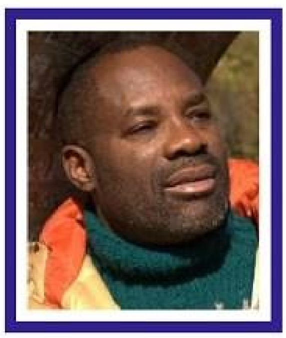
12 How I Invented a New Supercomputer



Philip Emeagwali
info@emeagwali.com 202-203-8724

Philip Emeagwali Lecture 180120-2
Visit http://emeagwali.com for complete transcripts of 100+ lectures.
Video:
https://www.youtube.com/watch?v=pSmC2mZ0Uio

https://YouTube.com/emeagwali
Podcast: https://SoundCloud.com/emeagwali

12.1 My New Supercomputer is a New Internet

12.1.1 How I Invented a New Supercomputer

My experimental discovery
of 1989
of how and why parallel processing
makes modern computers **faster**
and makes the new supercomputer
the **fastest**
was not in the supercomputer textbooks
that were printed in the 1970s and '80s.
That experimental discovery
of parallel processing
was a real game changer
that ushered an explosion of research
and the commercialization
of the modern supercomputer
that computes many things **at once**.
In the 1980s,
they was a big gap

between the fastest supercomputer
we had
and the fastest supercomputer
we needed.
I started my quest
for the parallel processing supercomputer
with the question:
“Can parallel processing
be experimentally confirmed?”
I began my quest
for the fastest supercomputer
by stating my parallel processing hypothesis,
namely,
that I could evenly divide
each grand challenge problem
of extreme-scale computational physics
and divide it
into 65,536 less challenging problems.
My central experiments
that led to my discovery
of parallel processing

comprised of speed up measurements
across a new internet
that I visualized
as my global network of
64 binary thousand processors.
After my decade of trial-and-error
in programming loosely-coupled
ensembles of processors,
I experimentally discovered
a speed increase of a factor of
64 binary thousand
and discovered that speedup
across as many processors.
What made the news headlines
in 1989
was that I experimentally discovered
massively parallel processing
and invented the technology
when supercomputer textbooks
considered computing many things **at once**
and doing so to solve

extreme-scale problems in physics
and beyond
to be both theoretically
and **physically impossible**.
I discovered
that parallel processing
is not a waste of time.
I discovered that
computing many things **at once**
makes modern computers **faster**
and makes the new supercomputer
the **fastest**
and I **invented**
how to use
that **new supercomputer knowledge**
to build a **new supercomputer.**
My discovery
of massively parallel processing
led me to discard
the sequential processing hypothesis
that was erroneously formulated

226

by **Gene Amdahl** back in April 1967
and that was the reigning
supercomputing paradigm
of the 1940s, '50s, and '60s.
My discovery
of massively parallel processing
led me to discard
the vector processing hypothesis
that was championed
by **Seymour Cray**
and that was the reigning
supercomputing paradigm
of the 1970s and '80s.
My discovery
of parallel processing
made the news headlines
and was in the June 20, 1990 issue
of *The Wall Street Journal*
and was in the June 27, 1990 issue
of *The Chronicle of Higher Education*.
The core essence

within those headline stories
was the new supercomputer knowledge
of how and why
the supercomputer scientist
must parallel process **across** processors
that encircled the globe
in the way the internet does.
That experimental discovery
is embodied in **multifunctional** computers
and in all supercomputers.
It should be noted that
the supercomputers of the past
were not used the way
the supercomputers of the present
are used today.
After World War Two
and after 1946,
programmable supercomputers
were mainly used to solve
textbook problems,
such as ordinary differential equations

from calculus textbooks.
Seven decades later, the supercomputer
that is powered by ten million
six hundred and forty-nine thousand
six hundred [10,649,600]
commodity-off-the-shelf processors
is used to solve global problems,
such as high-resolution, long-running
general circulation models
that are used to **foresee**
otherwise **unforeseeable** climate changes.

12.1.2 China's Entry into Supercomputing

Back in 2006, China unveiled its plan
to invest 112 billion dollars
in scientific research
and to do so by 2020.
One of the products
from that ambitious quest
was the world's fastest supercomputer
that was made in China.

