

---

Copyright © 2008 Ajit R. Jadhav  
All rights reserved  
<http://www.JadhavResearch.info>

# High Performance Cluster-Based Computing for Engineering Design

---

**Ajit R. Jadhav**

B.E. (Met.), M.Tech. (Ind. Met.), D.A.C., Ph. D. (Mech. Engg.) (thesis submitted)  
<http://www.JadhavResearch.info>

---

SAE Conference on Advances in Automotive Design and Styling  
ARAI, Pune, India  
18<sup>th</sup> February, 2008

---

# A “Black Ribbon” Slide

- I am competent, and yet,
- I have not been getting a suitable job or a contract in
  - Software Development in
    - CAE
    - Computer Graphics
    - 3D Imaging
    - Similar and related fields (e.g. computational mechanics)
    - Or, even in application engineering for CAE packages
- For 6.5 years out of the last 7 years...
  
- Find out turnover of industries in and around Pune
- And of American companies in US and in India!
  
- I think the black ribbon is well justified...

---

# Outline

- CAE in automobile engineering
- What kind of computational power is needed now? In near future?
- Clusters: A bird's eye view
  - Cluster = Nodes + Topology + Communications
  - Performance of clusters: real-world specifics—not “as advertised”
- What kind of problems get enabled through the use of clusters?
  - Clusters do make practical business sense
- Summary

---

# CAE areas in automobile engg. – 1

- Structural
  - Strength and Stiffness/Deflection: FEM
  - Impact / Crash (Vehicle as a Structure)
  - Fracture Toughness (Notch Sensitivity)
  
- CFD
  - Aerodynamic Drag Calculations
  - Combustion Modeling
  - Thermal Analysis
  - Air Conditioning

---

# CAE areas in automobile engg. – 2

- More Specialized Topics
  - Chassis Vibrations
  - Noise and Technical Acoustics
  - Electromagnetic
- Multi-Disciplinary
  - Fracture Mechanics of Composites
  - Simulation of Braking

---

## ...Also, CAE in manufacturing...

- Mold and Die Design
  - Press work
  - Jigs and fixtures
  
- Casting Design
  - Gator and runner design
  - Metal flow simulation
  - Solidification simulation
  
- Process Design
  - Rolling, Wire-drawing, Tube-making, etc.

# What is common to all these problems?

- The common thing is matrices
  - The end model is almost always a matrix...
  - ... It is so, regardless of...
    - the application domain
    - the computational method used
  - Almost always! (Explain!!)
- Why?
  - Because, PDE's, when discretized, typically lead to matrix equations.
  - Whether the discretization is through FDM, FEM, FVM, BEM...
- And, for accurate representation, the matrices become very large ...

---

# Precisely how large is large?

- What will you call as a large problem?
- That depends!
- It depends on how much computing power is available to you...
- Not on the cost...
  - Electronics is cheap
  - Unlike the mechanically powered machines

# Required FLOPS for CAE – 1

## Calculations on a single node

- Assume a typical year-2007 PC desktop
  - Intel P4, 3 GHz, Dual Core, with 1 GB DRAM at 533/733 MHz
  - The test program memory requirement: RAM >> Cache
    - say, 100 MB and 1 MB
  - Double precision floating point operations
  
- What is the MFLOP/s figure for such a (machine + problem)?
  - It's just about **85 MFLOP/s** (for a matrix of size 3,500 X 3,500)
    - Caution: Reported figures can be as high as 500 MFLOP/s, or even higher, for the in-cache computations (say, matrix of size 128 X 128)
    - But, all engineering design calculations involve large matrices
      - That necessarily means → out-of-cache memory access → Slow!!
  
