Exaflops

Design of Massively Parallel Systems

by

Dr. Narendra Karmarkar

narendrakarmarkar@yahoo.com
Massively Parallel Systems

• Given availability of
  - high density CMOS logic,
  - large volume of data, and
  - universal access through internet

Benefits of building **massively parallel systems** and creating several useful applications is well recognized

• However, several technical challenges still remain.
Imbalance between computing and communication

CMOS VLSI technology is very good for
• high density logic circuits
• and memory
and it continues to improve further as per Moore’s law

However, the major deficiency of current technology is related to movement and rearrangement of data,
• within a chip
• and between chips.

This imbalance between
• computing power and
• communications capability
Continues to grow with each generation of lithography
Obstacles to Scalability

Weak Scaling
Increasing problem size as you increase number of processors to get higher efficiency
Enables solution of bigger problem (to get better resolution etc.)
But may not decrease time to solution

Strong scaling
Solve the same problem faster using more processors
  e.g. suppose a single core desktop takes one hour to solve a problem

  Can you do it in 3.6 msec using million cores in parallel? – “ideal” strong scaling

Strong scalability requires strong communication capability,
it has been difficult to achieve due to

  higher interconnect complexity
Strong scaling is important in a number of applications:

**Rapid Feedback**
Applications that can benefit from **rapid feedback** based on simulation, rendering or other type of analysis or validation, so as to match the human **speed of creative thinking**, in both **engineering design** and **artistic design** (e.g. animation).

**Faster than real time simulation**
Important for application such as simulation of electric power grids, chemical or nuclear plants etc. to enhance safety.

**Reduce time-to-market**
Applications in which faster turn around of computational tasks of given size in support of product design can reduce **time-to-market**.
Energy cost and Heat removal

• Energy dissipation per bit communicated globally is very high, in comparison with energy dissipation per logical operation.

• At a system level total energy dissipation is unacceptably high and energy cost over life-time of the systems has become major fraction of the total cost of ownership.

• There has been significant design effort to reduce power dissipation in logic circuits and memory

• However this is not sufficient, since large fraction of the power dissipation comes from movement or re-arrangement of data within the system.
Latency and Bandwidth

Delay in moving data over larger distances is very high in comparison with gate delays and the ratio keeps increasing.

As compared to local bandwidth between ALU and register file, the bandwidth for global data movement is too low.

Each additional layer of metallization increases cost and it is not practical to increase the number of layers beyond certain point.

The physical complexity of interconnect is too high and grows non-linearly with system size.

In contrast computing power grows linearly with number of cores.
Therefore imbalance keeps growing.
These difficulties have persisted at microelectronics level and have worsened over several decades.

In early days of parallel computing these were,

Pushed to o.s. and compilers

Then to libraries

And now to application programmers

This increase in programming complexity is hampering full exploitation of the power of silicon technology.

**INSTEAD CAN THESE DIFFICULTIES BE SOLVED AT THE MICROELECTRONICS LEVEL ITSELF?**
Fundamental reasons?

- Are these difficulties due to any
  - Logical Necessity -- complexity theory result?
  - Physical necessity -- Laws of physics preventing better systems?

It turns out its neither; just historical --

- Incremental evolution of computer architecture is trapped in a local minima
- Fresh thinking needed to pull us out towards better alternatives
Alternative Approaches

- **Other Approaches:**
  - All Optical computing
  - Superconducting Logic, etc.

- **Our Approach:**
  - Recognize that CMOS is very good at logic
  - Don’t fix what isn’t broken
  - Processing power of CMOS grossly underutilized.

- Instead, supplement Moore’s law selectively:
  - Supply the “missing” component

- **Our new design - very powerful Global communication**
  - Low latency, very high bandwidth
  - Extremely low energy/bit, scales with gate capacitance
  - High packing density
  - Scales well with CMOS scaling
This design is based on confluence of ideas from many fields

- Math (Optimization theory, Discrete subgroups of lie groups)
- CS (Parallel Architectures)
- Physics (Path integrals, Quantum tunneling, Optics, Electron Optics)
- EE (CMOS & MEMS processes, Field emission devices, Control Theory)
- Material Science (Generalized interference lithography)
Several New Designs

- **Parallel supercomputers** based on:
  - massively parallel quantum tunneling,
  - electron optics and
  - finite projective geometry.

