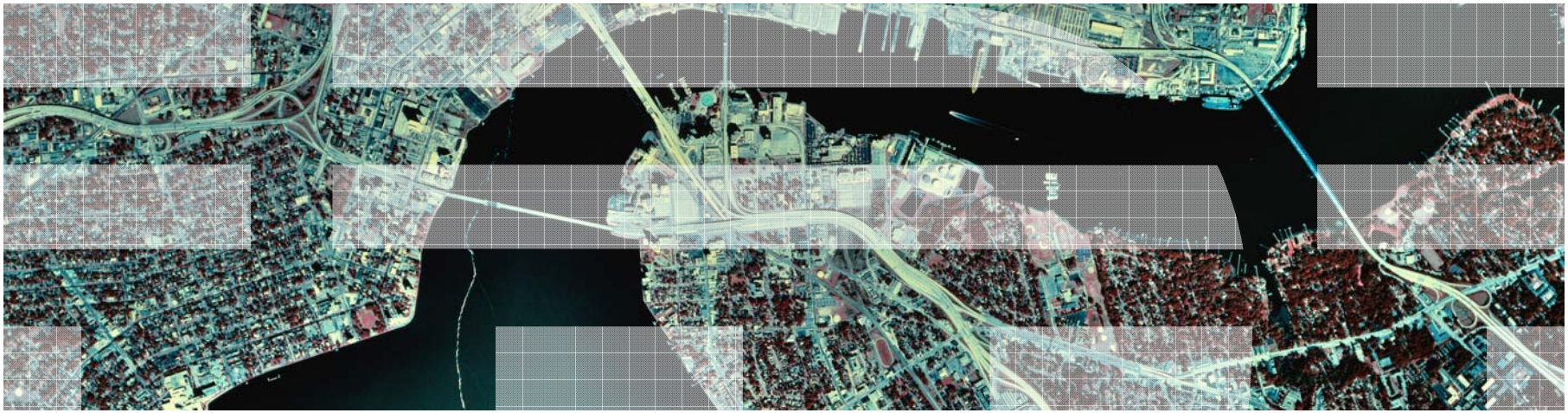
Ronald Luijten – Data Motion Architect
lui@zurich.ibm.com

