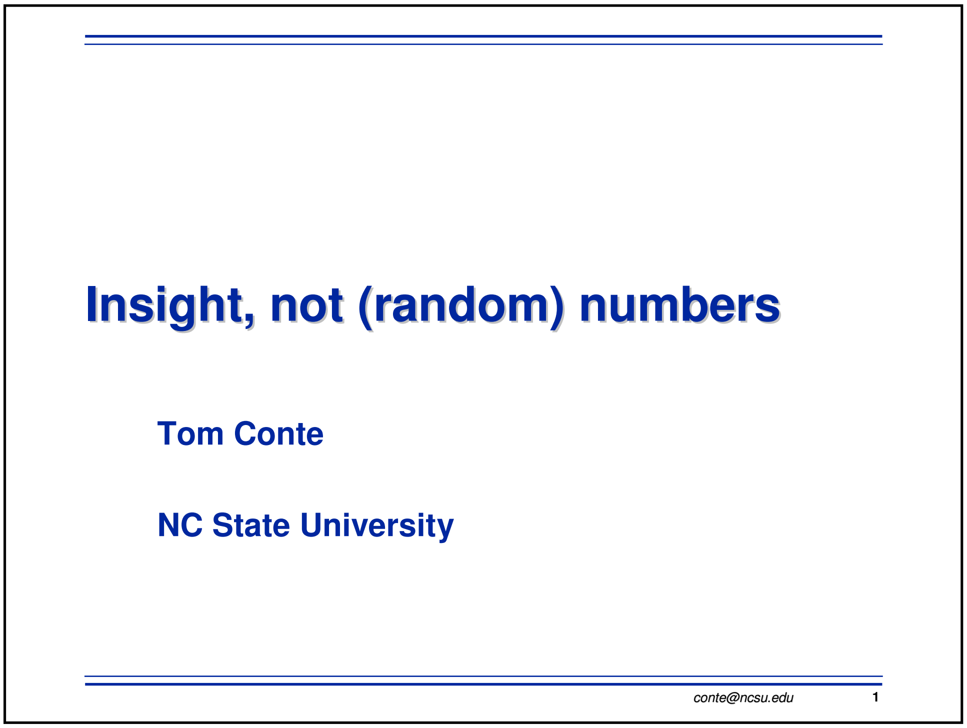
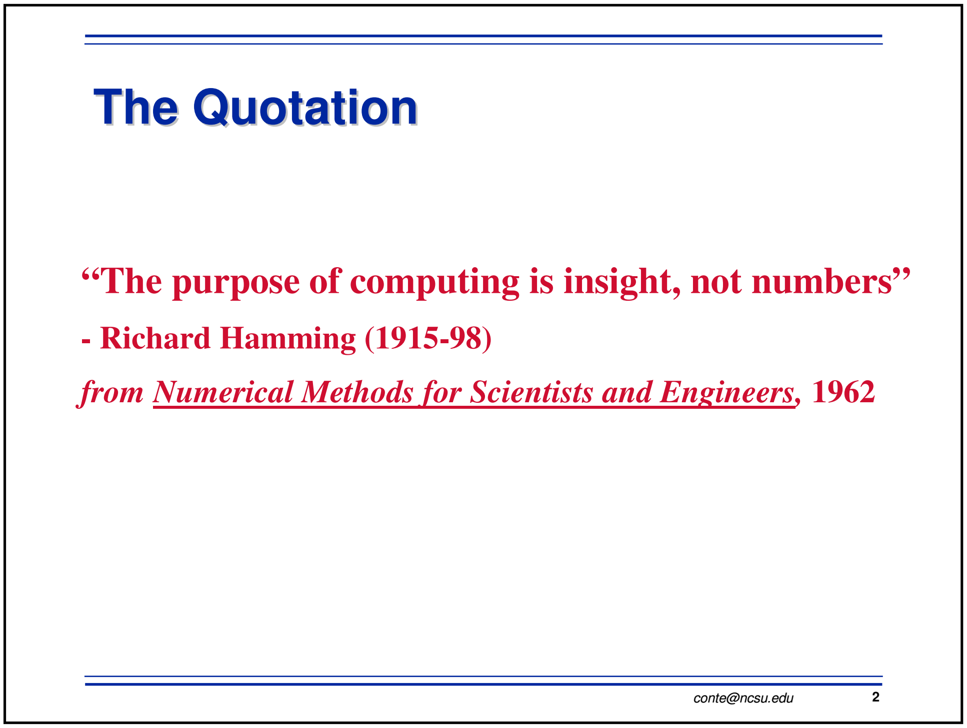


Insight, not (random) numbers

Tom Conte

NC State University

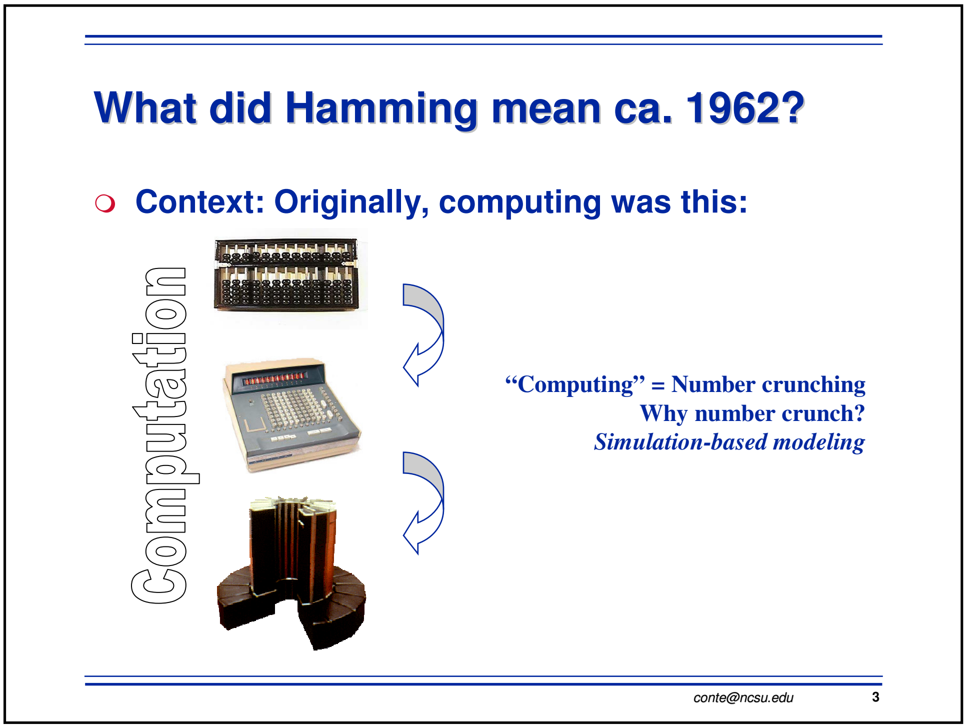


The Quotation

“The purpose of computing is insight, not numbers”

- Richard Hamming (1915-98)

from *Numerical Methods for Scientists and Engineers*, 1962

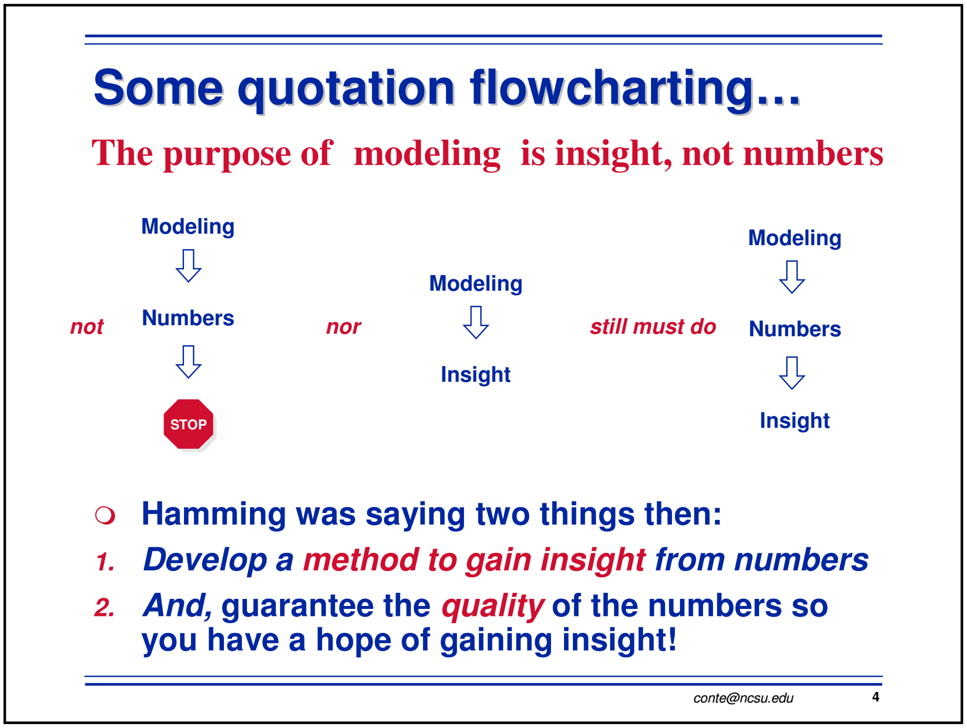


What did Hamming mean ca. 1962?

- Context: Originally, computing was this:

Computation

"Computing" = Number crunching
Why number crunch?
Simulation-based modeling



Some quotation flowcharting…

The purpose of modeling is insight, not numbers

not Modeling ⇩ Numbers ⇩ STOP

nor Modeling ⇩ Insight

still must do Modeling ⇩ Numbers ⇩ Insight

- Hamming was saying two things then:

1. *Develop a method to gain insight from numbers*
2. *And,* guarantee the *quality* of the numbers so you have a hope of gaining insight!

The current state of affairs: On insight from numbers

- We have “modeled” this car and determined:
  - 0-60 time in 4.9 sec, 500 horsepower and 383 lb.-ft. of torque. Engineered to rev with a redline of 8,250, Top speed of 205 mph

2005 BMW M6

(It’s $160,000)

(plus tax)

So??? Does it fit my needs?

What *are* my needs? (and who’s ‘me’, for that matter– driver, designer or salesman?)

On the road to insight… who consumes the numbers?

