Slide 1

Computing, Approximately

Ravi Nair and Dan Prener
IBM Thomas J. Watson Research Center

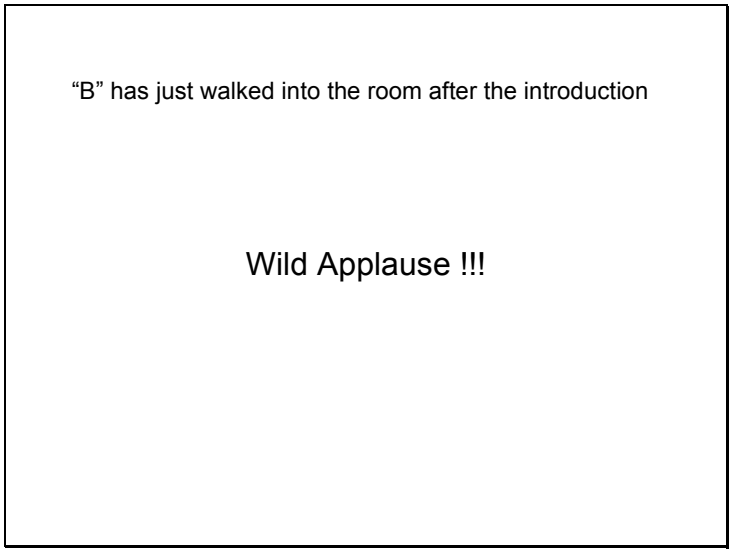
Slide 2

This talk is a plagiarized version of a speech given by a leading candidate “B” in the American presidential elections

The presentation depends on the participation of the audience

So please follow the directions on the screen

Slide 3



Thank you, ASPLOS.

You know, they said this day would never come.

They said our sights were set too high.

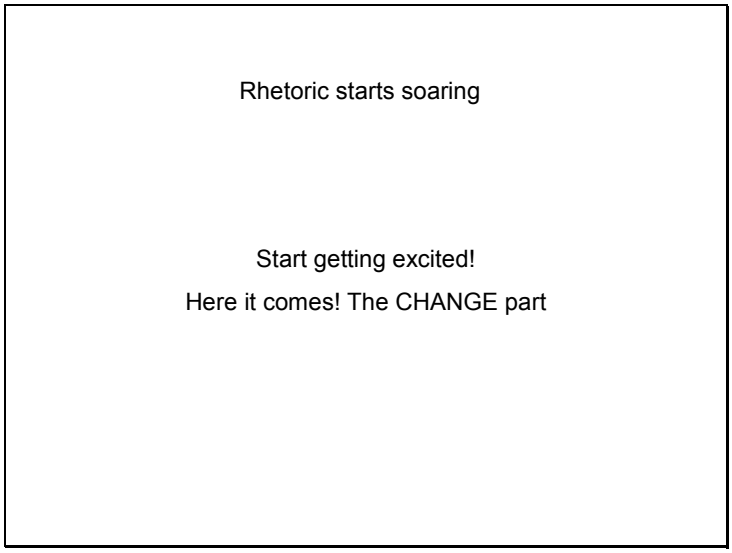
They said this conference was too prototypical; too timid to ever consider an idea such as this.

You know, we have been told for many years that computers must be precise, that architectures must be precisely defined.

We have been led to believe that designs must be verified to be precise, that implementations must be tested to be precise, and that programs must be precise.

(Slide)