- So, what is the real speed in real number crunching that we get?
  - Just about **0.1 GFLOP / s / desktop**
  - That is, **300 GFLOP / desktop / hour**

# Required FLOPS for CAE – 2

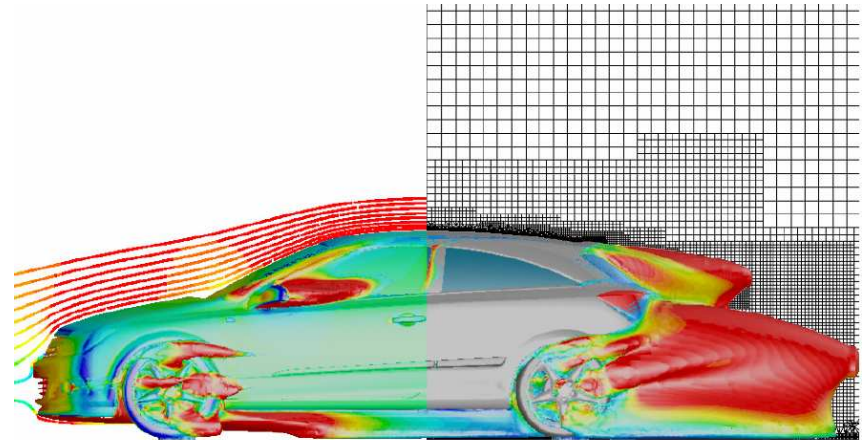
## Example: a single FEM calculation

- Today's "toy" problem. Assume a scalar field problem (e.g. s.s. heat cond.)
  - Problem size
    - No. of DOFs = Order of matrix (N) = 10,000 = 0.01 M
    - Simple-minded RAM requirement ~ 800 MB
  - Assume band Cholesky method
    - Estimated problem size:  $N^{2.33} =$  ≈ 2.15 GFLOP
    - Time on a single node: ~ 25 s
  - But what if an indirect / iterative method is not suitable?
    - Estimated problem size:  $1/3 N^{3.0} =$  ≈ 333.33 GFLOP
    - Time on a single node: ~ 3922 sec (1 hr 5 min)
  
- The new breed of problems
  - (Problem size figures taken from: ANSYS literature, March 2007)
  - Total DOF: 16 to 400 M
  - RAM: 128 GB and up
  - Problem Size: 100 G to 100 T FLOP ++
  - Time: 20 mins to 350 hrs (2 wks)
    - 1 TFLOP take 3.5 hrs on a single desktop

# Required FLOPS for CAE – 3

Example: An aerodynamics model

- 3D Mesh: **512<sup>3</sup>** (128 M cells)
  - “Low” Accuracy + Direct method
    - No special turbulence modeling
    - 6,561 FLOP/variable/matrix element (1/3 X 27<sup>3</sup>)
    - Problem Size: **818 GFLOP**
  - Similar problem size, for:
    - “High” Accuracy + Indirect methods (MG/CG)
  - Time Required
    - **3 Hours** (on a P4, 3GHz, 2007)
  
- But even just slightly more complexity...
  - **30 hours (> 1 day)**
  
- We do need FLOP/s here!!