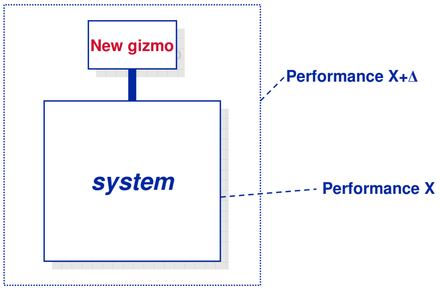
- In the case of a Bimmer, “number consumers” are:

1. The buyer (the obvious)
2. The marketer/salesman (the parasitic)
3. The car designer (the noble?)

- And in the case of computers? The same three suspects: buyer, marketer, architect
- Let’s take these on one at a time (…in reverse order)

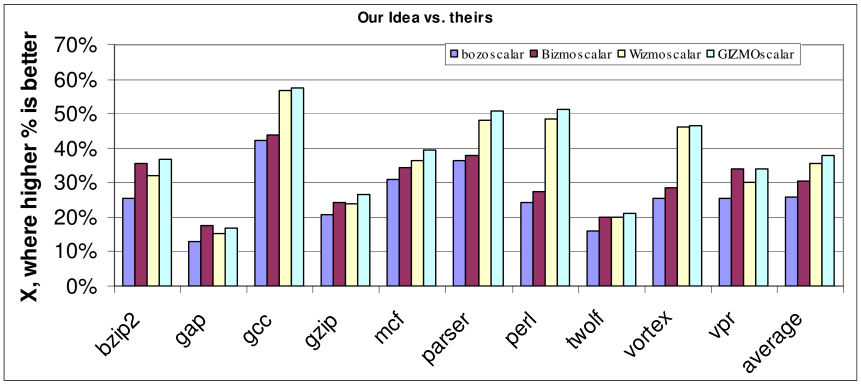
What are we trying to do with modeling?

- View #1: *The architect*



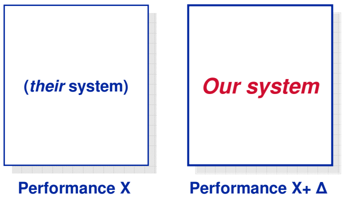
If Δ > 0, then my idea is a good idea…

Gizmoscalar *The future of Computer Architecture*



What are we trying to do with modeling?

- View #2: *Marketing*


(*their* system)

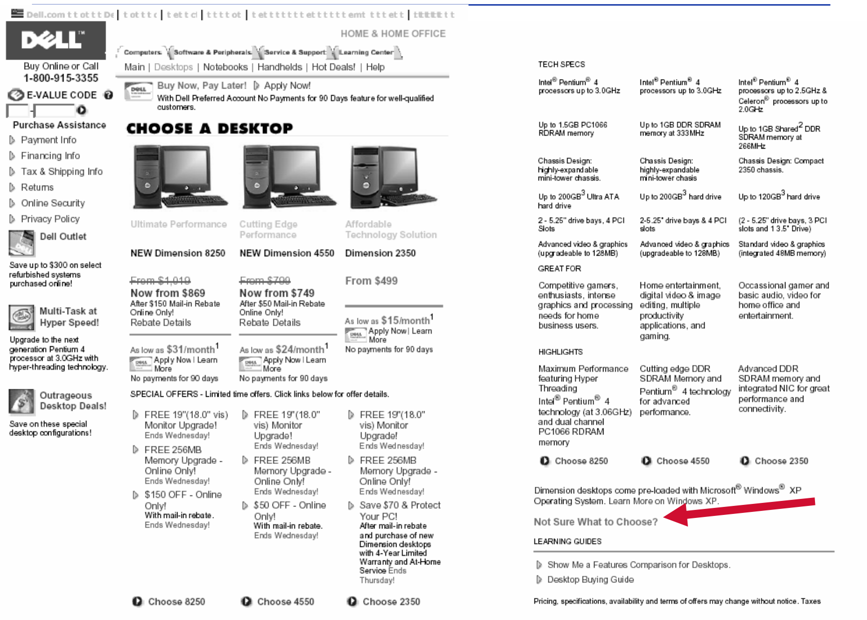
Performance X

Our system

Performance X+ Δ

If Δ > 0 Buy our system

Well, not really...


Dell.com t t ot t t De | t ot t t c | t et t c | t t t t ot | t et t t t t t t et t t t t t emt t t t et t | tttttt t t
DELL™
HOME & HOME OFFICE
Computers
Software & Peripherals
Service & Support
Learning Center
Main | Desktops | Notebooks | Handhelds | Hot Deals! | Help

Buy Online or Call 1-800-915-3355

E-VALUE CODE

Purchase Assistance

- Payment Info
- Financing Info
- Tax & Shipping Info
- Returns
- Online Security
- Privacy Policy

Dell Outlet

Save up to $300 on select refurbished systems purchased online!

Multi-Task at Hyper Speed!

Upgrade to the next generation Pentium 4 processor at 3.0GHz with hyper-threading technology.

Outrageous Desktop Deals!

Save on these special desktop configurations!

Buy Now, Pay Later! Apply Now!

With Dell Preferred Account No Payments for 90 Days feature for well-qualified customers.

CHOOSE A DESKTOP

Ultimate Performance	Cutting Edge Performance	Affordable Technology Solution
NEW Dimension 8250	NEW Dimension 4550	Dimension 2350
~~From $1,019~~ Now from $869 After $150 Mail-in Rebate Online Only! Rebate Details	~~From $799~~ Now from $749 After $50 Mail-in Rebate Online Only! Rebate Details	From $499
As low as $31/month[1] Apply Now | Learn More No payments for 90 days	As low as $24/month[1] Apply Now | Learn More No payments for 90 days	As low as $15/month[1] Apply Now| Learn More No payments for 90 days

SPECIAL OFFERS - Limited time offers. Click links below for offer details.

FREE 19"(18.0" vis) Monitor Upgrade! Ends Wednesday!	FREE 19"(18.0" vis) Monitor Upgrade! Ends Wednesday!	FREE 19"(18.0" vis) Monitor Upgrade! Ends Wednesday!
FREE 256MB Memory Upgrade - Online Only! Ends Wednesday!	FREE 256MB Memory Upgrade - Online Only! Ends Wednesday!	FREE 256MB Memory Upgrade - Online Only! Ends Wednesday!
$150 OFF - Online Only! With mail-in rebate. Ends Wednesday!	$50 OFF - Online Only! With mail-in rebate. Ends Wednesday!	Save $70 & Protect Your PC! After mail-in rebate and purchase of new Dimension desktops with 4-Year Limited Warranty and At-Home Service Ends Thursday!
Choose 8250	Choose 4550	Choose 2350

TECH SPECS

Intel® Pentium® 4 processors up to 3.0GHz	Intel® Pentium® 4 processors up to 3.0GHz	Intel® Pentium® 4 processors up to 2.5GHz & Celeron® processors up to 2.0GHz
Up to 1.5GB PC1066 RDRAM memory	Up to 1GB DDR SDRAM memory at 333MHz	Up to 1GB Shared[2] DDR SDRAM memory at 266MHz
Chassis Design: highly-expandable mini-tower chassis.	Chassis Design: highly-expandable mini-tower chasis	Chassis Design: Compact 2350 chassis.
Up to 200GB[3] Ultra ATA hard drive	Up to 200GB[3] hard drive	Up to 120GB[3] hard drive
2 - 5.25" drive bays, 4 PCI Slots	2-5.25" drive bays & 4 PCI slots	(2 - 5.25" drive bays, 3 PCI slots and 1 3.5" Drive)
Advanced video & graphics (upgradeable to 128MB)	Advanced video & graphics (upgradeable to 128MB)	Standard video & graphics (integrated 48MB memory)

GREAT FOR

Competitive gamers, enthusiasts, intense graphics and processing needs for home business users.	Home entertainment, digital video & image editing, multiple productivity applications, and gaming.	Occassional gamer and basic audio, video for home office and entertainment.

HIGHLIGHTS

Maximum Performance featuring Hyper Threading Intel® Pentium® 4 technology (at 3.06GHz) and dual channel PC1066 RDRAM memory	Cutting edge DDR SDRAM Memory and Pentium® 4 technology for advanced performance.	Advanced DDR SDRAM memory and integrated NIC for great performance and connectivity.
Choose 8250	Choose 4550	Choose 2350

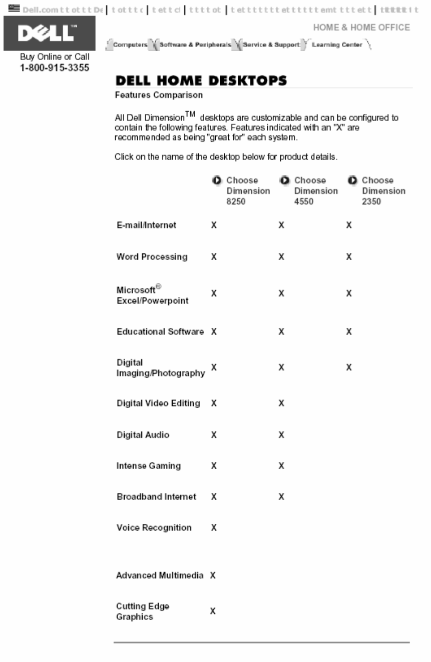
Dimension desktops come pre-loaded with Microsoft® Windows® XP Operating System. Learn More on Windows XP.

Not Sure What to Choose?

LEARNING GUIDES

- Show Me a Features Comparison for Desktops.
- Desktop Buying Guide

Pricing, specifications, availability and terms of offers may change without notice. Taxes



HOME & HOME OFFICE

Buy Online or Call
1-800-915-3355

DELL HOME DESKTOPS

Features Comparison

All Dell Dimension^{TM} desktops are customizable and can be configured to contain the following features. Features indicated with an "X" are recommended as being "great for" each system.

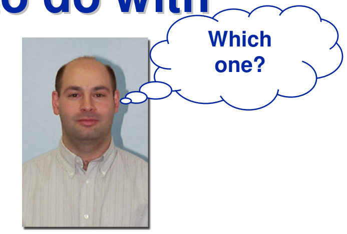
Click on the name of the desktop below for product details.

	Choose Dimension 8250	Choose Dimension 4550	Choose Dimension 2350
E-mail/Internet	X	X	X
Word Processing	X	X	X
Microsoft® Excel/Powerpoint	X	X	X
Educational Software	X	X	X
Digital Imaging/Photography	X	X	X
Digital Video Editing	X	X	
Digital Audio	X	X	
Intense Gaming	X	X	
Broadband Internet	X	X	
Voice Recognition	X		
Advanced Multimedia	X		
Cutting Edge Graphics	X		

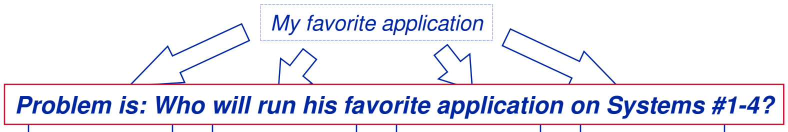
Insight for free
No numbers required!

What are we trying to do with modeling?

- View #3: *The (smart) users*



Which one?

