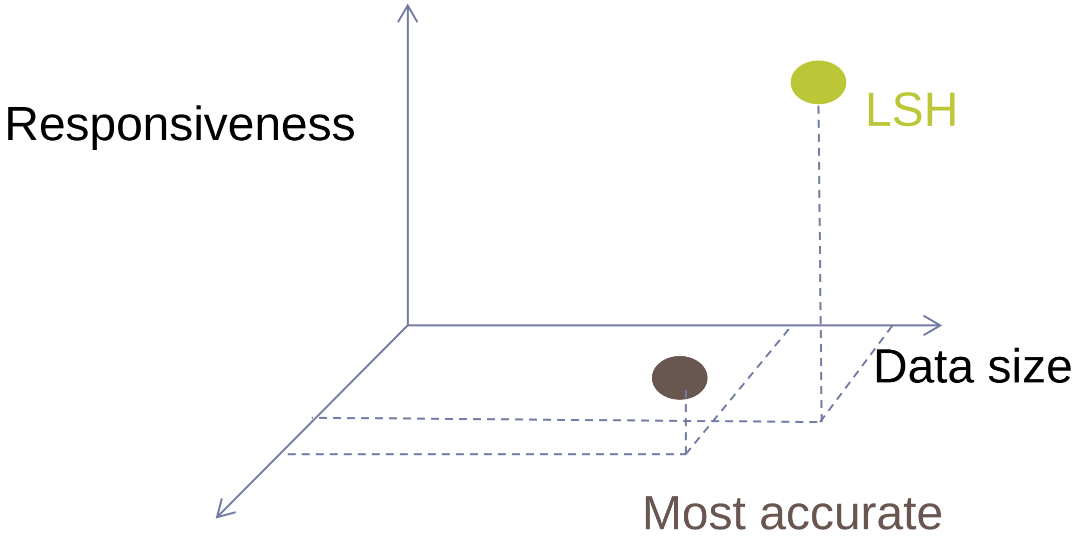
Approximate Algorithms

- Need: Process large data, give real-time responses
- Taking a small hit on accuracy enables a big performance boost



An example: Locality Sensitive Hashing

- Problem: Find nearest neighbors in a high dimensional space
- Known technique for computing approximate nearest neighbors
- Idea: Use "locality sensitive" hash functions to probabilistically aggregate nearby points



Applying LSH: Searching Twitter

- 400 million+ tweets per day
- Response time must not exceed 10 ms
- Streaming data, Dynamic updates
- LSH reduces search complexity from O(N) to O($N^{0.5}$)[*]
  - Fewer data accesses, but less predictable
  - Super-linear memory requirements O($N^{1.5}$)

[*] Gionis, A.; Indyk, P., Motwani, R. (1999). , "Similarity Search in High Dimensions via Hashing". Proceedings of the 25th Very Large Database (VLDB) Conference.

Related papers at SC'12

- *Large-Scale Energy-Efficient Graph Traversal - A Path to Efficient Data-Intensive Supercomputing; Nadathur Satish, Changkyu Kim, Jatin Chhugani, Pradeep Dubey*
  - *Session: Tuesday, November 13, 11:30 a.m.-12:00 p.m. Room 255-EF*
- *Optimization of Geometric Multigrid for Emerging Multi- and Manycore Processors; Samuel W. Williams, Dhiraj D. Kalamkar, Amik Singh, Anand M. Deshpande, Brian Van Straalen, Mikhail Smelyanskiy, Ann Almgren, Pradeep Dubey, John Shalf, Leonid Oliker*
  - *Session: Thursday, November 15, 2:30 p.m.-3:00 p.m. Room 255-BC*

Summary

- Irregularity on the rise due to application and technology trends
  - Models with higher complexity, data-driven, and real-time
  - Dense computing platforms
- Data-structure / algorithmic transforms can improve regularity and locality
- Potential for hardware improvements in cache/memory subsystem, gather-scatter, etc.
- Shortage of irregular application benchmarks
  - Graph500 is a good start but not sufficient

Notice and Disclaimers

- Notice: This document contains information on products in the design phase of development. The information here is subject to change without notice. Do not finalize a design with this information. Contact your local Intel sales office or your distributor to obtain the latest specification before placing your product order.
- INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, or life sustaining applications. Intel may make changes to specifications, product descriptions, and plans at any time, without notice.
- All products, dates, and figures are preliminary for planning purposes and are subject to change without notice.
- Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined.“ Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.
- Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.
- The Intel products discussed herein may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.
- Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or by visiting Intel's website at http://www.intel.com.
- Intel® Itanium®, Intel® Xeon®, Xeon Phi™, Pentium®, Intel SpeedStep® and Intel NetBurst® , Intel®, and VTune are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- *Other names and brands may be claimed as the property of others..

Notice and Disclaimers Continued …

- Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to http://www.intel.com/performance
- Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804