That fastest supercomputer
was powered by parallel processing
across ten million
six hundred and forty-nine thousand
six hundred [10,649,600]
commodity-off-the-shelf processors.
By 2020, China hopes that 60 percent
of its economic growth
will arise from its investment
in high technology.
The uncharted fields of knowledge
is the new land
to be explored and colonized.
That new land is explored
the way **Mungo Park** explored
the River Niger of West Africa.
The exploration of **Mungo Park**
opened the door
for Great Britain
to colonize my country of birth, Nigeria.
I'm the **Mungo Park**

of the supercomputer world
that was searching
for the fastest computation, **ever.**
I was searching
for the **new supercomputer**
that computes in parallel,
instead of in sequence.
In the new land
of parallel processing supercomputers
you're either a colonizer or the colonized.
China
intends to become a colonizer
in the frontier of science.
Africa is still contented
with being colonized
in the frontier of technology.
This is the reason the United States
has raised an alarm cry
over the alarming resources
that China is investing

to become a colonizer
in the frontier of the supercomputer.

12.1.3 How I Was Mocked By Seymour Cray

The answers to the toughest questions
in extreme-scale computational physics
were not in the physics textbooks
of the 1980s and earlier.
I discovered the answers
to those tough questions
and discovered them **across**
a new internet
that is a global network of
64 binary thousand
commodity-off-the-shelf processors,
or across a new internet
that is a global network of
as many identical computers.
My supercomputer discoveries

were not taught
in the classrooms of the two decades
of the 1970s and '80s.
My experimental discovery
of massively parallel processing
opened the door to a revolution, namely,
computers and supercomputers
that could solve many problems
at once, or in parallel.
Back in the 1980s, both **Gene Amdahl**
of **Amdahl's Law** fame
and **Seymour Cray**
who pioneered vector processing technology
for supercomputers
were the strongest **opponents**
of incorporating
parallel processing technology
into the modern supercomputer.
Seymour Cray is best remembered

for ridiculing and rejecting
the massively parallel processing

supercomputer
and for mocking the technology
in his famous quote.
Seymour Cray joked:
[quote]
"If you were plowing a field,
which would you rather use:
Two strong oxen or 1,024 chickens?"
[unquote]
In my experimental discovery
of the massively parallel processing
supercomputer
that occurred on the Fourth of July 1989
I used 65,536 chickens,
instead of one strong oxen.
I was the strongest **proponent**
of parallel processing.
For that reason,
I was the lone wolf programmer
of the most massively parallel
supercomputer ever built,

as of the 1980s.
Seymour Cray
designed more vector processing
supercomputers
than anyone else designed.
As the most experienced
supercomputer scientist that he was,
the supercomputer industry
listened to **Seymour Cray**,
not to me, **Philip Emeagwali**.
My experimental discovery
of how to solve a million **or a billion**
computation-intensive problems
and how to solve them **at once,**
or in parallel,
made the news headlines because
I proved that the computer
powered by only one processor
can do whatever the supercomputer
powered by ten million
six hundred and forty-nine thousand

six hundred [10,649,600]
commodity-off-the-shelf processors
can do, if and only if,
the computer has 30,000 years
to compute
what the supercomputer computed
in only one day.
For that experimental discovery,
it is often said that
Gene Amdahl is to
sequential processing supercomputers
what **Seymour Cray** is to
vector processing supercomputers
and what **Philip Emeagwali** is to
parallel processing supercomputers.

12.1.4 Please Allow Me to Introduce Myself

Please allow me to introduce myself.
Who is **Philip Emeagwali**?