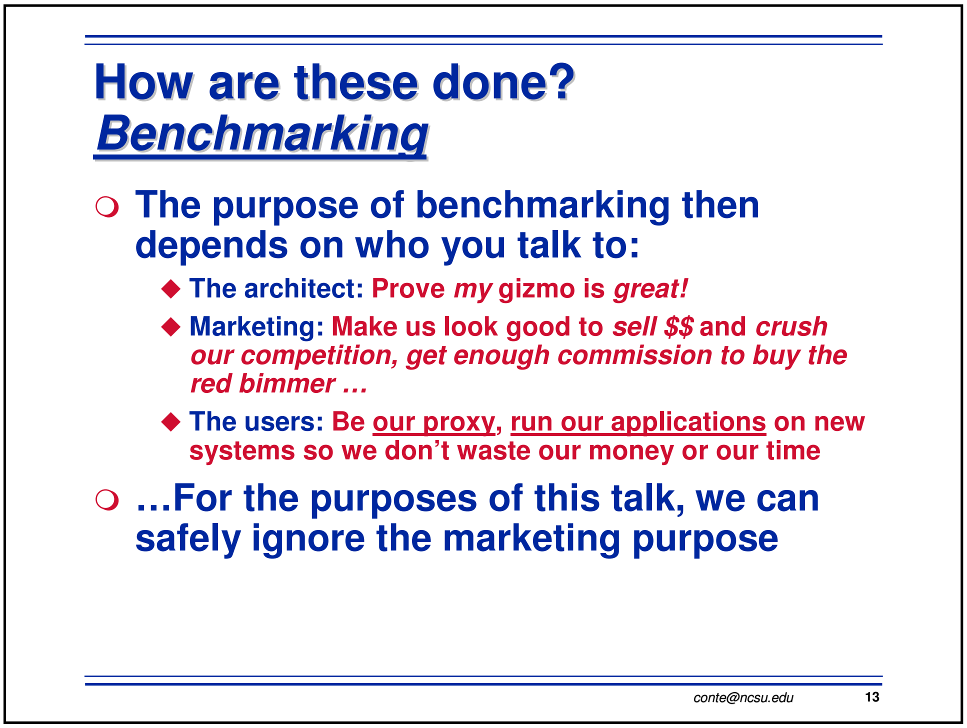


My favorite application

Problem is: Who will run his favorite application on Systems #1-4?

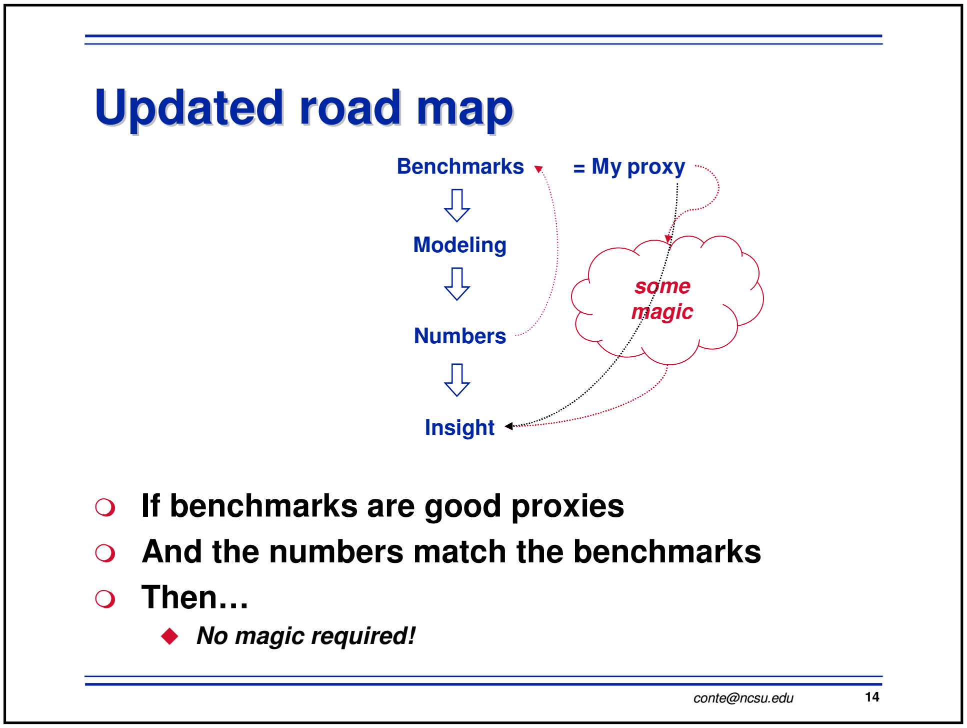
System #1	System #2	System #3	System #4
Performance X_1	Performance X_2	Performance X_3	Performance X_4

Pick max of X_i …



How are these done? *Benchmarking*

- The purpose of benchmarking then depends on who you talk to:
  - The architect: Prove *my* gizmo is *great!*
  - Marketing: Make us look good to *sell $$* and *crush our competition, get enough commission to buy the red bimmer …*
  - The users: Be our proxy, run our applications on new systems so we don't waste our money or our time
- …For the purposes of this talk, we can safely ignore the marketing purpose



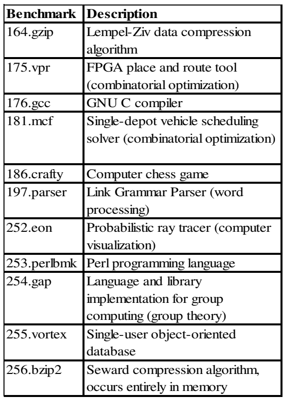
Updated road map

- If benchmarks are good proxies
- And the numbers match the benchmarks
- Then…
  - *No magic required!*

Part I: “If benchmarks are good proxies” When are they good proxies?

- Which benchmark do I believe?
- Answer: the one that is closest to what I do
- Question: Which one is that?
- Answer: Read the descriptions

What’s in SPEC…



Benchmark	Description
164.gzip	Lempel-Ziv data compression algorithm
175.vpr	FPGA place and route tool (combinatorial optimization)
176.gcc	GNU C compiler
181.mcf	Single-depot vehicle scheduling solver (combinatorial optimization)
186.crafty	Computer chess game
197.parser	Link Grammar Parser (word processing)
252.eon	Probabilistic ray tracer (computer visualization)
253.perlbmk	Perl programming language
254.gap	Language and library implementation for group computing (group theory)
255.vortex	Single-user object-oriented database
256.bzip2	Seward compression algorithm, occurs entirely in memory

- Say what you do is:

1. Surf the web
2. Database accesses
3. Logic simulation
4. CAD synthesis

- Which benchmark is the right one to listen to?

What’s in Mediabench…

Benchmark	Description
JPEG	Jpeg compression/decompression
MPEG	Decoding mpeg-1 and mpeg-2 video streams
GSM	Speech transcoding using RPE/LTP coding at 13kbits/s
ADPCM	Adaptive Differential Pulse Code Modulation algorithm for speech compression/decompression
G.721	CCITT G.711, G.721, and G.723 voice compressions
PGP	Public key encryption and authentication
PEGWIT	Public key encryption and authentication
SPHERE	Read and format NIST-formatted speech waveforms
RASTA	Filtering for speech recognition
Ghostscript	Postscript language interpreter, postscript graphics generation, PDF
Mesa	3D graphics library
EPIC	Image compression

- And which one here matches what you do with your cellphone?

A better way: Quantitative Benchmark Characteristics

Some examples:

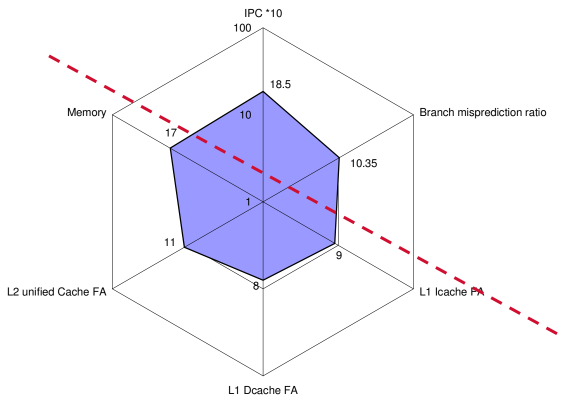
- IPC (with large memory system)
- Branch predictability (for gshare)
- Preferred L1 instruction cache size
- Preferred L1 data cache size
- Preferred L2 unified cache size
- Total virtual memory requirements (4KB page size)
- Others:
  - TLB requirements
  - Instruction frequency by type
  - System Call usage
  - …

Let’s try it out…

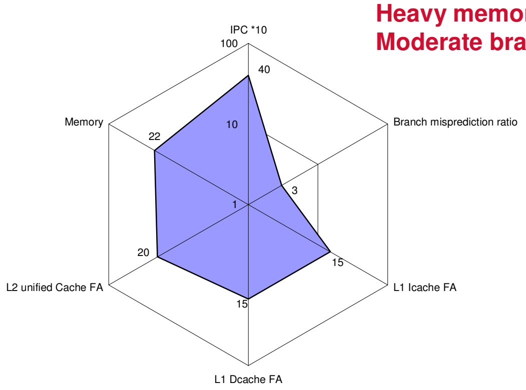
Consider these benchmark sets:

- MediaBench (UCLA)
- NetBench (UCLA and NWU)
- SPEC CPU CINT2000

Kiviat view

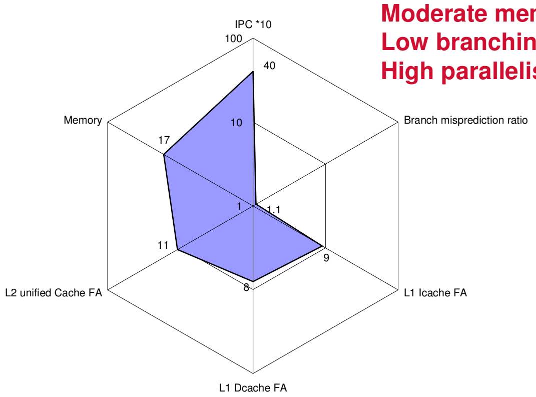


Kiviat view

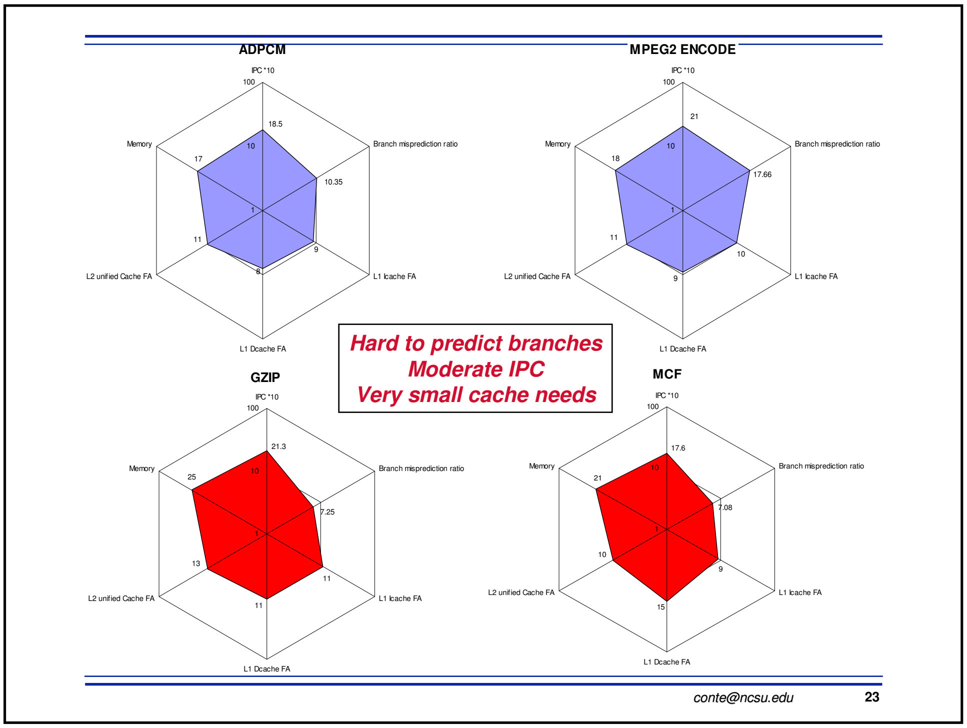


Heavy memory usage
Moderate branchiness

Kiviat view

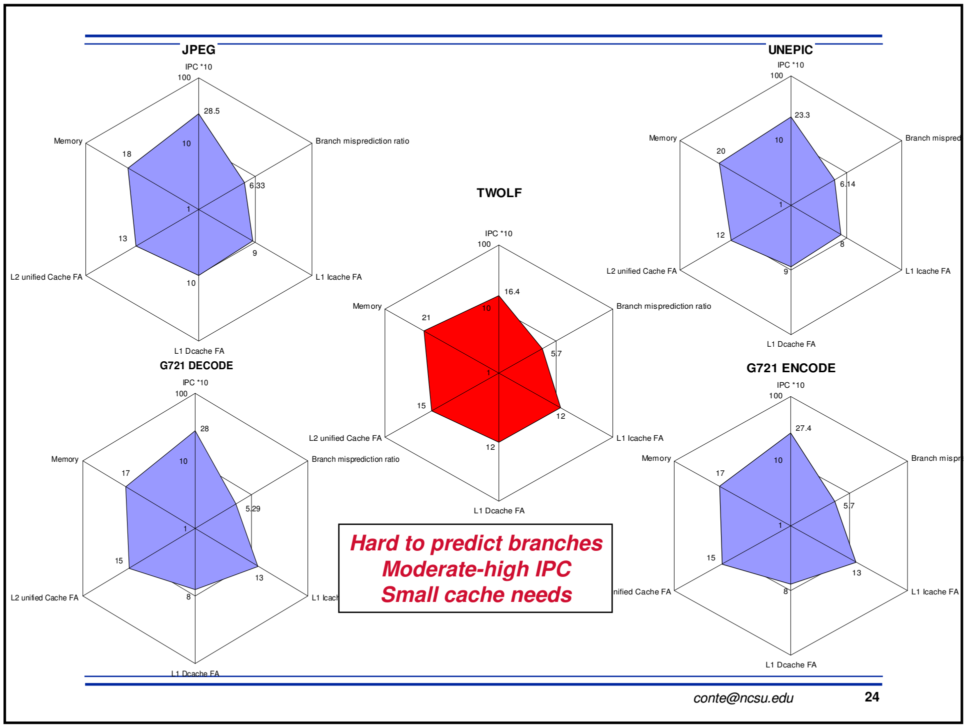


Moderate memory usage
Low branchiness
High parallelism



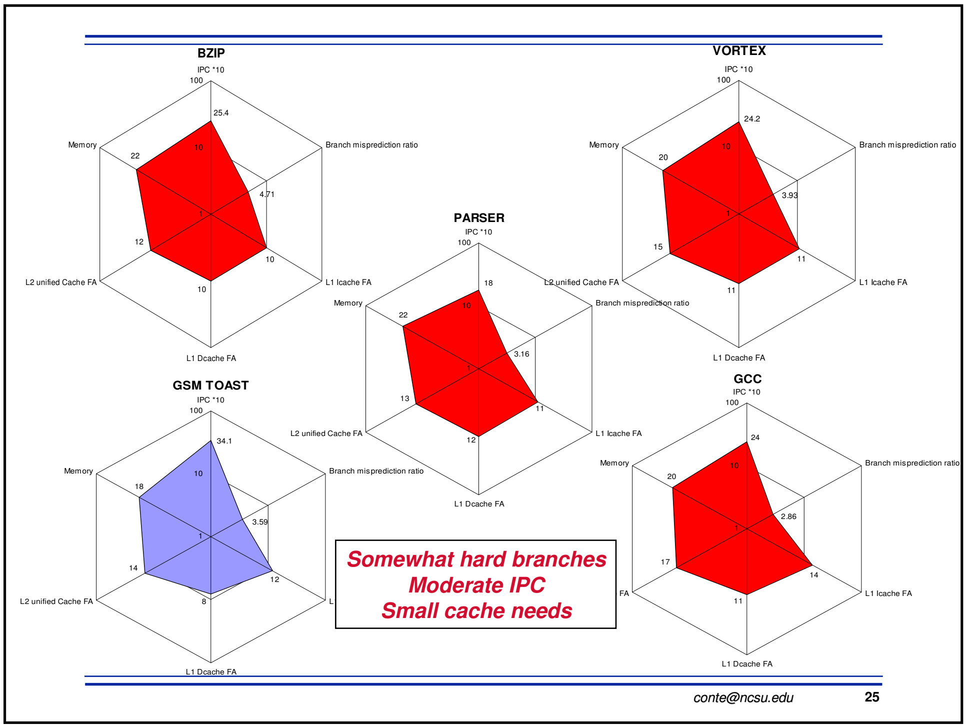
Hard to predict branches
Moderate IPC
Very small cache needs

conte@ncsu.edu 23



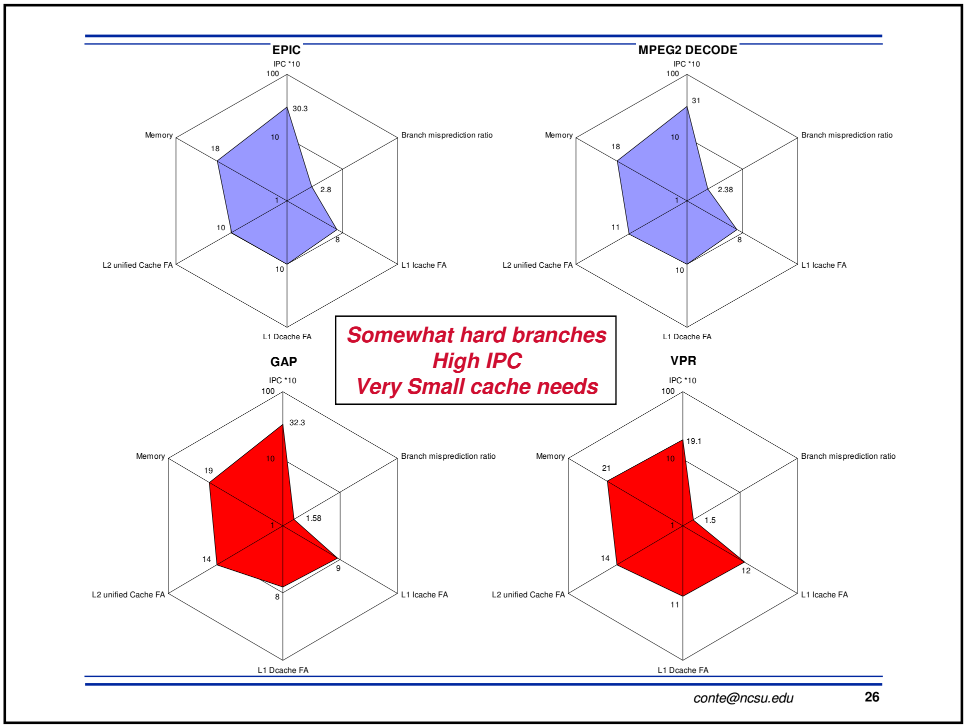
Hard to predict branches
Moderate-high IPC
Small cache needs

conte@ncsu.edu 24



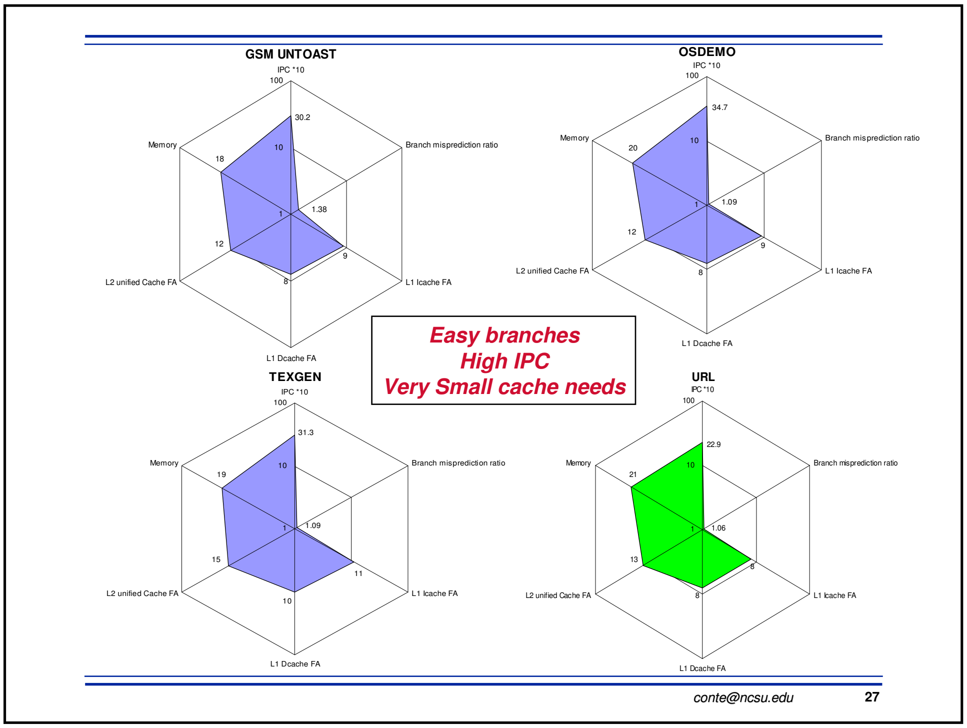
Somewhat hard branches
Moderate IPC
Small cache needs