IBM Forschungslabor Rüschlikon
2 März 2010

Challenges using Heterogeneous systems

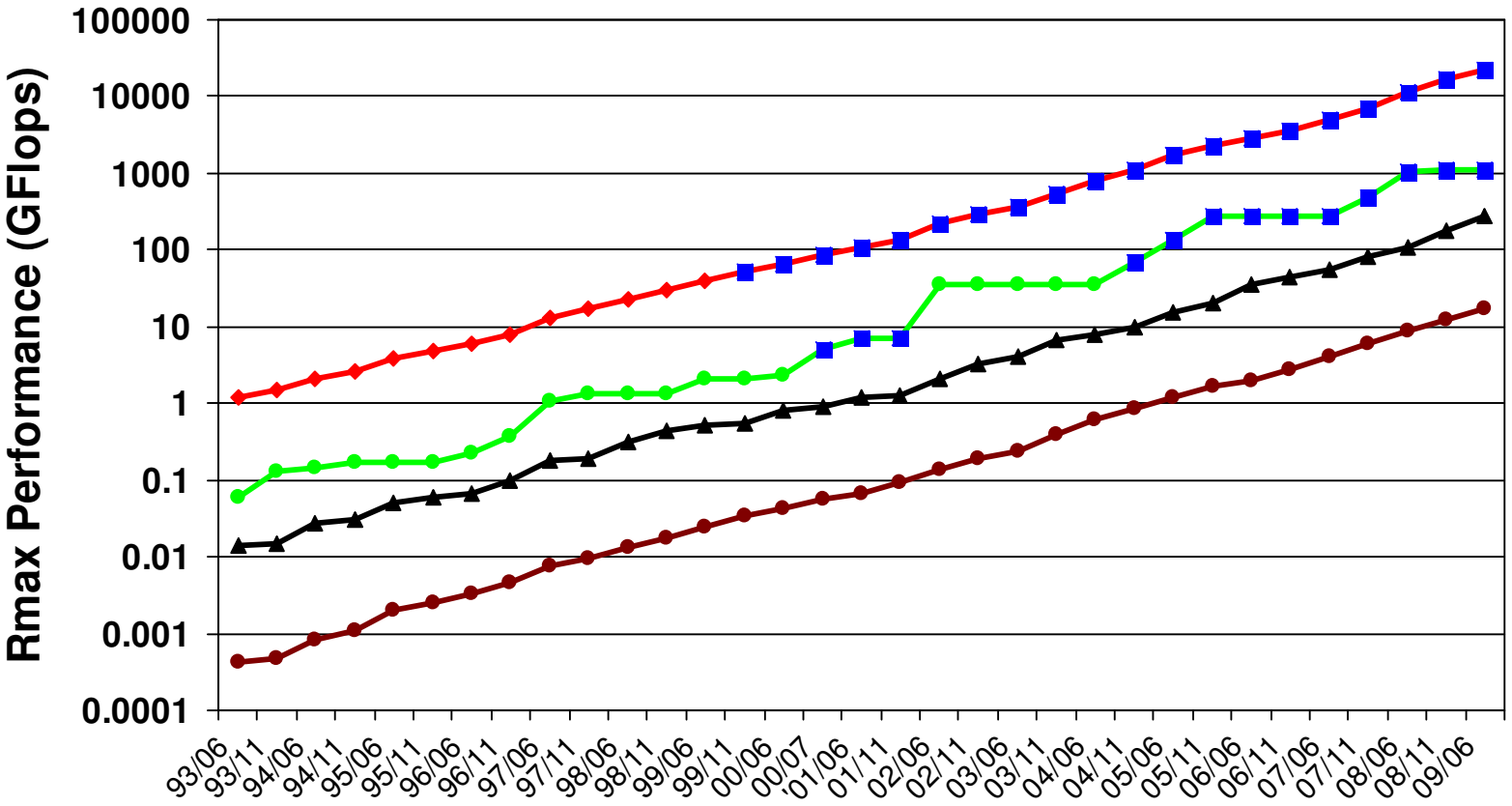


The questions on “Challenges using Heterogeneous systems”

- What makes it hard? Is it really that hard? How can we make it easier?
- Is it human aversion? Is it lack of experience or training?
- Is it the lack of a standard programming model?
- Is it the dearth of languages, compiler tools, software
- development environments and runtimes?
- How much of it is hardware limitations or specializations?

Why are we addressing these types of questions today?

- Fundamentally, HPC scaling is larger than what is supplied by CMOS scaling
- HPC: 2x every year, CMOS: 2x every 2 years
  - In 2018, be 1000x faster than in 10 years, CMOS offers 32x over same time frame



Implications…

- We need to get 32x in 10 years from somewhere else..
  - Architecture, including heterogeneous systems
  - Better programming
  - In fact, every place we can imagine
- We need to rethink the way we do our HPC systems