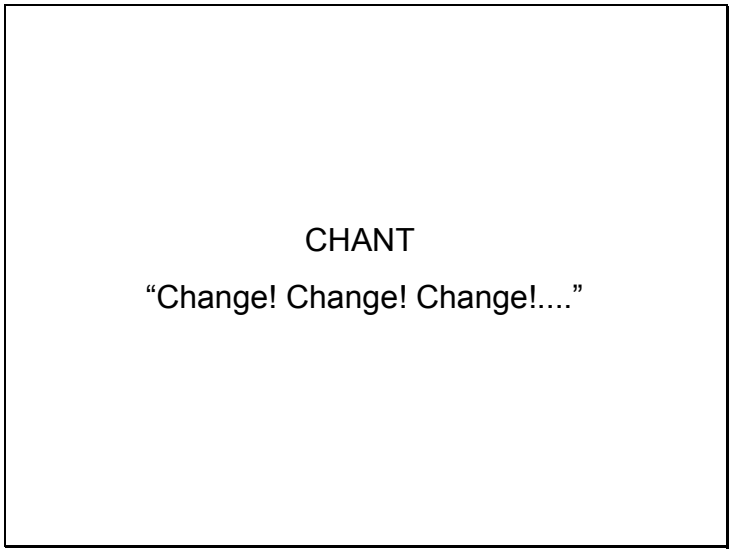
Slide 4



But, on this March evening - at this defining moment in history – I urge you to search your soul and ask, Is this preciseness important when it costs so much in power and area to implement?
Is this preciseness important when 3 digits of accuracy is perfectly acceptable as the value of pi in your application?
Is this preciseness important when the program itself produces only an approximate solution to the real problem?

We incur enormous increases in path length in order to achieve this meaningless preciseness.
We incur enormous increases in energy performing useless computations in order to achieve this meaningless preciseness.
We cannot afford it anymore.
It is time to turn the page.
(Slide)

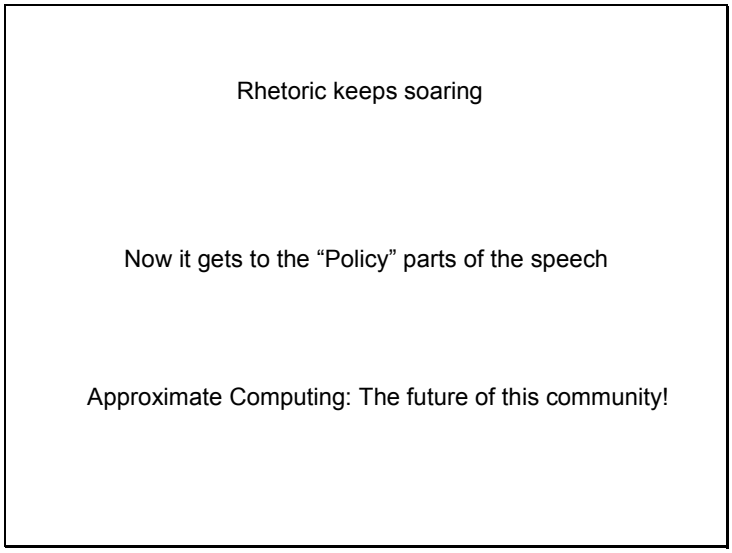
Slide 5



It is time to stand for CHANGE.

(Slide)

Slide 6

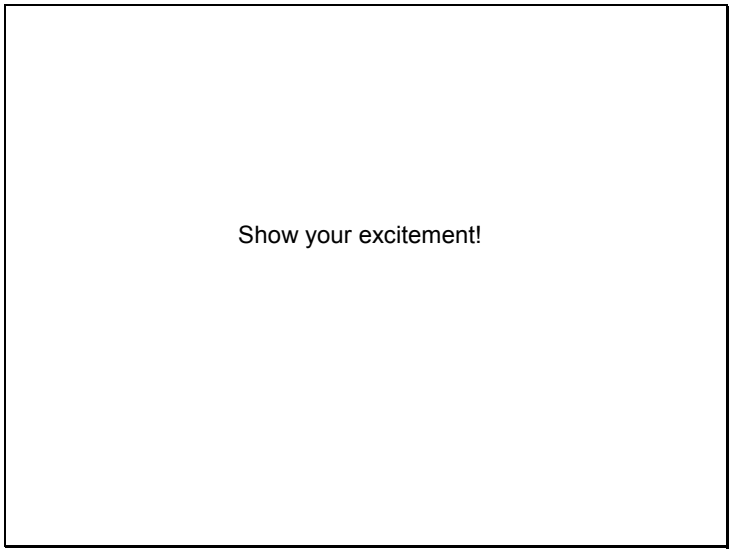


It is time to go beyond the bureaucratic rigidity of the incumbent style of computing and adopt a new style of computing – approximate computing.
Approximate computing – it is computing for all of us, not computing just for the top 2% of the scientific community.
Approximate computing – it is computing that will help you make decisions, rather than computing that helps you just produce numbers.
Approximate computing – it is computing that conserves energy – by producing reasonable results, rather than results that are “exact” – and exact, by whose measure of exactness?