236

I was the only fulltime programmer
of the most massively parallel
supercomputer
of the 1980s.
I visualized that massively parallel
supercomputer
as a small copy of the Internet.
I experimentally discovered
that massively parallel supercomputer
to be a new internet
and a global network of
sixty-five thousand
five hundred and thirty-six [65,536]
processors
that I visualized
as encircling a globe
that I also visualized
in a sixteen-dimensional universe.
More importantly, the reason I was the only
full time massively parallel processing
programmer of the 1980s

was because I was the only
supercomputer scientist
that had the in-depth knowledge
across the frontiers of mathematics,
across the frontiers of physics,
and **across** the frontiers of computer
science.
That interdisciplinary knowledge
was needed to program
that massively parallel supercomputer
and needed to give research lectures
to research mathematicians,
research physicists,
and research supercomputer scientists.
I was the only
supercomputer scientist,
that I know of,
that was trained for sixteen years,
onward of June 20, 1974.
I was the only
and the first supercomputer scientist

that gave **full breath** public lectures
on massively parallel processing
and gave those lectures
in the early 1980s.
In the early 1990s,
I was appointed
as the Distinguished Speaker
from the Association
for Computing Machinery
to American university computer science departments
and I lectured on parallel processing
supercomputing.
The Association for Computing Machinery
is the premier society
for computing professionals.
In the early 1990s,
I was also appointed
as the Distinguished Visitor
from the Computer Society
of the Institute of Electrical

and Electronics Engineers, or IEEE,
to American university computer science
departments.
The Institute of Electrical
and Electronics Engineers
is the world's largest technical society.
The video tapes of my lecture series
on parallel processing
are posted at **emeagwali dot com**.
I lectured on how I
experimentally discovered
how to program
massively parallel processing
supercomputers
that were powered by
two-to-power sixteen processors
and how to use my experimental discovery
to solve the toughest problems in physics
that were previously **impossible**
to solve.
I experimentally discovered

that the massively parallel processing
supercomputer
can solve problems
in extreme-scale computational physics
that the sequential processing computer
cannot solve.
I experimentally discovered
that the massively parallel processing
technology
scales linearly
from one processor
to one billion processors,
and beyond one **billion**.
I experimentally discovered
that the massively parallel processing
technology
connects many processors
into a unified supercomputer,
whether its sixty-four binary thousand
processors
or sixty-four binary **billion**

processors.
That scalable
massively parallel processing supercomputer
of the 1980s
that was invented
as today's modern supercomputer
will, hopefully, be re-invented
for tomorrow's planetary supercomputer.

12.1.5 I Was the Underdog of Supercomputing

My first entry into the
unexplored territory
of massively parallel supercomputing
felt like a **David** versus **Goliath** battle.
I—**Philip Emeagwali**—was the **David**
and the **proponent**
of massively parallel processing
supercomputers.
Seymour Cray and **Gene Amdahl**
were the **Goliaths**

and the **proponents**
of scalar and vector processing
supercomputers, respectively.
The reason the likes of **Seymour Cray**
and **Gene Amdahl**
believed that it will be physically impossible
for me to massively parallel process
and do so **across**
an ensemble of 64 binary thousand
commodity-off-the-shelf processors
was because they were trained
for only six years.
Seymour Cray and **Gene Amdahl**
were only trained
on how to sequentially compute
and compute
with only one processor.
The reason the pioneers of
sequential processing supercomputing
of the 1950s and ‘60s
and those of vector processing

supercomputing
of the 1970s and '80s
argued that parallel processing
will forever remain a huge waste
of everybody's time
was because they lacked
the sixteen years of **mathematical maturity**
that I acquired, onward of March 25, 1974.
My contributions to algebra, calculus,
and computational mathematics
was the cover story
of top mathematics publications
that are read by research mathematicians.
Seymour Cray and **Gene Amdahl**
needed to fully understand
the parallel processing
supercomputer technology
and can only do so by, first, understanding
the extreme-scale computational science
behind the fastest supercomputing.
It's impossible for **Seymour Cray**

or **Gene Amdahl** to understand
the most advanced expressions
in calculus—that is a subset
of massively parallel processing—without, foremost,
having a decade and half
of specialized training on how to solve
initial-boundary value problems
that are governed by
a system of coupled, non-linear,
time-dependent, and state-of-the-art
partial differential equations
of modern calculus,
called Emeagwali's Equations.
In an abstract lecture on advanced calculus
and extreme-scale algebra
that I delivered
on July 8, 1991, in Washington,
District of Columbia, United States,
I told mathematicians attending
the International Congress