- As compared to contemporary designs,
  - overcomes the bandwidth and latency obstacle.
  - large improvement in teraflops per kilowatt,
  - significant reduction in programming complexity
  - broad applicability to many domains, including those requiring a lot of pointer chasing.

  can be implemented using known fabrication techniques.
New Designs

• Secondary Storage
  - low latency, multi-ported
  - based on magneto-optics
  - implements shared memory directly at physical level.
  - highly valuable feature to business data bases as well as applications requiring frequent shared access to massive amounts of common data such as Google earth.

• High bandwidth switches
  - required for building next generation internet infrastructure.

• based on insights derived from our advances in interior point theory of global optimization
Interior Point Approach

• More than just world’s fastest linear programming solver

• Has many other powerful applications to:
  - combinatorial problems
  - logic, theorem proving
  - inductive inference, machine learning
  - Global optimization
Global Optimization

• Traditionally, optimization theory deals with “convex” problems essentially unique global minimum

• However, optimization problems, with multiple, global minima, arise in many contexts
  - In nature
  - In engineered systems

• They lead to number of problems
  • Finding Optimal Solution
  • Proving Optimality
  • Inverse Problem
  • Synthesis
Problems related to Global Optimization

• Finding Optimal Solution
  Given energy function – find minima (value and locations)

• Proving Optimality
  - Given one such solution,
  - How does one prove that there is no better solution?
  - What types of proofs are possible?
  - How long such proofs might be?
  - How do different global minima relate to each other?

• Inverse Problem
  - If we know all (or many) global minima,
  - How can we find the energy function?
  - This is like “reverse engineering” the nature.
Synthesis

• Given specification of desired global minima in terms of their
  - value
  - locations
  - or shape of energy landscape around the minima,
  - how do we design a system with appropriate energy function,
    subject to further constraints?
  - mixture of “direct” and “inverse” problems in the same context.

• All these questions are
  - related to each other and also to
  - how systems occurring in nature behave
  - how artificially engineered systems might be designed.
Energy Functions with Multiple Global Minima in Physical Systems

Example – Crystal Structure

Potential energy function is periodic in 3D

The related structure has Translational Symmetry
Mathematical Analysis of the structure

- Motion of electrons in the resulting structure can be analyzed theoretically.
- Theory of semiconductors based on such analysis
- 3D translational **symmetries** of the structure form a **group**
- “unit cell” of the crystal – called **fundamental domain**
- Translated copies of the fundamental domain exactly fills up 3-space.

Similarly, we artificially create a structure with **symmetries we want**
Artificially Created Symmetry
Analysis of artificially created symmetric structure

- We are able to analyze motion of electrons in such structure theoretically, based on our new theory. (previous theory could only analyze electron optical systems with dominant “central” optical axis)

- Picture shows 2D projections of calculated 3D electron trajectories.

- The structure is based on finite projective geometry, which forms the basis of inter-processor communication in this architecture.

As before, copies of the fundamental domain made by action of the symmetry group fill the free space exactly.

Within each fundamental region, there is a (conceptual) “tube” for electron flow.

Quantum tunneling at the origin of each tube is very fast ( ~ 10^{-14} sec.)
**Basis for energy efficiency of the structure**

Quantum tunneling is a loss-less process.

Movement of electrons thru’ free space is also loss-less as there are no collisions.

Energy electrons gain in the first half of the flight (acceleration phase) is returned to the field in the second half of the flight (deceleration phase).

As a result, this *surface normal* communication link in 3rd dimension is more *energy efficient*, than conventional *surface parallel* links on the chip itself.

Only energy loss is in charging/discharging of gate capacitances in drivers and receivers, which is similar for on-chip local links between logic blocks.

Such 3D links are formally as powerful as infinite number of 2D links. Once you have these, you can reduce the number of metallization layers in CMOS and save cost.
Encoding information on electron channels.

- Again exploit multiple global optima in quantum systems –

- Choose a quantum system where a priori (without any electric field) tunneling probability has number of global maxima for distinct outcomes

- Create bias in favoring one of the outcomes by applying voltage, encoding one out of many possibilities.
Multi-threading and multiple global minima

In a multi-threaded information processing system, many things happen in parallel at the logical level.