There are many, many situations where intelligent approximation can provide fast, reasonable results with lower energy consumption.
Take the case of typical table-driven techniques. Imagine looking up a table for a match on a certain set of parameters. Today, if the computer didn’t find a match, it throws its hands up in the air and reports back to the user – sorry, I cannot help you.
Suppose instead the computer found some near matches. And returned the value for a near match. Or better still, suppose the computer interpolated

between these matches to provide an estimated result back to the user. Wouldn't that be useful? Wouldn't that save time and energy?
Time and energy which the user would have spent to refine his query and send it back?
Time and energy that would be used to compute the "exact" value for that set of parameters?
That is approximate computing. That is intelligent computing.
Take another example. Green computing. Wouldn't it be nice if our algorithms and systems, when power-constrained, operate at a more-efficient level of precision – whenever it is appropriate to do so?
For example, switch to single-precision, instead of double precision?
For example, relax data coherence requirements, whenever it can be tolerated?
For example, suggest to the processor that the number of iterations of a convergence loop is flexible and may be reduced?
For example, allow a variable to use a default value if it misses in the cache?
(Slide)

Slide 7



Here's another domain: Nanotechnology.
As devices get smaller, they get more unreliable.
So what are we likely to do?
We are likely to add redundancy mechanisms – at great cost in energy and real-estate – to make computers in this new regime behave exactly like the computers we are familiar with.
Instead, we should target such new technology for applications that use algorithms, programming models and architecture that are aware of the inherent unreliability.
Not skirt around the unreliability.
This is how organic systems work. This is how the brain works.
Should all computers be modeled after the human brain? Of course not!
The point here is that a lot of interesting problems that we want to solve using computers need only approximate answers – similar to what the human brain provides.
And we all know how power-efficient the human brain is.

There is an unprecedented amount of data being produced in the world today.
Like the data from the millions of event generators, the millions of sensors all over the world.
Yet, cost and energy considerations are limiting the compute capability needed to process and analyze this data.
The use of power-efficient, approximate computing techniques to process this information is our only hope.
Hope.
(Slide)

Slide 8

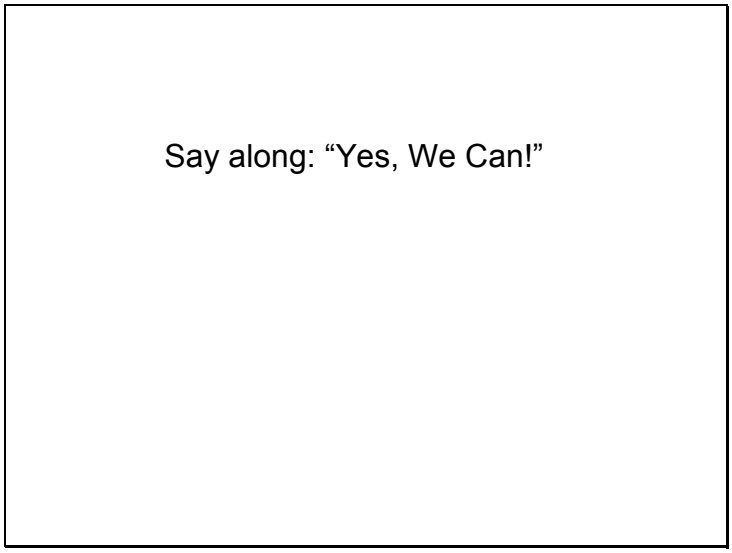
I can see the skepticism when I talk about hope.

We know that hope is not blind optimism. It's not ignoring the enormity of the task ahead or the roadblocks that stand in our path.

Hope is that thing inside us that insists, despite all evidence to the contrary, that something better awaits us if we have the courage to reach for it, and to work for it, and to fight for it.