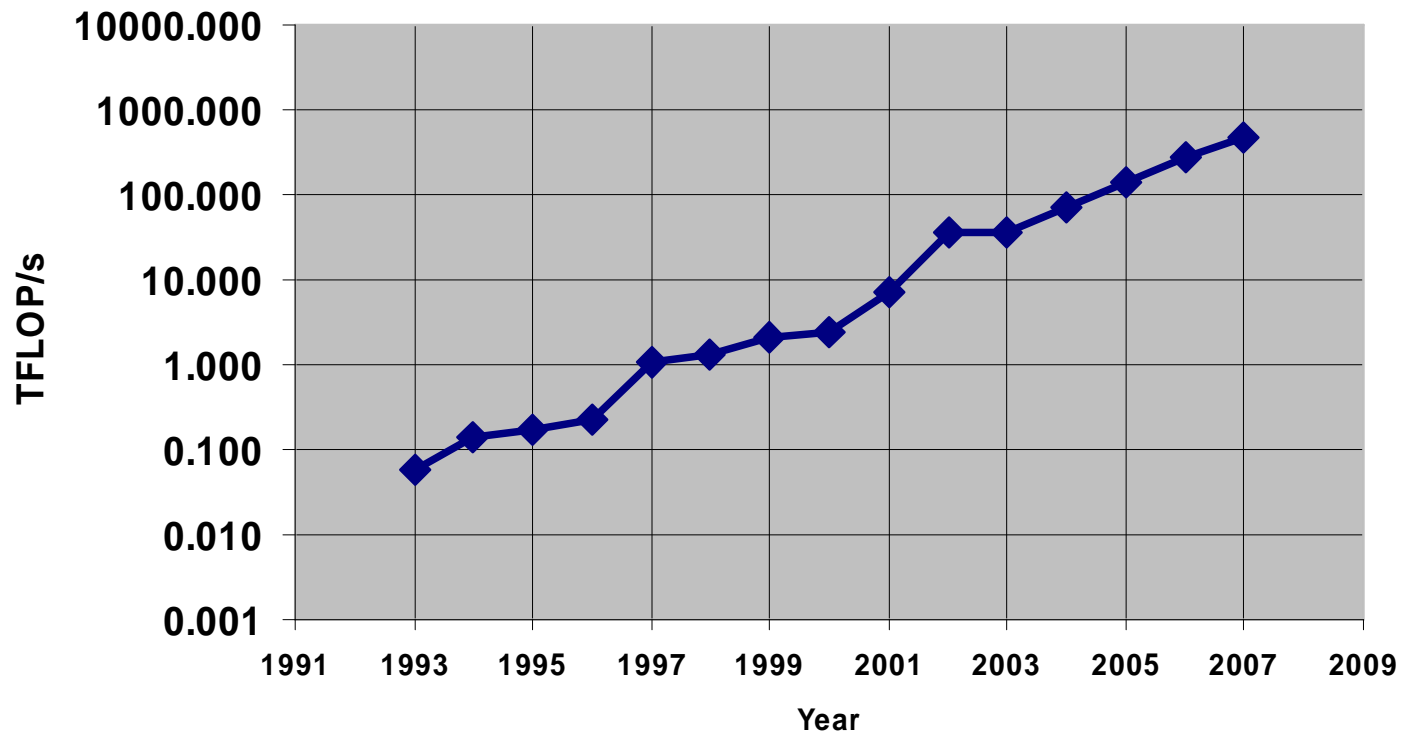
---

# Where do we get that much power from?

- Supercomputers
  - Like IBM's BlueGene or C-DAC's PARAM
  - Often, carry specialized operating systems and systems software
  - Almost always require a specialist knowledge to operate them
- Clusters of PCs
  - Use commodity hardware
  - Use the usual operating systems: Linux, and even Windows!
  - Use public domain software for interconnecting the PCs
  - You don't need a PhD to know how to operate a cluster
  - They are scalable: From 10 to 10,000 PCs

# Just how much has been available?

Fastest Supercomputer in the World



# Computational power is a practical need...

- IDC Report (June 2006) (quoted in the Microsoft whitepaper):
  - The overall high performance technical computing market
    - 4 consecutive years of 20+ % growth rate
    - Revenue in 2005: \$ 9.2 billion (~ Rs. 40,000 Crores)
    - HPC Clusters share: more than half. (~ Rs. 20,000 Crores)
  - Companies are buying HPC clusters.... Why?
    - Price, Performance, System throughput, Total cost of ownership
    - Basically because it's better than submitting the job to a mainframe
  - *“Now we are seeing smaller clusters in the design centers themselves, along with 150-to-200 node clusters for the really big jobs... The results come back on a server on even on the individual engineer's desktop computer.”*

# A photograph of a cluster of 14 PCs

- A cluster requires
  - Fast interconnecting switches
  - And, BTW, special air-conditioning too
    - One PC: 350 watt
    - This is a significant amount of heat for hundreds or thousands of PCs in a single room