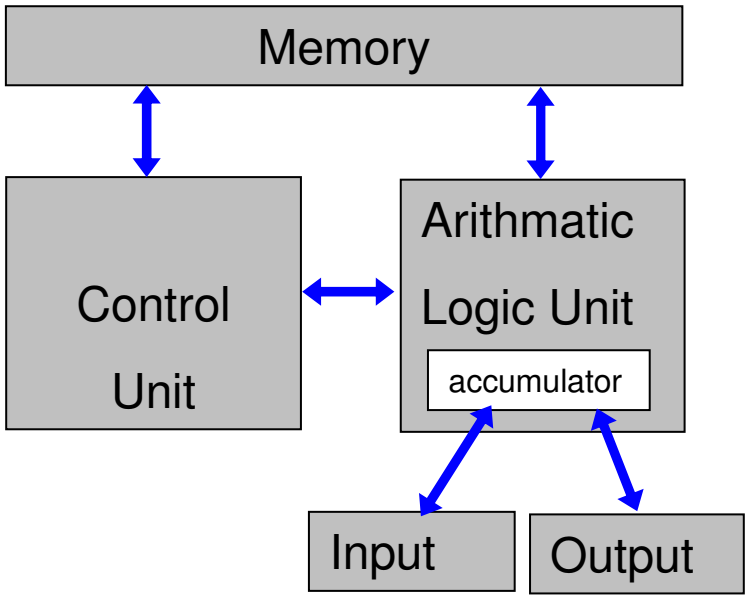
Implications…

- We need to get 32x in 10 years from somewhere else..
  - Architecture, including heterogeneous systems
  - Better programming
  - In fact, every place we can imagine

- We need to rethink the way we do our HPC systems


More’s law

Von Neumann Architecture



- John von Neumann, 30 June 1945
- 'Stored-program' concept
- Developed on the Manhattan project

- Computer architecture Basically unchanged
  - Many Improvements done
  - Basic structure the same

- But chip cost equation has changed!
  - Early generations: transistors are very expensive, pins are free
  - Today: transistor is virtually free ($10^{-8}$$ per transistor), pins are expensive ($10^{-2}$$ per pin)

- Source: Wikipedia (von Neumann Architecture); R. Kurzweil, 2009

A few good remarks I heard today

- “Data is everything”
- “simplicity needed to scale”
- “figure out how we access data and structure data”
- “ride commodity of GPUs and memory, but not of interconnect”
- “ICTs lag behind”
- “Seymour Cray and a few guys understood everything about the system”

The DATA is the problem, not the compute

- Computing is free – don't worry if some of these resources aren't used: they're free
  - Worry about the energy, though!
- Most energy is spent moving data
- We don't use CMOS scaling effectively for building our exascale systems
  - Ie. We are throwing part of 32x in 10 years out of the window
  - We're spending area to hide latency, at great energy expense
    - BTW, This is a key reason for success of Cell and GPU
  - We need to learn how to deal with latency with our programming models
- You get what you measure – measuring sustained perf as percentage of peak is wrong focus
- Most critical resource is our memory bus
  - We do not measure 'goodput' on this – you don't get what you don't measure

- I doubt that commodity CPUs and memory subsystems will present us good Exascale building blocks
- This is an Exascale level business model question