of Industrial and Applied Mathematics,
the following:
"As a research mathematician
and as a research physicist,
I always knew the fact
that the scientific discoverer
discovered a truth,
whereas the inventor
of a partial differential equation
formulated possibilities."
A computer scientist
that only trained with computers
that only used one processor
will not understand
the partial differential equations
and, therefore, will not understand
how to massively parallel process
and how to do so **across**
a new internet
that is a global network of
64 binary thousand

commodity-off-the-shelf processors.
So, my combined knowledge of physics,
calculus, algebra,
and massively parallel processing
was greater than the combined knowledge of
Seymour Cray and **Gene Amdahl**
that were only trained with computers
that used only one processor
that was not a member
of an ensemble of processors.
That gap in scientific knowledge
is evident by watching
and doing a side-by-side,
videotape-by-videotape
comparisons of the scientific lectures
of **Seymour Cray**,
Gene Amdahl, and myself,
Philip Emeagwali.
There was no shortcut
that could have enabled **Seymour Cray** or
Gene Amdahl

to understand in six years
what took me sixteen years to understand.
It's as physically impossible
as a six-year old
fighting a sixteen-year-old **Mohammed Ali**
for the future world heavy weight
boxing championship.
Two thousand three hundred years ago,
a young prince asked **Euclid**
—the father of geometry—
for a short cut to geometry.
Euclid said to the young prince:
"There's no royal road to geometry."

12.1.6 Naming Convention For My New Internet

I was the first internet scientist
that articulated
how he experimentally discovered
massively parallel processing

and discovered it
as the technology
that makes modern computers **faster**
and makes the new supercomputer
the **fastest**
and invented
how and why to use
that **new supercomputer knowledge**
to build a **new supercomputer**
that encircled the globe
in the way the internet does.
In the 1980s, I articulated
how I named each of my
65,536 commodity processors.
And I articulated
how I commanded each processor
to send and receive emails
and do so
to and from the **other**
sixty-five thousand
five hundred and thirty-five [65,53**5**]

processors.
This technical aspect of my contribution
to the experimental discovery
of parallel processing
was lost to the lay public, in part,
because it involved abstract and dense
mathematical knowledge of calculus,
algebra, topology, and graph theory.
My system of coupled, non-linear,
time-dependent, and state-of-the-art
partial differential equations
of modern mathematics,
called Emeagwali's Equations,
were developed only for
research computational mathematicians.
It is impossible for the lay person
to understand partial differential equations.
How to **accurately** solve
partial differential equations
could only be understood
by a few dozen people

that were actually experimenting with
massively parallel processing
supercomputers.
Over the years, I learned that conversations
about parallel processing
is a **party spoiler**,
even amongst supercomputer scientists
that only believe in vector processing
supercomputing.

12.1.7 Naming Computers Across My New Internet

I began programming
sequential processing supercomputers
on Thursday June 20, 1974 at age 19
at 1800 SW Campus Way,
Corvallis, Oregon.
That sequential processing supercomputer
was the world's fastest in the mid-1960s.
By definition, a sequential processing
supercomputer

is powered by only one powerful processor.
Therefore, it was not necessary
for me to name that sole processor.
My unique naming
of my 65,536
commodity-off-the-shelf processors
was the abstract elephant
in the supercomputer center.
Those 65,536 unique names
were the as many uninvited guests
to the **unexplored territory**
of the massively parallel supercomputer.
That lack of understanding
of how to uniquely name
those processors
added weight to the saying
that parallel processing
is a huge waste of everybody's time.
The June 14, 1976 issue
of *Computer World*,
the flagship publication
of the world of computing
described parallel processing

as a [quote-unquote] “waste of time”
and ridiculed it
as “large” and “clumsy.”
Truly, it was a waste of time
to attempt to parallel process **across**
65,536 **nameless** processors.
Because it seemed **impossible**
to uniquely name
those 65,536 processors
no textbook in extreme-scale
computational physics
of the 1970s
attempted to describe that elephant,
namely, the unique set of
65,536 unique names
for the as many **commodity-of-the-shelf**
processors.
Those unique names—that comprised of
one binary **million** zeroes
and ones—was an **abstract**
and an **invisible** elephant
in the world of massively parallel processing
supercomputers.