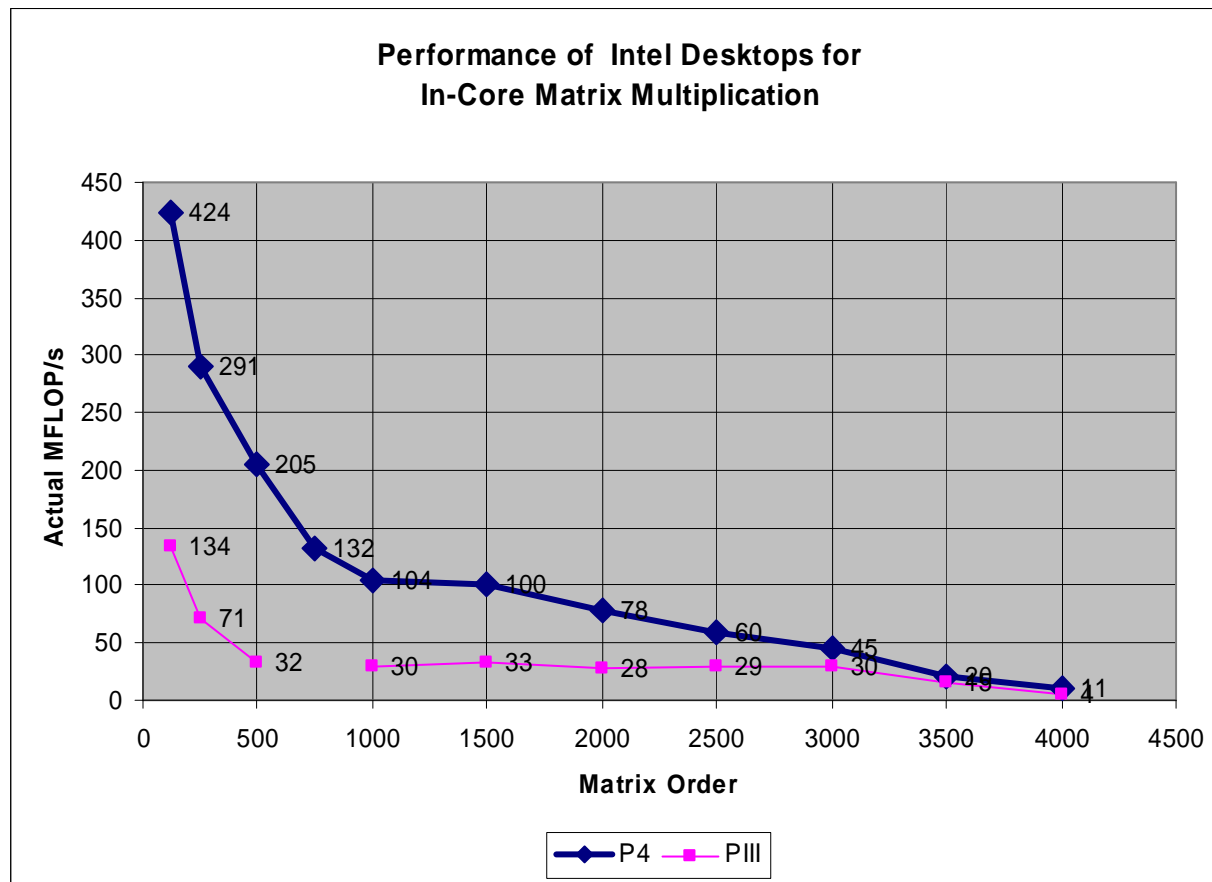
---

# Cluster computing: Some details...

- Cluster = Nodes + Connections + Topology
- Only the nodes compute
- Communications are inherently slower
- Topology can affect calculations...