conte@ncsu.edu 25



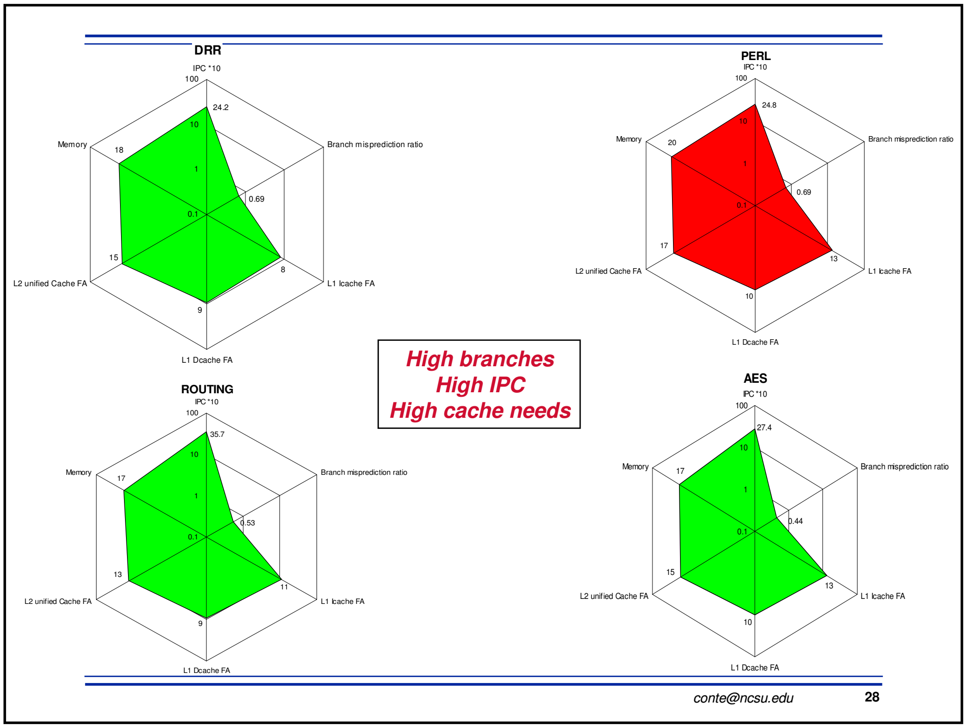
Somewhat hard branches
High IPC
Very Small cache needs

conte@ncsu.edu 26



Easy branches
High IPC
Very Small cache needs

conte@ncsu.edu **27**



High branches
High IPC
High cache needs

conte@ncsu.edu **28**

What’s it all mean?

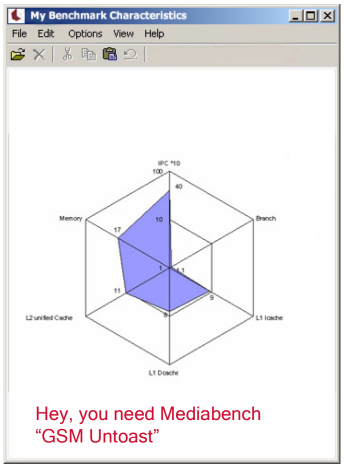
1. Benchmarks have distinct characteristics
2. Some benchmarks are similar…
   - Across different benchmark suites
   - Across different application domains

- Bottom line: *There’s hope* (!) that characteristics can be used to guarantee a benchmark is a good proxy

The missing piece:
How to make a million dollars

- Create a tool that runs unobtrusively
- The tool collects statistics about *usage* characteristics
- So you know then which benchmark to choose as your proxy …

- Taking it a step further, the tool finds the benchmarks that have those characteristics



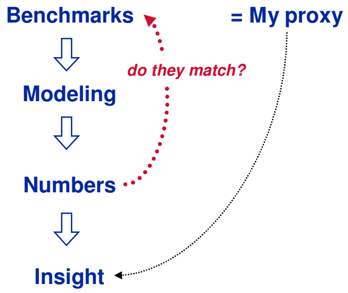
Hey, you need Mediabench “GSM Untoast”

But what about benchmark *suites?*

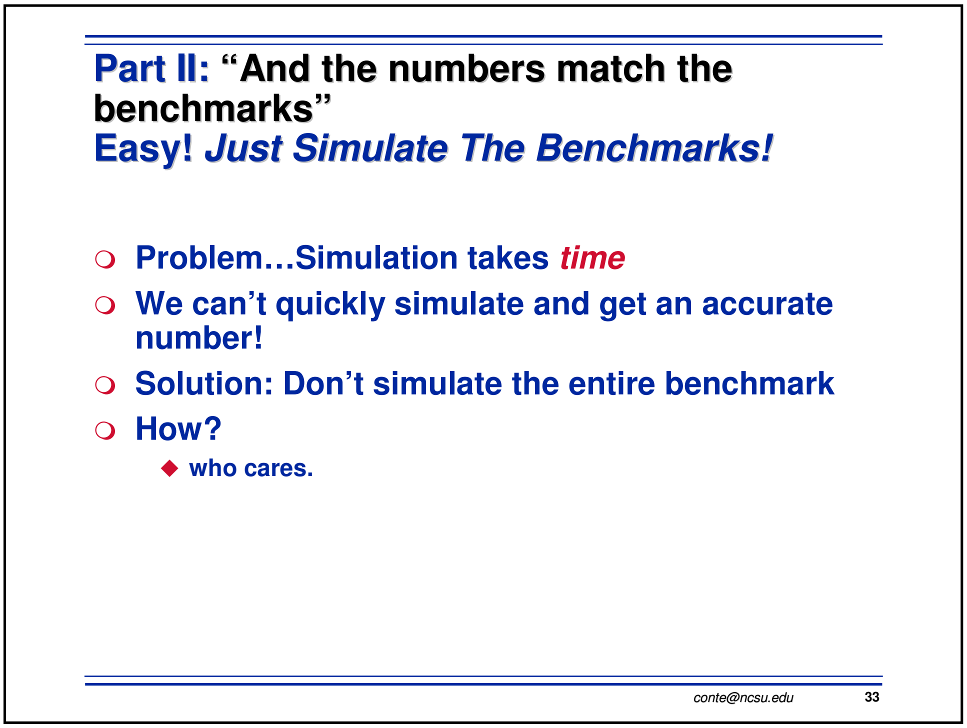
- Who creates benchmark suites today?
  - Mostly industry
  - Why? *Marketing!*

- Who speaks for the users?
  - They do. *Trust them.*
- A modest proposal:
  - Poll what users care about
  - Create benchmarks for them
  - Have an impartial panel select among these based on quantitative characteristics
  - Use this to create a benchmark suite
  - Rigorously review the suite every year
  - IMHO, better suited to academia than industry

Back to the road map

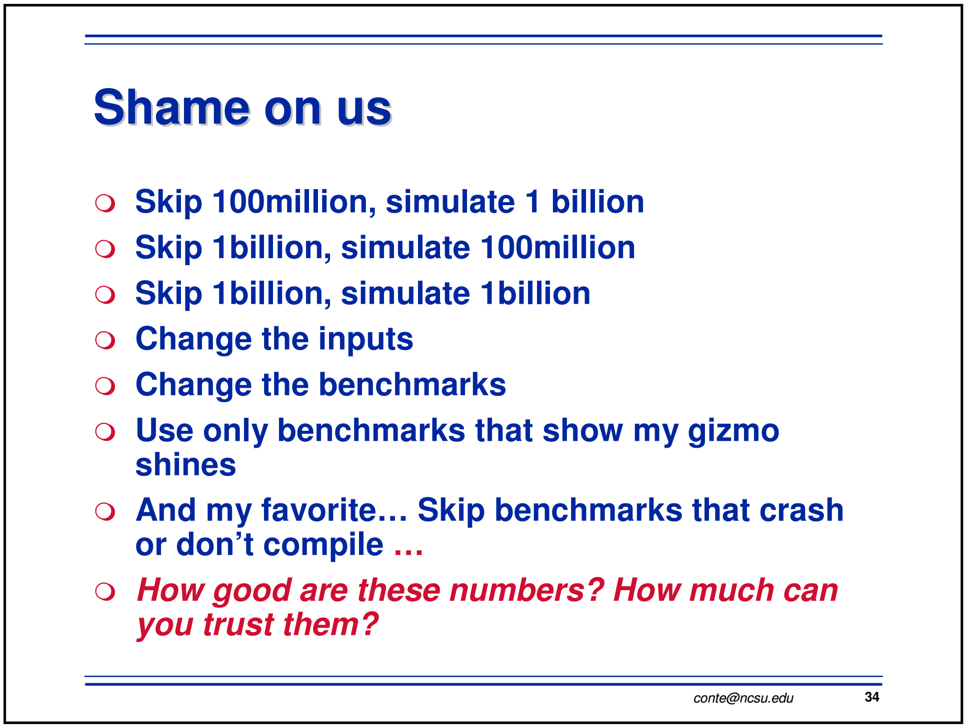


- "If benchmarks are good proxies"
  - How? Use benchmark characterization
- "And the numbers match the benchmarks"
- Then… *insight*



Part II: “And the numbers match the benchmarks”
Easy! *Just Simulate The Benchmarks!*

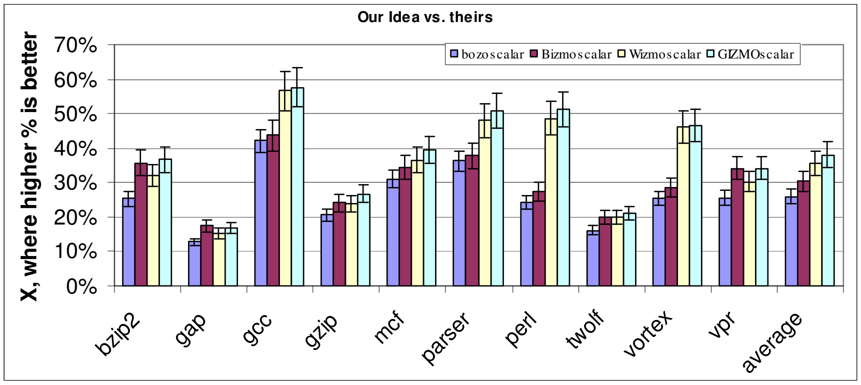
- Problem…Simulation takes *time*
- We can’t quickly simulate and get an accurate number!
- Solution: Don’t simulate the entire benchmark
- How?
  - who cares.