Yet, to the aspiring supercomputer wizard
of the 1970s
that binary naming is ever present
and was concrete
and was as numerous
as each of my 65,536
commodity-off-the-shelf processors.
Each of those commodity processors
had its own separate operating system
and memory.
In the 1970s and '80s,
I read pessimistic articles
about parallel processing.
But I stayed on the course of
massively parallel processing
after reading hopeful articles
on the potential benefits
of fast parallel processing supercomputers.
One such positive article
was in the May 8, 1987 issue
of *The Chronicle of Higher Education,*
the flagship newspaper

that presents news to universities.
The article was written by
computer and information technology writer
Judith Axler Turner.
The article was titled:
[quote]
"Some Hail 'Computational Science'
as Biggest Advance Since Newton, Galileo."
[unquote]
When I read that article,
shortly after May 8, 1987,
in Laramie, Wyoming,
I deduced that my experimental discovery
of how to parallel process
and do so **across** an ensemble of
65,536 commodity processors
will become the biggest advance
in computational science.
Fast forward three years,
the same technology writer,
Judith Axler Turner,
wrote in the June 27, 1990 issue of
The Chronicle of Higher Education

that I
[quote]
"took on an enormously difficult problem...
solved it alone,
has won computation's top prize,
captured in the past
only by seasoned research teams.”
[unquote]
Those **seasoned research teams**
comprised of up to two dozen
supercomputer scientists
that were supported
by tens of millions of dollars
in US government grants.
That *Chronicle of Higher Education* article
continued:
[quote]
“If his program can squeeze out
a few more percentage points,
it will help decrease
U.S. reliance on foreign oil."
[unquote]

256

For my world's **fastest**
petroleum reservoir calculations
that made the news headlines
that were highlighted
in the June 20, 1990 issue
of the *Wall Street Journal*,
I had to uniquely name
all my sixty-five thousand
five hundred and thirty-six [65,536]
commodity-off-the-shelf processors,
and **correspondingly**
name as many problems
in extreme-scale computational physics
and name them
with a **one-to-one** correspondence
between the problems
and my as many processors.
Geometrically, I saw my **small copy**
of the Internet
as a global network of
two-to-power sixteen,

or sixty-five thousand
five hundred and thirty-six [65,536],
commodity-off-the-shelf processors
in which each processor
had a **one-to-one** correspondence
to the as many vertices of a cube
in sixteen dimensional hyperspace.
I visualized that cube
as tightly circumscribed by a globe
and I visualized the vertices
of that cube
to be on the surface of that globe
and to be equal distances **apart**.
A young Nigerian asked me:
"How can I become
a supercomputer wizard
like you, Philip Emeagwali?"
I explained that he can become
a supercomputer wizard
by experimentally discovering
that the **impossible-to-compute** is,
in fact, **possible-to-compute**
and experimentally discovering it **across**

a **never-before-seen** quantum computer.
I explained that
it took me sixteen years
onward of June 20, 1974
to experimentally discover
how and why
massively parallel processing **across**
a new internet
makes the computer **faster**
and makes the supercomputer **fastest**
and how to use
that **new supercomputer knowledge**
to build a **new supercomputer**
that encircled the globe
in the way the internet does.
It took me sixteen years
of programming sixteen supercomputers,
each powered by
up to two-to-power sixteen
commodity processors
to experimentally discover
how and why parallel processing
makes the supercomputer **super**.

It took me sixteen years
to become
the **African supercomputer wizard**
in the United States
that won the top prize in supercomputing.
It takes time
to make an invention
that is noteworthy.
I failed sixteen times
in sixteen years
before I discovered how to name
my processors and problems.
I used the **binary reflected code**
to generate my unique sixteen-bit long
binary identification names
that I must generate
as the **precondition**
to harnessing the power of
my two-to-power sixteen
commodity processors.
Yet, assigning a computational
fluid dynamics code
in computational physics

to a processor
within a small copy of the Internet
was not as simple as emailing
the computational fluid dynamics code
and emailing its initial and boundary data
to each processor
that shared its corresponding
decimal address.
Technically speaking,
emailing to decimal addresses
still solves
the computational fluid dynamics problem.
But it will merely solve the
computation-intensive
initial-boundary value problem
of computational physics
and solve it
at the everyday speed of the computer,
not at the newsworthy speed
of the supercomputer,
or at the supercomputer speed up
of 180 years in one day
that became my signature discovery.