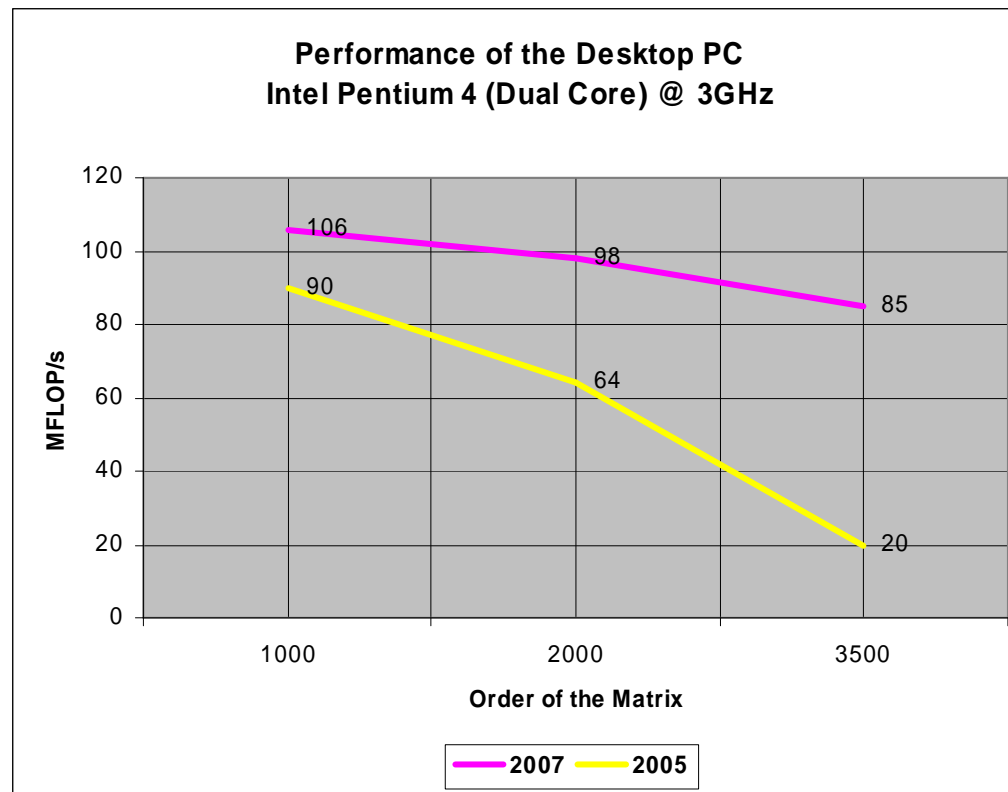
# Performance at a single node – 1

(Pentium Processor: in '03 and '05)



# Performance at a single node – 2

P4, out-of-cache, '05 vs. '07



---

# Performance at a single node – 3

- There is a lot of a degradation in out-of-cache matrix multiplication
  - From 300 MFLOP/s to just about 85 MFLOP/s
  - The slow-up is about 4 times!
- Reasons
  - Cache incoherence
  - Poor data locality
- Yet, in the above examples, matrices were at least in the core
- Else, we should expect yet another hit to performance
  - Out of core data → Virtual memory → Disk thrashing
  - → a further 10 to 100 times slower operation

# Cost of inter-node communications

- Simply adding hardware won't make it faster either!
- Today's commodity hardware for interconnection:
  - Fast Ethernet Switch: 100 Mbps (full duplex)
  - Understanding this kind of "fast" figure...
    - But that is just 12.5 MB/s (assuming full duplex)
    - Which, in turn, is just 1.56 MFLOP/s. (Theoretically.)
    - Recall: Even a P4DC had 85.00 MFLOP/s in actual computations !
    - So, best commodity communication is about 50 times slower as compared to the worst computation on a single node
- Gigabit Ethernet Switch
  - 10 times faster as compared to the Fast Ethernet...
  - But costs a lot more! And, still, 5X slower as compared to computations
- Performance deteriorates very rapidly as computers are added...
- All-to-all communication cost scales as  $N^2$