Shame on us

- Skip 100million, simulate 1 billion
- Skip 1billion, simulate 100million
- Skip 1billion, simulate 1billion
- Change the inputs
- Change the benchmarks
- Use only benchmarks that show my gizmo shines
- And my favorite… Skip benchmarks that crash or don’t compile …
- *How good are these numbers? How much can you trust them?*

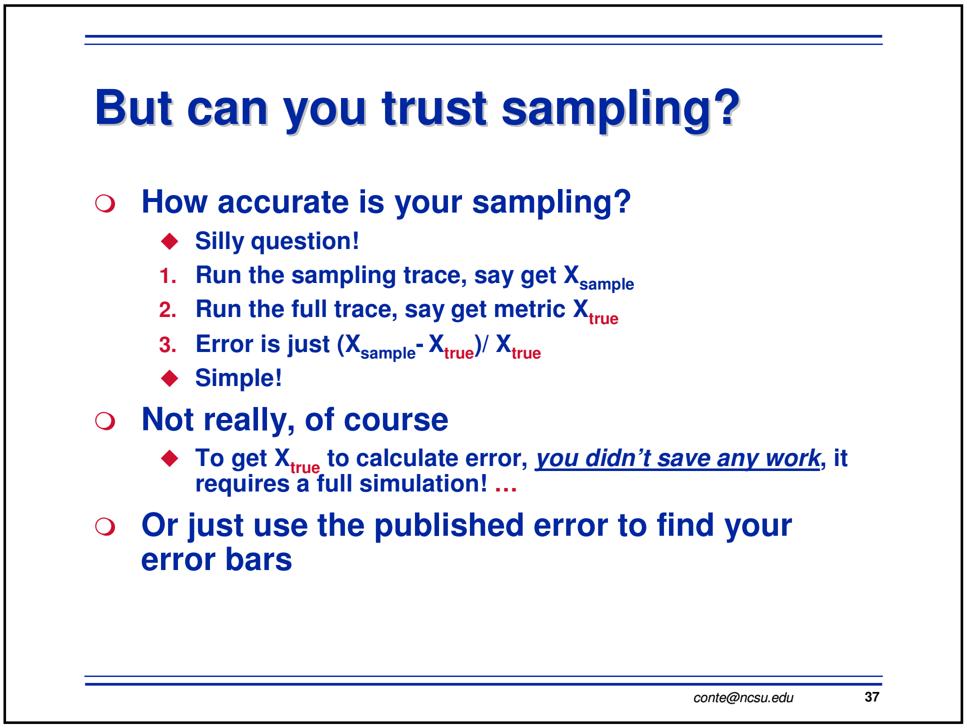
Why don’t we include error bars? I think I know why: Gizmoscalar revisited



(by the way, this is from one of my/my students’ papers)

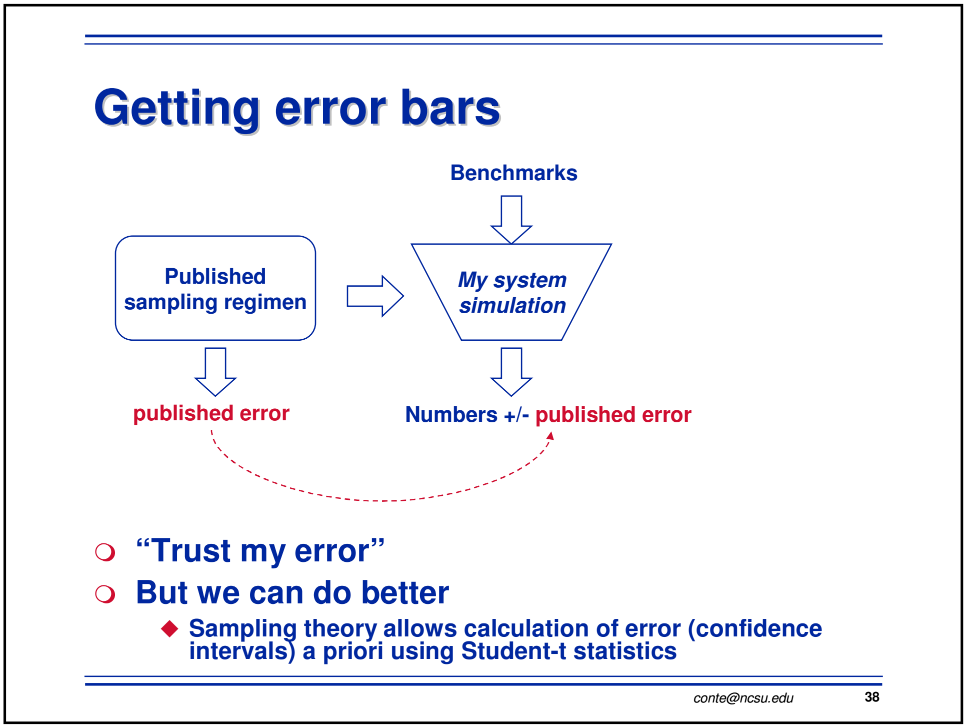
Better way: Sampling

- How to predict who will be the next president of the US:
  - Solution #1: Ask all Americans
    - Takes too long
  - Solution #2: Ask random Americans
    - Which ones? Be careful! (e.g., not just TX… not just Austin, TX)
- Saves a lot of work …
  - Pick random pieces of a benchmark ***trace*** and simulate only those
- Great idea!
  - History of sampling for fast architecture simulation:
  1. Original credit due to Laha, Patel, Iyre in 1988
  2. Early work for cache sampling only – Kessler, Fu, …
  3. Processor sampling work ca. 1992 and onwards – Menezes, Poursepanj, …
  4. Latest work on whole system sampling – cast of thousands



But can you trust sampling?

- How accurate is your sampling?
  - Silly question!
  1. Run the sampling trace, say get X_{sample}
  2. Run the full trace, say get metric X_{true}
  3. Error is just $(X_{sample}- X_{true})/ X_{true}$
  - Simple!
- Not really, of course
  - To get X_{true} to calculate error, ***you didn’t save any work***, it requires a full simulation! …
- Or just use the published error to find your error bars



Getting error bars

Benchmarks

Published sampling regimen

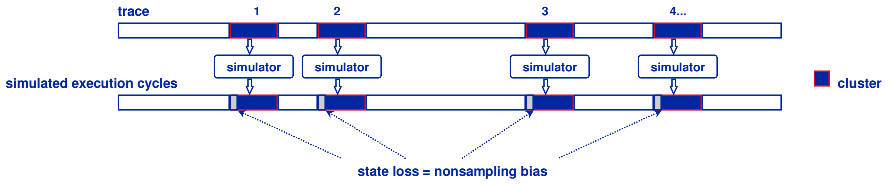
My system simulation

published error

Numbers +/- published error

- “Trust my error”
- But we can do better
  - Sampling theory allows calculation of error (confidence intervals) a priori using Student-t statistics

Trace sampling according to sampling theory



- “Trace sampling” = cluster sampling:
  - *n* clusters of *m* execution cycles each
  - Actually *indirectly sampled*: *n* clusters of *m’* instructions each
- Error due to three effects:
  - sampling bias (e.g., ask 3 people)
  - sampling variability (e.g., ask only people in TX)
  - nonsampling bias (e.g., ask people in Canada who their friend will vote for)
    - If reduced, sampling theory applies, error bars can be calculated!

Nonsampling bias

- Nonsampling bias due to *indirect sampling*
  - The measured population is different from the actual
  - For us: *System state is unknown at start of each cluster simulation …*
- If eliminated, then

$$\text{metric}_{\text{true}} = \text{metric}_{\text{sample}} \pm 1.96 \cdot S, \quad \text{for 95\% confidence interval}$$

$$\text{standard error,} \quad S = \frac{s_{\text{metric}}}{\sqrt{N_{\text{cluster}}}}$$

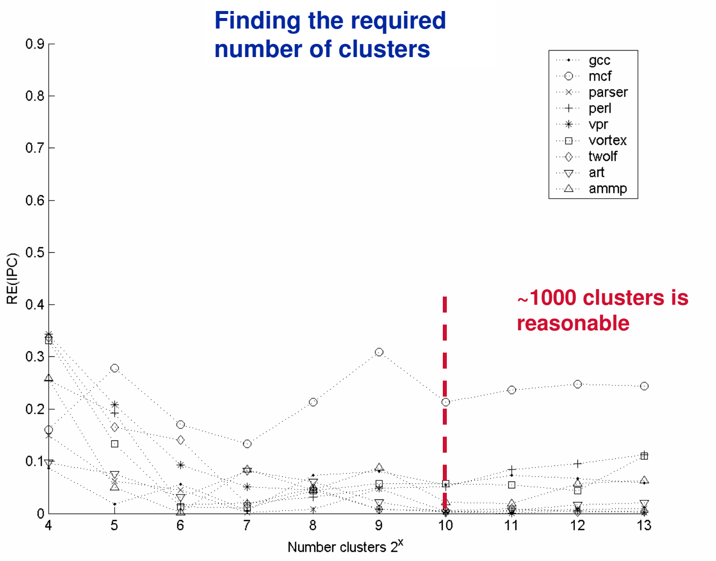
$$\text{standard deviation,} \quad s_{\text{metric}} = \sqrt{\frac{\sum_{i=1}^{N_{\text{cluster}}} \left(\text{metric}_{\text{cluster}}^{i} - \text{metric}_{\text{sample}}\right)^2}{(N_{\text{cluster}} - 1)}}$$

Sampling bias and variability

- Sampling bias reduced via three parameters:
  - Cluster size
  - Number of clusters
  - Overall sample size = Cluster size X Number of clusters
- Sampling variability improved via random sampling
  - Gaps between clusters are selected using random variable of uniform distribution
  - If you get this wrong, error bars may be too tight

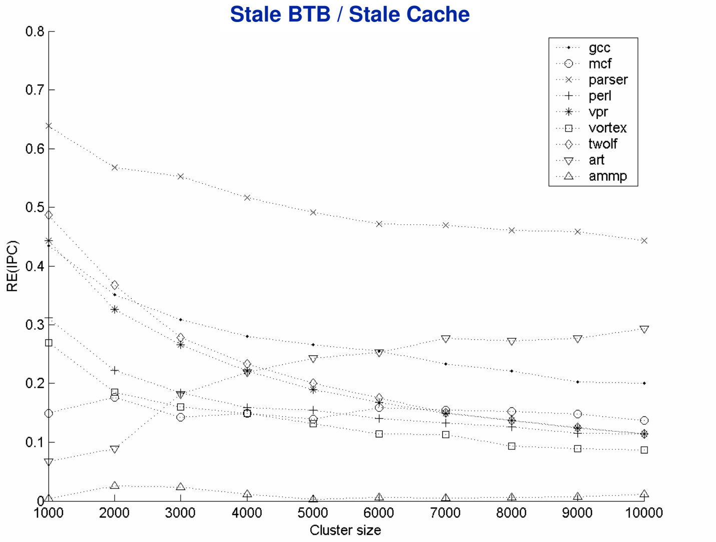
Example

- SPEC CPU Cint2000
- 4-issue, 64-entry window
- L1: 32KB, 4-way, L2: 1MB, 8-way
- Memory bus contention modeled
- BTB: 64k-entry gshare, 1k-entry ret addr stack
- Four nonsampling bias removal choices:
  - Leave BTB stale / simulate it during the gap (warm)
  - Leave caches stale / simulate it during the gap (warm)