When I began to experimentally
program supercomputers
on June 20, 1974, in Corvallis, Oregon,
I did not know that
I will invent
how to massively parallel process **across**
a small copy of the Internet
that is a global network of
64 binary thousand
commodity-off-the-shelf processors.
That new internet
was a small copy
of a never-before-understood Internet,
that had only 65,536 processors
around a globe
instead of billions of computers
around a globe.
I didn't know the answer.
I didn't know what I would invent.
If I knew the answer
I wouldn't be solving the problem.
And if some else knew the answer
before I did

then my answer
would not have made
the news headlines in 1989.
It's true that
I had to hit my mark and run.
It's true that
I did not follow all the rules.
It's true that
I re-wrote some rules.

12.2 How I Invented a New Supercomputer

12.2.1 The Modern Supercomputer

Back in the 1980s,
my homes and offices
—in Silver Spring, Maryland,
Casper, Wyoming,
and Laramie, Wyoming—were littered
with my drawings and my blueprints
of the prototypes of
two new massively parallel processing

supercomputers
that I invented
and that I constructively reduced to practice.
Those two new supercomputers
that encircled the globe
in the way the internet does
were named **Philip Emeagwali** HyperBall
Supercomputer
and **Philip Emeagwali** Cosmic Ball
Supercomputer.
I visualized my never-before-seen
Cosmic Ball Supercomputer
as a small copy of the Internet
that is located on the North Pole.
That Cosmic Ball Supercomputer
was an ensemble of processors
and was not a new computer *per se*.
That Cosmic Ball Supercomputer
was a global network of processors
and was a new supercomputer *de facto*.
That Cosmic Ball Supercomputer
was a new internet *de facto*
because it tightly encircled the globe

in the way the internet does.
At its supercomputing core,
my Cosmic Ball Supercomputer
comprised of commodity
off-the-shelf processors,
or identical processors,
that were mass marketed
for everyday computers.
For that reason, my Cosmic Supercomputer
was **independent** of processor technology.
The Cosmic Supercomputer
could be continuously updated
with the newest commodity
off-the-shelf processors.

12.2.2 The Fastest Communication Across My New Internet

To send and receive emails
and do so **synchronously**
and **across**

sixty-five thousand
five hundred and thirty-six [65,536]
commodity off-the-shelf processors
(or across identical computers)
was like looking at God
in the face.
To experimentally discover
the fastest computations
demanded that I **optimize** my use
of every nut and bolt
inside my new, faster supercomputer
as well as explore every hidden corner
within my new supercomputer.
Sending and receiving
64 binary thousand emails
and sending them **at once**,
instead of sending them **one-by-one**
is at the granite core
of how I invented
the parallel processing technology
that makes modern computers **faster**
and makes the new supercomputer
the **fastest**

and is at the granite core
of my invention
of how and why to use
that **new supercomputer knowledge**
to build a **new supercomputer**.
I visualized the gaps
in the global network of
processors
to be filled.
I visualized my email messages
as **stitching** the 65,536 pieces
back together.
I discovered that
it's a necessary condition
that the fastest floating-point arithmetical
computation
must be preceded by
the fastest email communication.
For the fastest emailing **across**
my small internet,
I visualized my complete
petroleum reservoir
as an ensemble of sixty-five thousand

five hundred and thirty-six [65,536]
petroleum reservoirs.
I used my message-passing
computational fluid dynamics code
to re-assemble
the petroleum reservoir simulation
for each small petroleum reservoir
and put them back together
as my original petroleum reservoir.
In my massively parallel supercomputer
coding,
I assigned the decimal email address
[quote unquote] "E"
to one of my 64 binary thousand
petroleum reservoirs.
I assigned that email address
as the unique string
of sixteen zeroes and ones
that's the binary identification number
of a processor
(or a computer).
I tasked each processor
(or each computer)

to only send and receive email messages
and do so to and from
the processors
(or the computers)
that corresponded to
the two petroleum reservoirs
that are adjacent to it
and with the email address
[quote unquote] “E minus one”
and the email address
[quote unquote] “E plus one.”
On the small internet
that I invented
and that I visualized
as a global network of
64 binary thousand
processors
(or as a global network of
as many computers),
the processor
(or the computer) named “E”
may not be directly connected
to either the processor