New approach needed

Given the fundamental challenges, we need to take a new look at

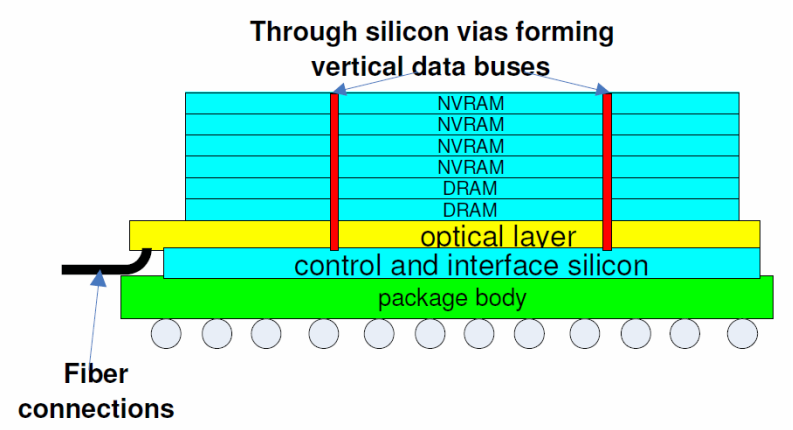
- Exascale architecture
- packaging

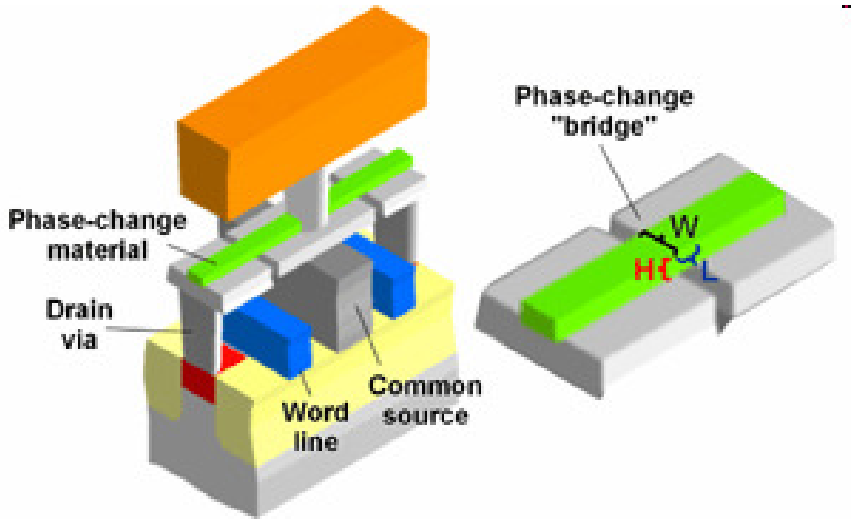
at the datacenter level level – from data motion