But we have to do it together. (Slide)

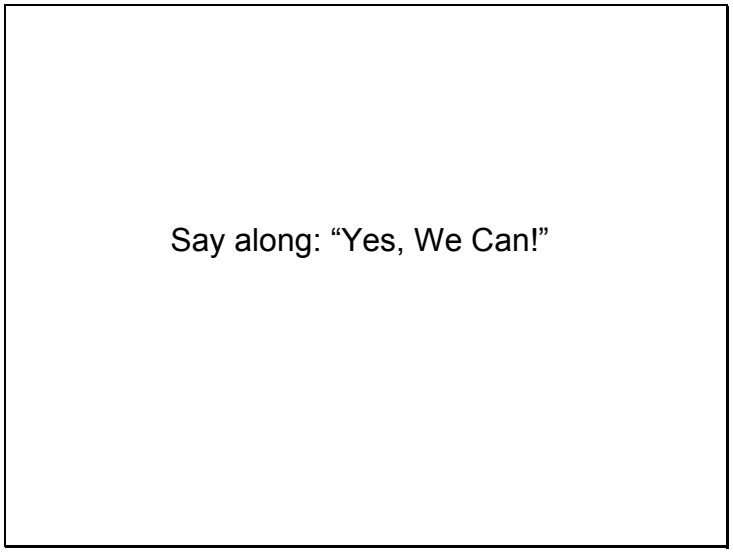
Slide 9



And YES, WE CAN.

Yes, we can begin designing new models for algorithms, for architectures, and for implementations to support approximate computing. (Slide)

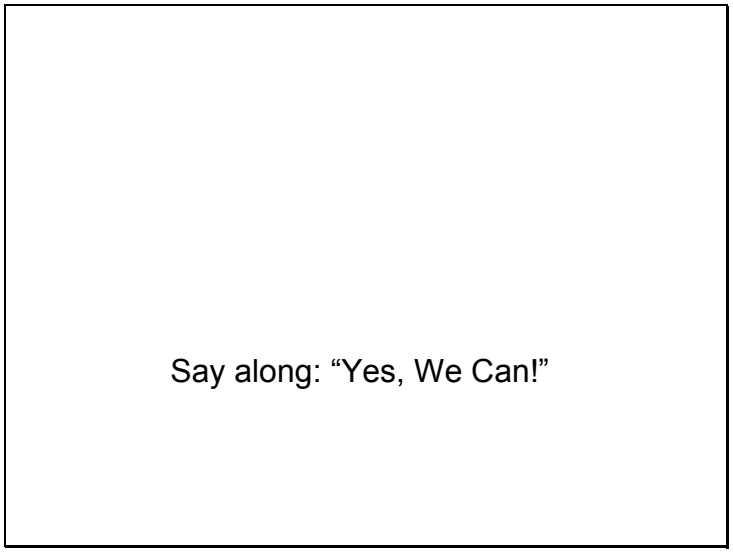
Slide 10



YES, WE CAN.

And yes we can help bridge this widening gap between the generation of data and the energy-efficient processing of this data. (Slide)

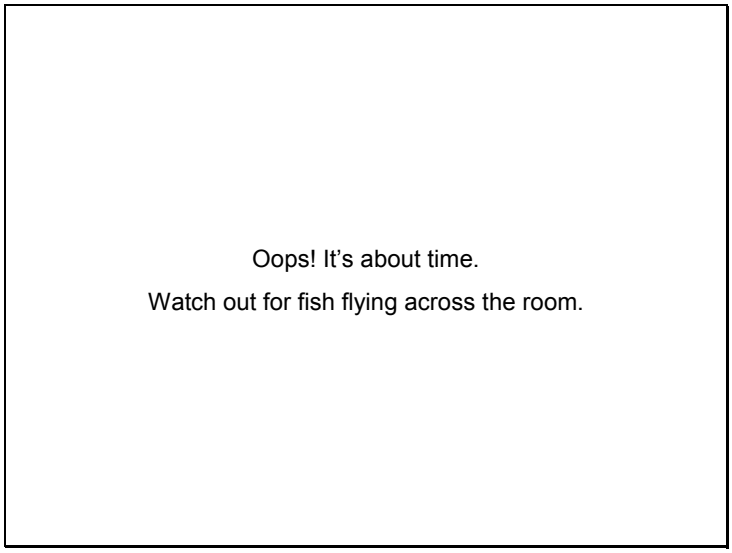
Slide 11



YES, WE CAN.

(Slide)

Slide 12



Years from now, you'll look back and you'll say that this was the day - this was the place, Seattle - where the ASPLOS community remembered what it means to hope.
That is the message we can carry to Redmond and to Santa Clara, to Austin and to Poughkeepsie, and beyond.
That together, as ordinary scientists we can achieve extraordinary things.
Because, at this moment, we are not hardware architects or software architects, we are the computer architects of the world, and we are ready to believe again.
(Slide)

Slide 13

Thank you, ASPLOS!