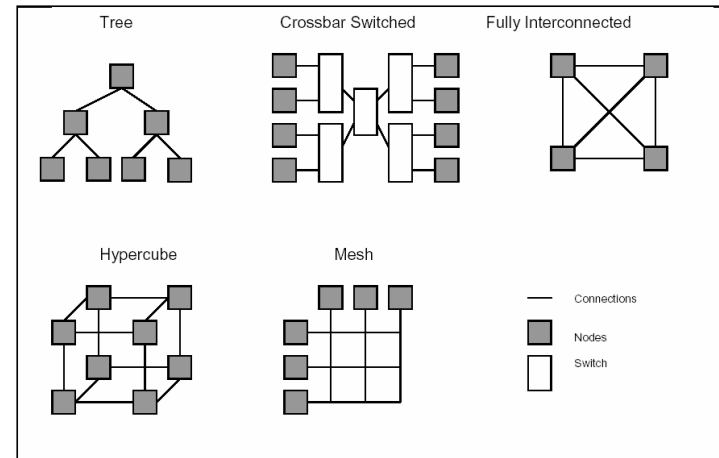
# To reduce communications...

- One Solution: Use Multi-Core SMP nodes
  - 4/8/16/... processors per node
  - Many such nodes per cluster (10's, 100's, even 1000's)
  
- Another Solution: Use Clusters of GPUs
  - What is a GPU?
    - Graphical Processing Unit
    - Employs massively multi-threaded architecture
    - Designed for the pipelined operations typically required in graphics processing
    - nVIDIA Tesla: A single board carries 128 processors!!
  - Why is it so hot?
    - At 1 to 4 GFLOP/s per core, a single board can give up to 500 GFLOP/s peak
  - But...
    - Only for single precision arithmetic!!

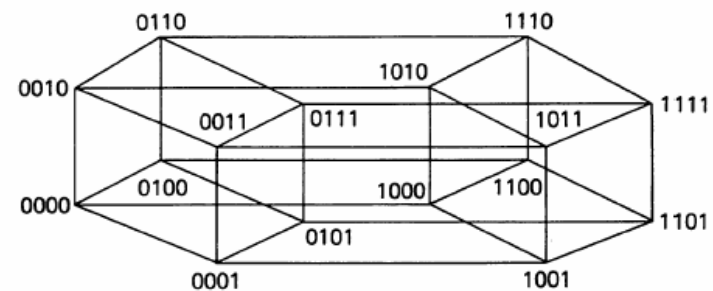
# Topology: What is it? Why does it matter?

- Topology means: How the nodes are connected to each other

- Tree
- Ring
- Lattice
- Hyper-Cube



- Topology matters, because
  - Each problem has a different kind of data dependency
  - Which, in turn determines communication patterns



# Topology: To what extent can it matter?

- Let us take an example of a particular algorithm...
- The time required for Gaussian elimination:
  - $O(N^3)$  on a single machine
  - $O(N^2)$  on linear array of  $N$  nodes
  - $O(N)$  on a mesh of  $N \times N$  size, or a hyper-cube with  $N^2$  nodes
- What does that mean? ... Something like total times taken by...
  - A complete ODI match, both innings put together (8 hrs)
  - The last 3 overs of the second innings (15 min)
  - One delivery (30 s)
- Obviously, topology can make a lot of difference to the program execution time.
- But requires special things (algorithms, software, initial setup time...)

---

# Overall, what to expect with clusters?

- Improvement in Computational Performance
- How much?
- Enough to bring about a qualitative change...

---

# Qualitative change? In what way?

What kind of problems get enabled?