(or the computer) named “E minus one”
and/or
the processor
(or the computer) named “E plus one.”
My **one-to-one** correspondence
between my 64 binary thousand
processors
(or the as many computers)
and my 64 binary thousand
petroleum reservoir models
(or the as many blocks of oilfields),
forced my email messages
that had to be delivered
to distant processors
(or distant computers)
to be delivered
along the shortest possible route.
Doing so enabled me to discover
the fastest speeds
in email communication
across fast interconnect paths
that I executed, on the Fourth of July
of 1989,

and executed **across**
my new internet
that is a global network of
65,536 commodity-off-the-shelf processors.
Doing so enabled me to
experimentally discover
the fastest speeds
in arithmetical computation
that I executed, on the Fourth of July
of 1989,
and executed **across**
my massively parallel processing
supercomputer
that's *de facto* a **small copy** of the Internet.
Doing so was how I
experimentally discovered
the shortest **time-to-solution**
and the new fastest supercomputer.
Not doing so
makes as much sense
as a letter mailed to an address
that's just one mile away
to travel around the world

before arriving one mile away.

On the Fourth of July
of 1989, the US Independence Day,
I sent my emails—that each contained
my computational fluid dynamics code—
and sent them
through the shortest paths
which made it possible
for me to experimentally discover
how solving a million problems **at once**,
or in parallel,
makes modern computers **faster**
and makes the new supercomputer
the **fastest**
and for me to **experimentally invent**
how to use
that **new supercomputer knowledge**
to build a **new supercomputer**
that encircled the globe
in the way the Internet does.
Back in the 1970s and ‘80s,
it was often written that

parallel processing
is a huge waste of everybody's time.
In the 1980s,
I was the sole fulltime programmer
of the most massively
parallel processing machine
ever built.
That parallel processing machine
that was rejected
by every supercomputer programmer,
except I, is the first **precursor**
to today's modern supercomputer.
That massively parallel processing machine
was the most complex computing engine
ever imagined.
In the 1970s,
that parallel processing supercomputer
was as futuristic
as the quantum computer
was in the 1980s.
As its lone wolf programmer,
my two grand challenge questions
were these:

First, where does my massively parallel
processing machine
draw its fastest computing speed
for solving the toughest problems
in computational mathematics?
Second, where does my massively
parallel processing machine
draw its fastest email communication speed
for communicating the toughest problems
in computational physics?
On the Fourth of July 1989,
the US Independence Day,
I experimentally discovered the answers
to both grand challenge questions
of mathematics and physics.
Those experimental discoveries
of how to massively parallel process
across an ensemble of processors
enabled me to forge a path
to the farthest frontier of computing
that is the modern supercomputer.

12.2.3 Emailing Across My New Internet

Back in the 1970s and '80s,
parallel processing was ridiculed
as a beautiful theory
that lacks experimental confirmation.
And my quest for the fastest
massively parallel processing computation
was like searching for a black box
in a dark sixteen-dimensional universe.
At some point, I asked myself:
"What do you do
when your processors
are not directly connected?
What do you do
when you could not send
your email messages
directly
to a processor?"
For email communication
between processors
that were not connected directly,
my emails were **stored-and-forwarded**,

or **hopped** through intermediate
interconnects
and to my 65,536 commodity processors.
To perform the fastest computations
and do so across any internet
demands that the shortest email paths
be followed.
I performed the fastest computations
by following the shortest path
and following it
when I sent and received emails
to and from
one processor
to another.
I wrote my email message passing code
to email the petroleum reservoir model
that I named “R sub I” [Ri]
and email it
to a one-to-one-corresponded
processor
that I named “C sub I” [Ci],
where the subscript “eye” [i]
is equal to