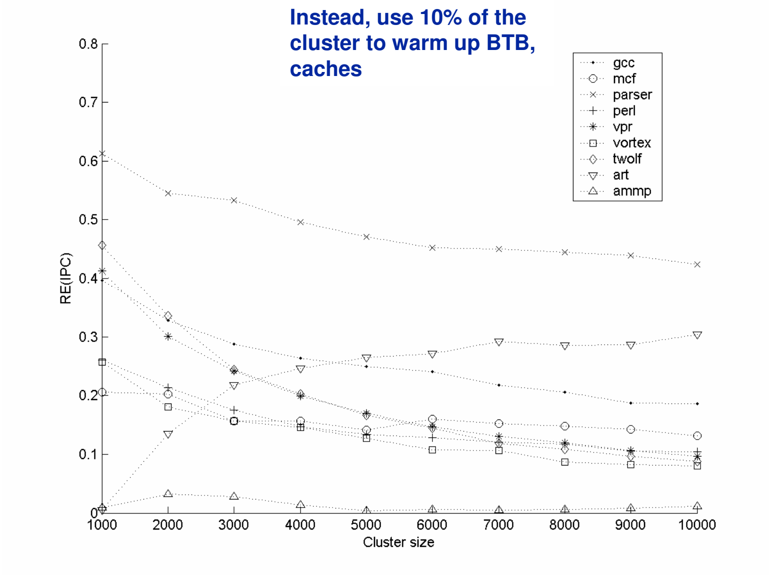
Finding the required number of clusters

~1000 clusters is reasonable

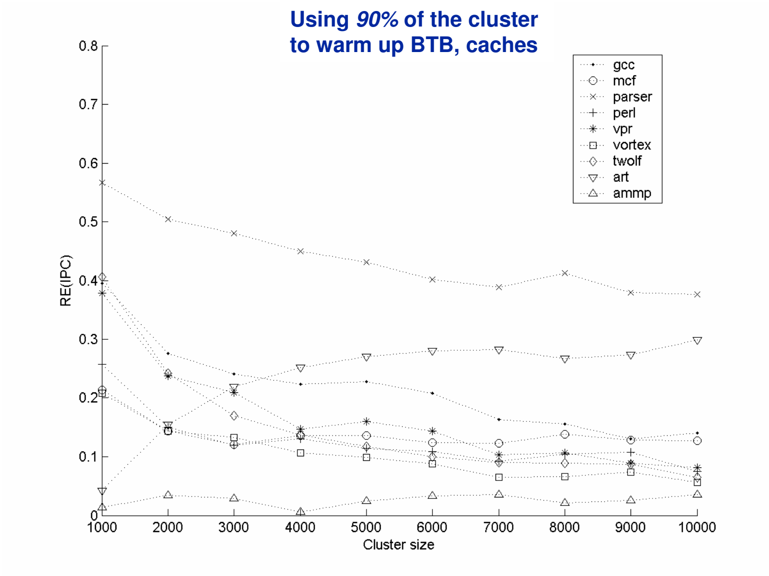


Stale BTB / Stale Cache

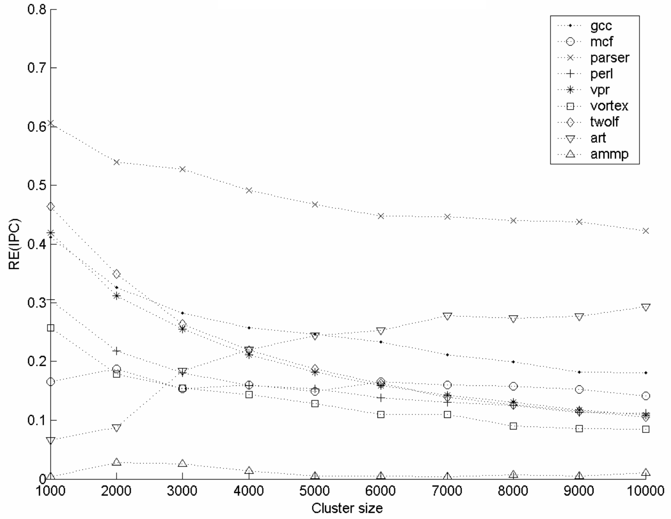
Instead, use 10% of the cluster to warm up BTB, caches



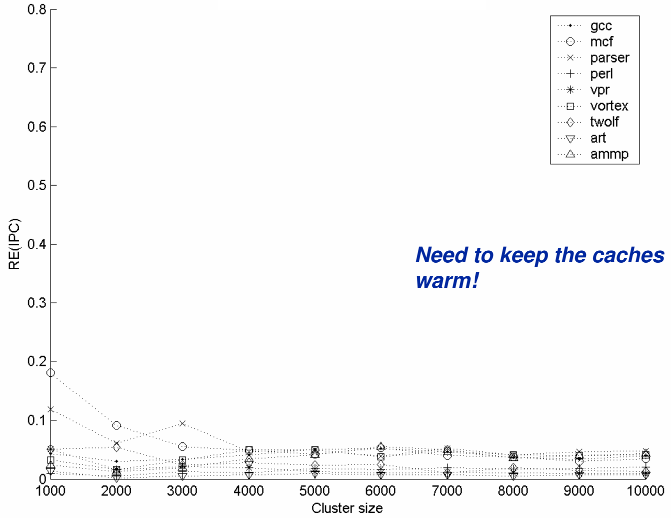
Using *90%* of the cluster to warm up BTB, caches



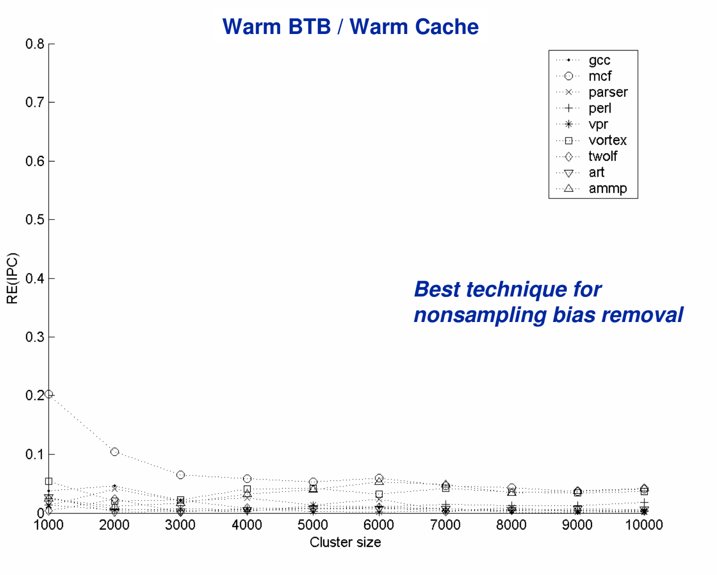
Warm BTB / Stale Cache



Stale BTB / Warm Cache



Need to keep the caches warm!



Best technique for nonsampling bias removal

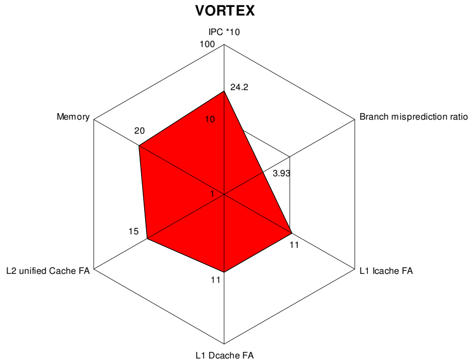
Finding the right clustersize: Do the statistics predict the actual error?

yes if sampled +/- CI = actual

Cluster size	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
gcc	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
mcf	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
parser	yes	no	yes	no	yes	no	yes	yes	yes	yes
perl	yes	yes	yes	yes	no	no	no	no	no	no
vortex	no	yes	yes	no	no	no	no	no	no	no
vpr	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
twolf	yes	no	yes	yes	yes	yes	yes	yes	yes	yes
ammp	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
art	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes

- Clustersize = 3000 …
- (Why is vortex so difficult?)

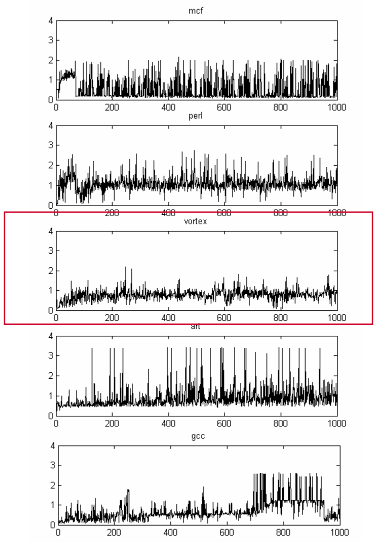
Consider the characteristics

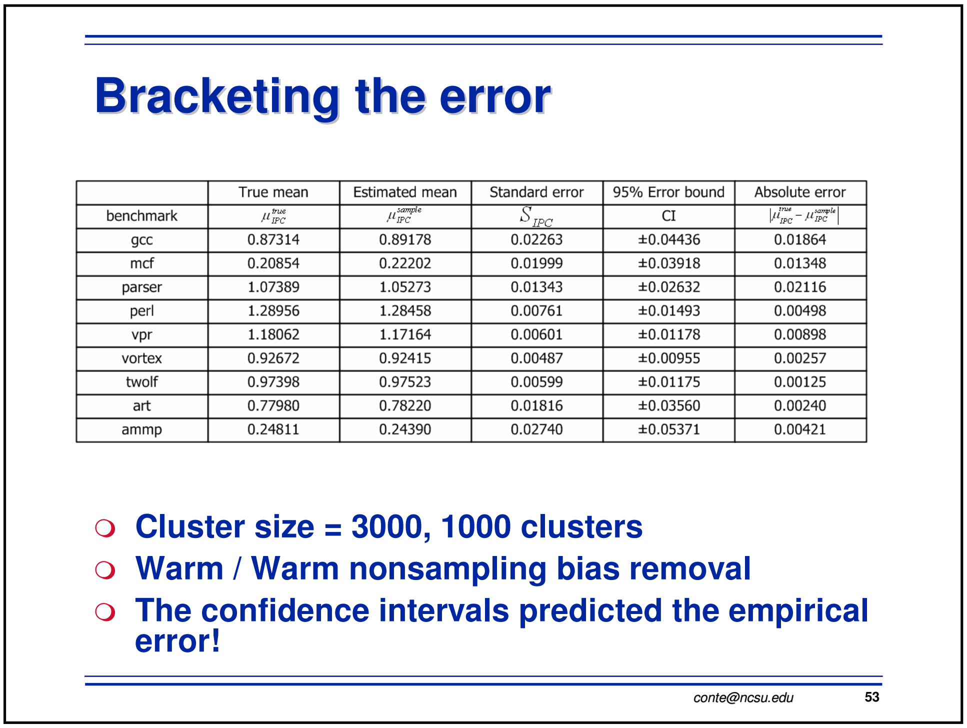


- High branch misprediction rate
- Moderate memory footprint, but it's enough …

But also low cluster variability

- The lower the variability
- …the smaller the standard error
- …the tighter the predicted confidence interval
- Thus more stress is placed on nonsampling bias removal …
- Some benchmarks are tougher than others