Also including technology disruptions

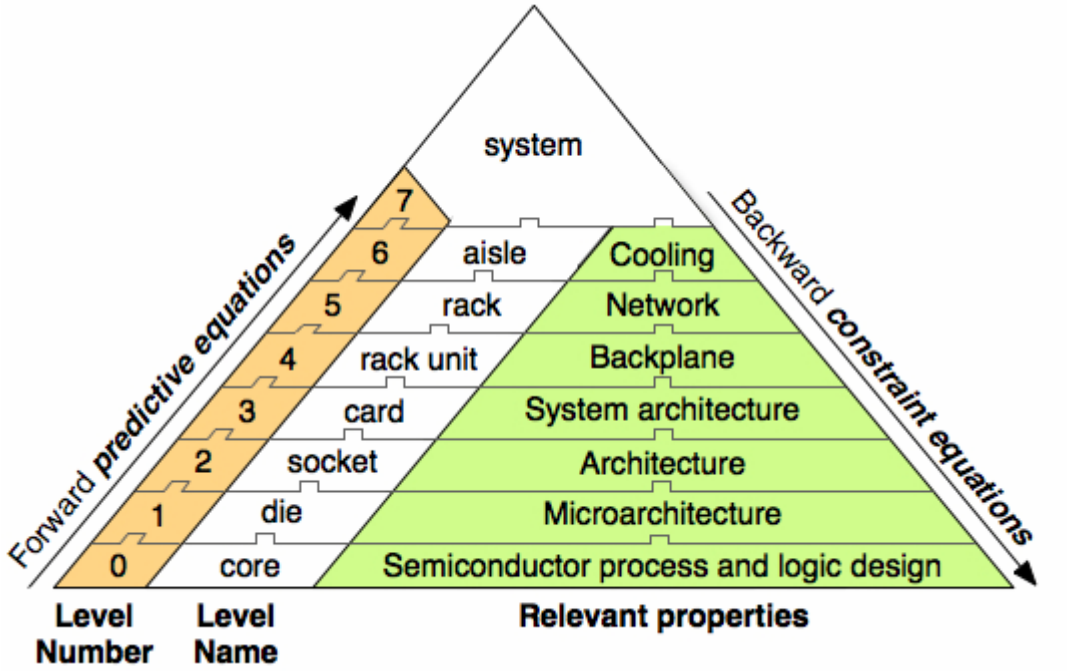
→ holistic approach needed

→ too complex to perform on whiteboard





First principles based holistic modeling effort at IBM Rüschlikon



Multi-Level Model Equation Examples

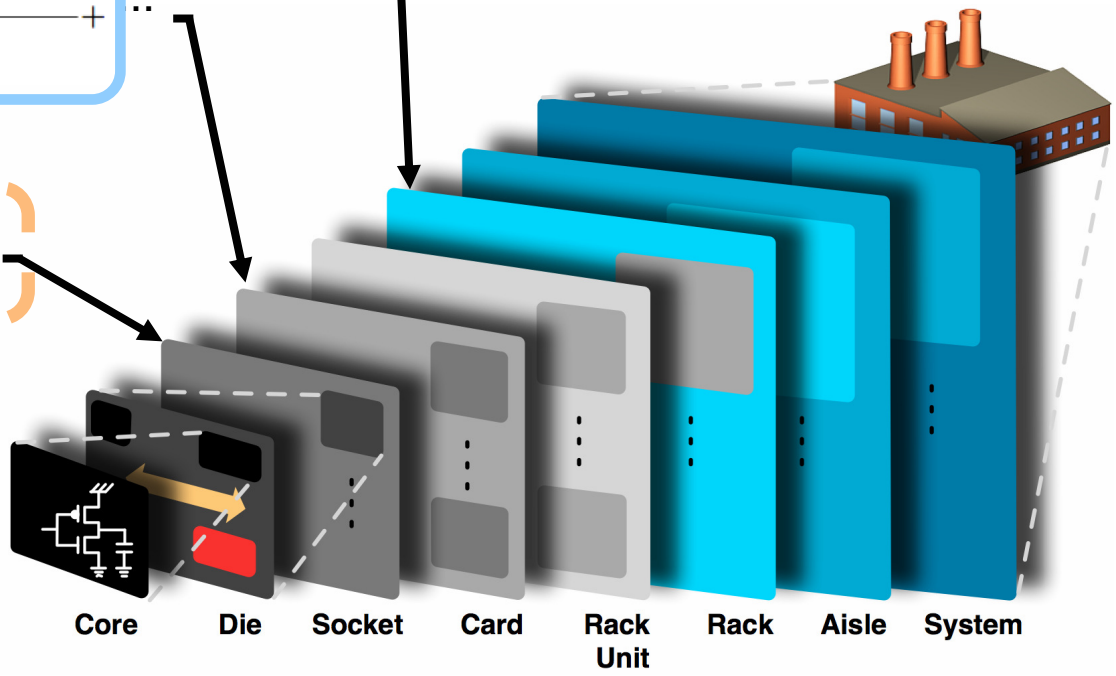
$$P_{rack} = \frac{n4_u \cdot P_{rackunit} + \sum_{x \in \{\text{netw. XLs}\}} P_{accn_x} + P_{netw}(B_{netw}, T_{netw}, n4u)}{\eta_{\text{RPOL}}}$$

$$C_{card} = \frac{\mathcal{A}2_{sw} \cdot K_{io} \cdot \left(\frac{n2_m}{A2_{app} \cdot O_{acc}(n2_m) + n2_m - A2_{app} \cdot n2_m} \right) \cdot C_{module}}{1 + \max\left(Biobus_{app} - \frac{Bperiph_{iobus}}{n2_m + n2_{aio}}, 0 \right)} + \ldots$$

$$P_{module} = \left(\frac{n1_d \cdot P_{die}}{n_{pads} \cdot V_{dd}} \right)^2 \cdot n_{pads} \cdot R_{pads} + n_{1d} \cdot P_{die}$$

$$P_{core} = K_{pc} \cdot V_{dd}^2 \cdot f \cdot \rho_{app} + K_{i_s} \left(e^{\frac{q \cdot V_{bias}}{k \cdot T}} - 1 \right)$$

$$f = K_{scale} \cdot \frac{(V_{dd} - V_t)^\alpha}{V_{dd}}$$



Notation: C_{xx}: computation throughput (FLOPS) P_{xx}: Power; f: clock frequency; K_{xx}: constant; V_{xx}: voltage; R_{xx}: resistance;
η_{xx}: power supply efficiency; ρ: application active/idle ratio, etc.