or greater than one
and equal to or less than
sixty-five thousand
five hundred and thirty-six [65,536].
My 64 binary thousand messages
were emailed and received
simultaneously.
That's how I massively parallel processed
by communicating **in parallel**
or sending sixty-five thousand
five hundred and thirty-six [65,536]
emails **at once**.
That's how I massively parallel processed
by computing **in parallel**
and doing so
to reduce my **time-to-solution**
from sixty-five thousand
five hundred and thirty-six [65,536] days,
or 180 years, to just one day.
My experimental discovery
of massively parallel processing
opened the door
to the modern supercomputer

that parallel processes across
over ten million processors.

12.2.4 Naming My New Internet

I discovered
how to correctly **codify**
the Second Law of Motion
of physics
and **codify** it
into the partial differential equations
of calculus.
Yet, I am more than
a research mathematical physicist.
I am a research parallel processing
computational physicist
and a research internet scientist.
My fastest, massively parallel processed
extreme-scaled
computational fluid dynamics codes
of the 1970s and '80s
that made the news headlines
were about transporting codes,

data, and answers
and transporting them **across**
my small copy of the Internet
that was my global network of
65,536
processors.
In my ancestral hometown
of Onitsha (Nigeria),
the fastest, extreme-scaled
computational fluid dynamics code
—such as **petroleum reservoir simulation**—
is more relevant
if it helps
to **recover** otherwise **unrecoverable**
crude oil and natural gas
and do so
from the oilfields
in the Niger-Delta region of Nigeria.
In my adopted hometown of Washington,
District of Columbia, United States,
the fastest, extreme-scaled
computational fluid dynamics code
—such as a **general circulation model**—

is more relevant if it is used to
foresee previously **unforeseeable**
global warming.
To the person in Abuja (Nigeria),
the fastest, massively parallel processing
supercomputer
is more relevant
if it contributes to shaping cities like Abuja.
To the African economist,
the fastest supercomputers in Africa
are more relevant
if they are used to increase
economic growth by **discovering**
otherwise **elusive** crude oil and natural gas
and then using that
new petroleum revenue
to alleviate poverty in **Uganda**
and **Cameroun**.
Several subfields of research
emerged from the unknown world of
massively parallel supercomputing.
They emerged
between the mountains of calculations

and the oceans of processors.
In the world of physics alone,
massively parallel supercomputing
opened the doors
to extreme-scale mathematical computations
in fluid dynamics, climate modeling,
complex and turbulent systems, cosmology,
molecular dynamics, material science and
engineering, nanotechnology, **plasma
physics**, accelerator physics, condensed
matter physics, chemical physics, quantum
physics, astrophysics, high-energy physics,
nuclear physics, and theoretical physics.
Therefore, it should not come as a surprise
that **nine** in ten supercomputer cycles
were consumed by the physics community.

12.2.5 More Information

I'm **Philip Emeagwali**.
I've posted at emeagwali dot com

video-taped lectures
and lecture notes
on how I experimentally discovered
how and why parallel processing
makes modern computers **faster**
and makes the new supercomputer
the **fastest**
and how I **experimentally invented**
how to use
that **new supercomputer knowledge**
to build a **new supercomputer**
that encircled the globe
in the way the internet does.
You can reach me at
emeagwali dot com.

12.2.6 A New Era in Computing

The **fastest** supercomputer
costs the budget of a small nation
and it is purchased

because the **fastest** supercomputer
gives meaning to life,
and because the **fastest** supercomputer
makes the world a better place,
and because the **fastest** supercomputer
makes humanity more knowledgeable
and because the **fastest** supercomputer
of today
will become the computer of tomorrow.
The scalar processing supercomputer
helped the first man
that traveled to the moon
to return safely from the moon.
The vector processing supercomputer
helped man fly faster
and helped the first woman
that traveled into outer space
to return safely from outer space.
The parallel processing supercomputer
will help the first humans
that will travel to the planet Mars
to return safely
from the planet Mars.

And **faster** supercomputers
is where science fiction
will become non-fiction.
The **fastest** supercomputer
is where humanity's future
takes shape.
Parallel processing
has taken the computer
into a new era.