It is but natural that the physical device carrying out such processing be governed by energy function with multiple global minima.

Instead of forcing a system with unique global minimum to Perform multi-threading through system software.
Introduction to finite projective geometries

Consider a finite field $F = GF(s)$ having $s$ elements where $s = p^k$, $p$ – prime number, $k$ – positive integer.

Take a $(d+1)$ dimensional vector space over the field $F$

Each ray through the origin gives a point of the projective space

Each $(n+1)$ dimensional subspace of the vector space give $n$ dimensional Subspace of the projective geometry

Number of points in the projective space given by

$$|P^d(F)| = \frac{s^{d+1} - 1}{s - 1}$$
Architecture based on Projective Geometry

1. Select parameters of the geometry

\[ p, k, d \quad \mathbb{P}^d (GF(p^k)) \]

Let \( \Omega_l \) denote the collection of all projective subspaces of dimension \( l \).

Thus,

\[ \Omega_0 \quad \text{- set of all points} \]
\[ \Omega_1 \quad \text{- set of all lines} \]
\[ \Omega_{d-1} \quad \text{- set of all hyperplanes etc.} \]

Consider collection of subspaces of dimensions 0, 1, \( \ldots \), \( d_{\max} \)

2. Each hardware resource is associated with a subspace

3. Two resources corresponding to subspaces \( X, Y \) are connected iff \( X \subseteq Y \)

and \( \dim (X) = \dim (Y) - 1 \)
Rule based allocation of computational resources

Smallest dimension of subspace containing all the required data and available computing resources does the computation.

This rule can be directly implemented in hardware.

Examples of computation done at various dimensions

1. Computation done at points (dim = 0) ------ PIM

2. Computation done at lines (dim = 1)
   Consider a binary operation $a \leftarrow a \circ b$
   Suppose $a \in$ Memory Module corresponding to point $a$
   Suppose $b \in$ Memory Module corresponding to point $b$
   Pair of points $(a, b)$ determine a unique line
   Processor associated with the line is responsible for the operation

3. Computation done at plane (dim = 2)
   Join operation in relational databases
Decomposition of binary relations

Map

1. items → point
2. Pair of points → line
   (Rows of relation)
3. All rows mapping to the same line stored on same memory module

Let p = processor corresponding to plane < i, j, k >
• Each processor performs portion of the join it can “see” directly

• Parallelizes automatically

• Location and access to a pair depends on its value and not just on some arbitrary address

• Generalizes content-addressable memory to combination of values

• Exploit “semantic” locality based on values, traditional special and temporal locality based on addresses

4. Computation at higher dimensions (dim > 2)

Very powerful for Algorithms involving

Tensors

Hypergraphs etc.

as compared graph search algorithms, hypergraphs can capture more semantics in structural form
Virtual Memory Organization based on subspaces

Generally, virtual memory is organized in a hierarchical fashion:

Total memory space is divided into pages.
Pages consist of words and words consist of bits.

In the projective geometry architecture there are superpages organized in the form of lattice.

Two superpages are either disjoint or intersect in another superpage (or page at the bottommost level).

Each superpage is associated with a subspace of the projective space, and intersection of two superpages is associated with the intersection of corresponding subspaces.

Accessing many pages from the same superpage can lead to predictive fetching of the entire superpage enabling exploitation of another kind of locality that is frequently present in many applications.
A Perfect Access Pattern is a collection of n ordered pairs of points s.t.
1. First members of all pairs $(p_1, p_2, \ldots, p_n)$ form a permutation of all pts
2. Second members $(q_1, q_2, \ldots, q_n)$ - permutation.
3. The lines $(l_1, l_2, \ldots, l_n)$ determined by these pairs form permutations of all lines of the geometry.