- Modeling for turbulence in CFD
  - 3D turbulence modeling can come within reach
  - Small changes in design can take you to the sweet zone of
    - fuel efficiency and power
  
- Combustion modeling
  - What can be the worth of just a single idea?
    - It can be at least Rs. 200 crore (\$ 50 Million) worth!!
      - That is, in Indian scenario alone—not international
      - (Remember the recent patents related news in the media!)
  - Question: How much does a 128 node cluster cost?
    - Answer: Less than Rs. 1 crore

---

# Some more problems that get enabled...

- Structural
  - It becomes possible to run FEM...
    - For each sub-cycle of the design iteration
    - For each component
    - “Analysis at Fingertips”
      - It is easy to connect the cluster to your own desktop PC
  - Optimization
    - For instance, shape optimization
    - Automation still not possible in the immediate future
    - But can lead to reduction in material costs anyways!

# Some specific or practical examples...

- For example, consider the following practical questions:
  - What is the effect of the rebound of a nonlinear spring?
  - What happens to a new type of foam in the driver's seat after 180,000 miles of running? (Are you going to wait for all those miles and years to end?)
  - What will happen to loose objects inside the car—say a lunchbox—in a 25 mph collision?
  
- Dr. Steve Rohde identifies six areas where clusters are beneficial
  - Formerly, of General Motors (Ref: The Microsoft Whitepaper)
  - Robust designs
  - Integration of domains and optimization
  - Clever approaches
  - Collaboration
  - Integration with business models
  - Process automation

---

# But, as always, “look before you leap”...

- Software Issues
  - Software rewrites are inevitable
    - Debugging, reliability...
  - Many serial algorithms are not easy to parallelize
    - Just try Gaussian Elimination (!!)
    - An exception is: MC and similar methods (e.g. FAQ—our new approach)
- Commercial availability of the software
  - Most companies are already coming out with cluster-enabled versions
  - Some software that worked very fast on single workstations may not necessarily work equally well on clusters too...
    - Re-evaluate!

---

# Summary and Conclusions – 1

- “Real” CAE applications require a lot of computing power
- Clusters provide scalable computing power
  - Small Clusters (10 nodes)
  - Supercomputer (10,000 nodes)
- However, clusters entail software rewrites
  - You have to get a special version of your FEA software
    - Whether commercial or “open source” !!
    - Most “open source” software is not yet parallelized
- Yet, the hardware is pretty easy to build...
  - A cluster can be built by your own in-house IT team
  - No need to buy a separate supercomputer

---

# Summary and Conclusions – 2

- Clusters is a relatively inexpensive technology
  - Rs. 1 Crore → 128 Nodes (+ Switches + A/C)
  - That will give you, a speedup of, say 100 times
  - Total computational power of, say 10 to 50 GFLOP/s
    - Realistic estimate. No hidden costs. (Figures for large matrices.)
  - That is like (lower-end) supercomputers from turn of the last decade!!
  
- What does a speedup of 100 mean?
  - From 30 days to 7.2 hours
  - From an overnight run (12 hours) to a cup of coffee (7 minutes)
    - You can have a major analysis done by your engineer right while your conference call was still in progress...
  
- Clearly, clustering is an enabling technology
  
- Use it to your business advantage

---

# References and further information

- Sources (Internet searches...)
  - Hrvoje Jasak
    - The car CFD figure
  - Henry Neeman (University of Oklahoma)
    - Chart: Supercomputer performance over the years
  - S. Turek (University of Heidelberg)
    - Acute observations on real world problems (sparse matrices)
  - M. T. Heath (University of Illinois, Urbana-Champaign)
    - Class notes giving comparisons of algorithmic complexities
  - <http://www.microsoft.com/hpc>
    - White paper: “Desk-side supercomputing in automotive industry”
  - <http://www.hipc.org>
    - General information on high performance computing
  - <http://www.nvidia.com>
    - GPUs
  - <http://www.random.org>
    - Using out-of-tune radio receiver for random number generation
  - Others...

---

# Thank you!

- Author's Web site
  - <http://www.JadhavResearch.info>