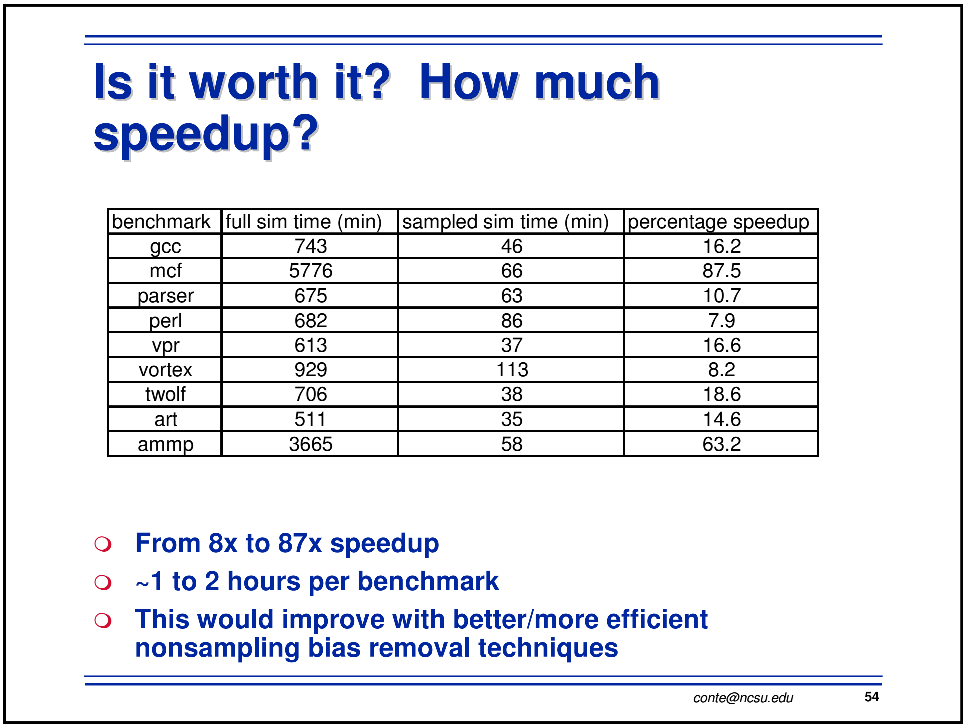




Bracketing the error

	True mean	Estimated mean	Standard error	95% Error bound	Absolute error
benchmark	μ_{IPC}^{true}	μ_{IPC}^{sample}	S_{IPC}	CI	$\lvert\mu_{IPC}^{true} - \mu_{IPC}^{sample}\rvert$
gcc	0.87314	0.89178	0.02263	±0.04436	0.01864
mcf	0.20854	0.22202	0.01999	±0.03918	0.01348
parser	1.07389	1.05273	0.01343	±0.02632	0.02116
perl	1.28956	1.28458	0.00761	±0.01493	0.00498
vpr	1.18062	1.17164	0.00601	±0.01178	0.00898
vortex	0.92672	0.92415	0.00487	±0.00955	0.00257
twolf	0.97398	0.97523	0.00599	±0.01175	0.00125
art	0.77980	0.78220	0.01816	±0.03560	0.00240
ammp	0.24811	0.24390	0.02740	±0.05371	0.00421

- Cluster size = 3000, 1000 clusters
- Warm / Warm nonsampling bias removal
- The confidence intervals predicted the empirical error!

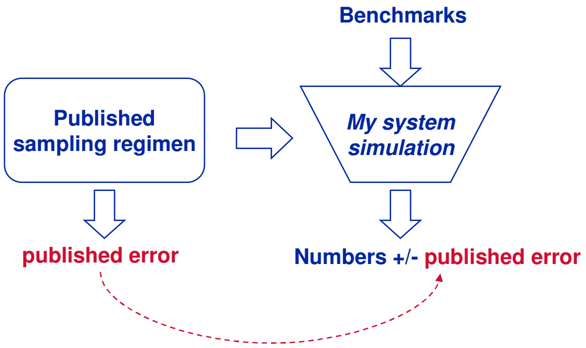


Is it worth it? How much speedup?

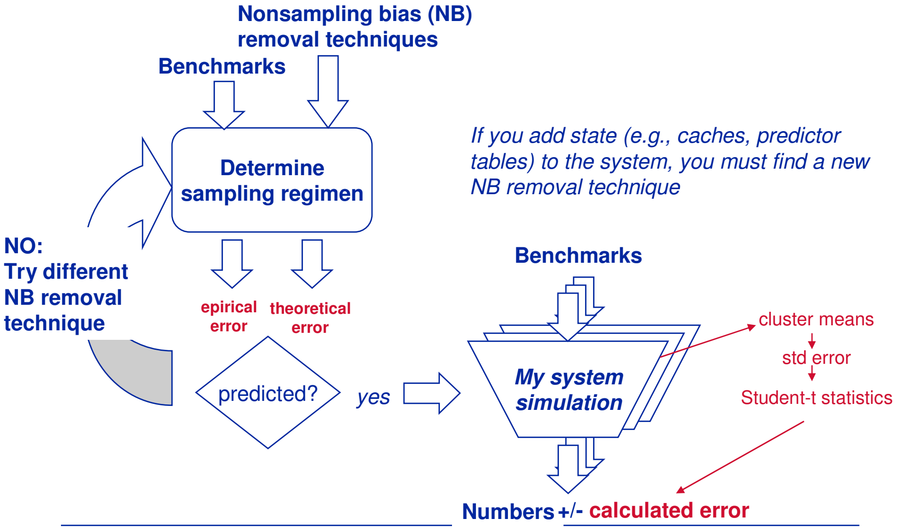
benchmark	full sim time (min)	sampled sim time (min)	percentage speedup
gcc	743	46	16.2
mcf	5776	66	87.5
parser	675	63	10.7
perl	682	86	7.9
vpr	613	37	16.6
vortex	929	113	8.2
twolf	706	38	18.6
art	511	35	14.6
ammp	3665	58	63.2

- From 8x to 87x speedup
- ~1 to 2 hours per benchmark
- This would improve with better/more efficient nonsampling bias removal techniques

We had this: “Trust my error”



Now we have this



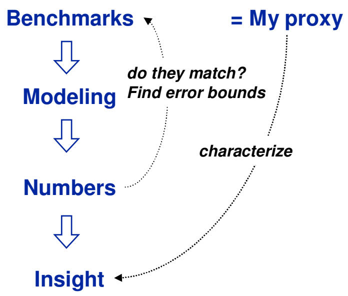
If you add state (e.g., caches, predictor tables) to the system, you must find a new NB removal technique

Sampling last thoughts

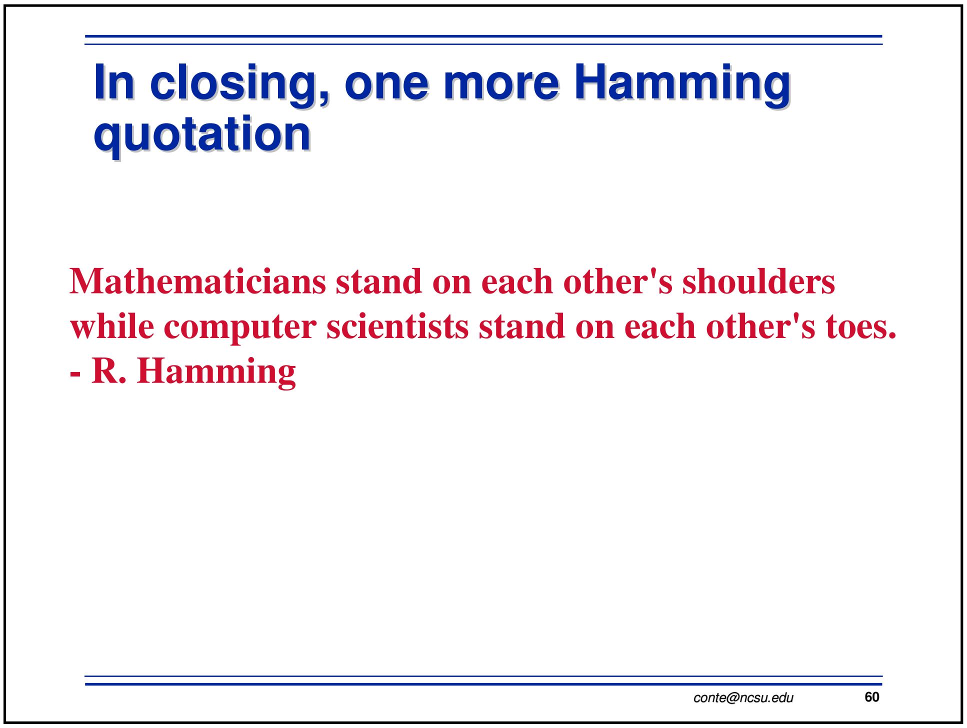
- If regimen developed correctly
  - can use the derived sampling regimen to calculate confidence intervals
- *You know how much you can trust your numbers*
- Much more research is needed into effective nonsampling bias removal techniques
- Should we develop benchmarks just for finding sampling regimens?

- All results should include confidence intervals – even if it makes your gizmo look bad

The road map



- And I think Hamming would be happy with that



In closing, one more Hamming quotation

Mathematicians stand on each other's shoulders while computer scientists stand on each other's toes.
- R. Hamming