<table>
<thead>
<tr>
<th>N</th>
<th>Point Pairs</th>
<th>Corresponding lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$(p_1, q_1)$</td>
<td>$l_1 = &lt;p_1, q_1&gt;$</td>
</tr>
<tr>
<td>2</td>
<td>$(p_2, q_2)$</td>
<td>$l_2 = &lt;p_2, q_2&gt;$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>N</td>
<td>$(p_n, q_n)$</td>
<td>$l_n = &lt;p_n, q_n&gt;$</td>
</tr>
</tbody>
</table>
Clearly, if one schedules binary operations corresponding to such a set of index–pairs for parallel execution

1. There are no read–write conflicts in memory accesses.
2. There is no conflict in processor usage
3. All processors are fully utilized
4. Memory bandwidth is fully utilized

Hence the name **perfect pattern**

**Definition**

A collection of perfect patterns is called **complete** if every index–pair \((a,b)\) \(a \neq b\) occurs in exactly one pattern
**Perfect Access Patterns for 4d Geometry:**

Let \( n = \) number of planes \( = \) number of lines

<table>
<thead>
<tr>
<th>Triplet of points</th>
<th>Triplet of lines</th>
<th>Planes</th>
</tr>
</thead>
<tbody>
<tr>
<td>((p_1, q_1, r_1))</td>
<td>(u_1 = &lt;p_1, q_1&gt;)</td>
<td>(v_1 = &lt;q_1, r_1&gt;)</td>
</tr>
<tr>
<td>((p_2, q_2, r_2))</td>
<td>(u_2 = &lt;p_2, q_2&gt;)</td>
<td>(v_2 = &lt;q_2, r_2&gt;)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>((p_n, q_n, r_n))</td>
<td>(u_n = &lt;p_n, q_n&gt;)</td>
<td>(v_n = &lt;q_n, r_n&gt;)</td>
</tr>
<tr>
<td>(h_1 = &lt;p_1, q_1, r_1&gt;)</td>
<td>(h_2 = &lt;p_2, q_2, r_2&gt;)</td>
<td>(h_n = &lt;p_n, q_n, r_n&gt;)</td>
</tr>
</tbody>
</table>

A **perfect pattern** is a collection of \( n \) (non-collinear) triples such that

- Lines \( u_1, u_2, \ldots, u_n \) determined by first pair of points from each triplet forms a permutation of all lines.
- Similarly, lines determined by pair \((q_i, r_i)\) form each triplet form a permutation of all lines and lines determined by pairs \((r_i, p_i)\) also form a permutation.
- Planes \( h_1, h_2, \ldots, h_n \) determined by the \( n \) triplets form a permutation of all planes
Automorphisms of the geometry.

- It maps points $\rightarrow$ points
- lines $\rightarrow$ lines
- planes $\rightarrow$ planes
- Any subspace $\rightarrow$ Another subspace
- Of dimension $k$ $\rightarrow$ Of dimension $k$

Automorphism group of the geometry acts \textbf{transitively} on all subspaces of the geometry.

Perfect patterns and complete collection of perfect patterns can be generated using the automorphism group of the geometry.

For implementation of perfect patterns using electron optics, we express the required electromagnetic fields in terms of \textit{symmetrised} multi-pole expansion using suitable subgroup of the automorphism group.

To show pictures of high dimensional geometries we often use 2D projections of electron beam trajectories implementing that geometry.
Here are some examples of 2d geometries

7 point geometry
13 point geometry
21 point geometry
GF(4) is an extension of GF(2)

21 – point geometry containing 3 copies of 7 – point geometry
57 point geometry
91 point geometry
Electromagnetic Cavity Machine

Electron flow tubes
Electromagnetic Cavity Machine

The **surface** of this cavity is lined with CMOS logic circuits.

Organized as millions of **ultra low power cores** designed to enable **massive multithreading**.

A peta-flop configuration requires only about a square meter of silicon real estate for computing circuits even if they are designed to run at slower clock speed to significantly improve performance per watt.

Besides computing, the surface circuits also provide for two types of communication devices between computing resources.

The first type is for the traditional 2D nearest neighbor communication.
Surface Normal Communication

The second type is for a new type of surface normal communication
- based on massively parallel quantum tunneling and
- free space electron optics through the cavity volume

This type of global communication uses highly symmetric flow patterns derived from mathematical structure of finite projective geometry.

Its implementation involves a novel electron optical system that does not have any dominant “central optical axis”

The electromagnetic fields to guide electrons along the required massively parallel trajectories is set up by creating appropriate boundary conditions on the surface of the electromagnetic cavity.

The electrodes required to apply such boundary conditions are patterned on the surface of the cavity by standard lithographic techniques.

The drivers, receivers and other controlling electronics required for this purpose is located on the surface of the electromagnetic cavity.
Communication based on electron optics through the cavity volume

- We choose a complete set of perfect patterns or a k-fold covering of perfect patterns.

- Corresponding to each perfect pattern, the cavity volume is partitioned into fundamental domains.

- Each fundamental domain contains a “tube” in free space for field-emitted electrons.

- The ends of the tube have field emitters and detectors located on the cavity surface.

- When the perfect pattern changes the emitters and detectors are connected by a new configuration of electron tubes.
Illustration of this process with a simple example

• Choose the **smallest** \((p=2, k=1, d=2, 7\text{-point})\) projective geometry.

• Choose one perfect pattern for the geometry.

Example of one perfect pattern for 7 point 2d geometry
Fig. 7
Partitioning of the cavity volume into fundamental domains

An example of building block for partitioning free space

Some curved sides of the building blocks have tangents which correspond to same element of lie algebra
Example of space partition of the cavity volume corresponding to a perfect pattern

A Perfect pattern for 7 point 2d geometry

Corresponding volume partition
Example of tubes for electron flow for the perfect pattern

A Perfect pattern

Corresponding electron flow tubes
The vertical component of the electron motion is due to electric field.

The circular component is due to magnetic field in the axial direction resulting in overall helical motion.

Different perfect patterns from a complete collection can be achieved simply by changing the ratio of magnitude of the electric to magnetic field.

The **physical complexity** of connecting \( n \) sources to \( n \) destinations in one hop using a complete set of perfect patterns is \( O(n) \).

Alternative arrangements that have been used in contemporary designs for one hop connections is \( O(n^2) \).
Alternative approaches

Multihop Networks

- Increases latency,
- multiplies energy used per bit communicated
  by the number of hops since each time a bit is received, detected, amplified and retransmitted, additional energy is consumed.

Earth Simulator (Japan)

- a single hop, very good architecture from the point of view of generality, but uses connections of $O(n^2)$ physical complexity

Further refinements --- replace electrical cables with optical fibers and dense WDM. This saves optical fibers (which is the cheapest component anyway).
However the total physical complexity is still $O(n^2)$ since there are $O(n)$ such MUX/DMUX components, each of $O(n)$ complexity.
In contrast, our design provides communication channels with high bandwidth, low latency, single hop, electromagnetically reconfigurable, energy efficient, and the physical complexity of the overall design is $O(n)$. 
Other benefits

High Packing Density

Unlike optoelectronic devices in the infra-red range, whose packing density is order of magnitude worse than current logic circuits, due to diffraction limit, field emission devices for surface normal communication can be packed with much higher density ($\sim 10^8$ / sq. cm)

Single Electron Devices

In our design, both logic circuits and communication devices on the cavity surface will be ultimately based on quantum tunneling, in tangential and normal directions respectively, and both can be operated as single electron devices.

“A conveyor belt for electrons”
Conclusion
The new design leads to massively parallel systems that are
highly energy efficient, and compact
significantly simpler to program
and broadly applicable to many domains

• These advantages more than compensates for
  moderately higher complexity of our physical design.

• After all, one can manufacture millions of identical copies of
  hardware, once designed.

• This will enable focusing of programming efforts
  on the creative aspects of software design for various applications.

• The power of silicon technology and many ideas in computer
  science can be fully utilized.
For more information/ papers :

Follow links on wikipedia page about author
email: narendrakarmarkar@yahoo.com

For more technical information see

MIT mathematics department web site
www-math.mit.edu/crib/08/jul25.html

For general description see
Video of google presentation on youtube
Hypothesis Graphs

Organisation of the projective geometry machine in terms of perfect patterns allows exploitation of parallelism at a fine grain level.

To enable this in presence of conditional branches, we use a concept of hypothesis graphs, which is a slight extension of data flow graphs, augmented with Boolean predicates.

A Boolean variable associated with a data variable represents validity of the data value.

Operations in the data flow graph compute Boolean predicates in the same way as data operations.

When Boolean predicate associated with a node evaluates to false, that branch of the computation terminates. No computation is ever rolled “back”.

Although this concept was originally meant to help deal with conditional branches, it is also turning out be extremely useful for efficient computation and simulation under multiple scenarios if they share significant amount of common computation.