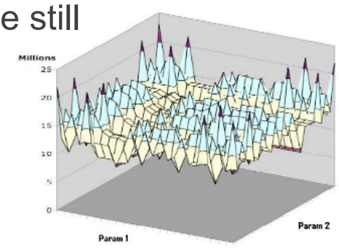
ArchExplorer.org:
Leveraging Modular Simulation to Automate Design-Space Exploration

VEERLE DESMET SYLVAIN GIRBAL **OLIVIER TEMAM**
Ghent University *INRIA & Thales TRT* ***INRIA***
Belgium *France* ***France***

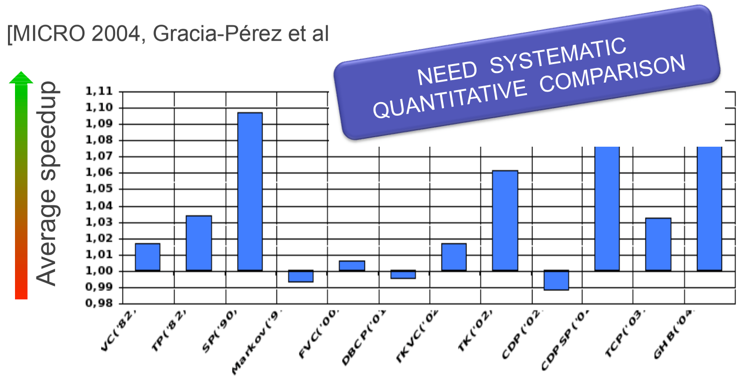
Is there a complexity tipping point ?

- Trend towards greater complexity
- Intuition and experience still best design drivers ?



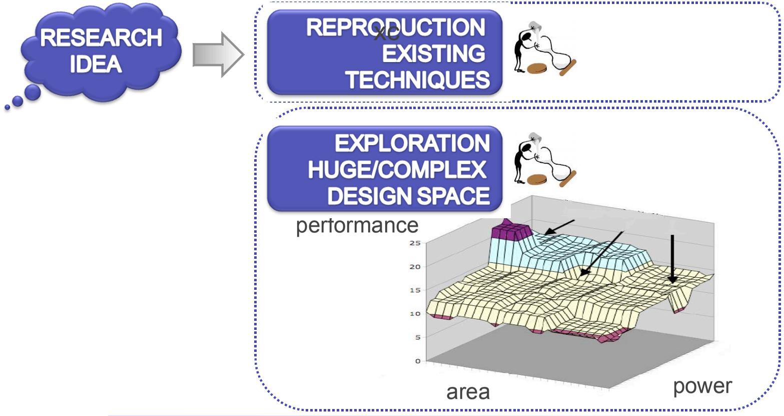
An example

[MICRO 2004, Gracia-Pérez et al



NEED SYSTEMATIC QUANTITATIVE COMPARISON

Why is systematic comparison not (yet) common practice?

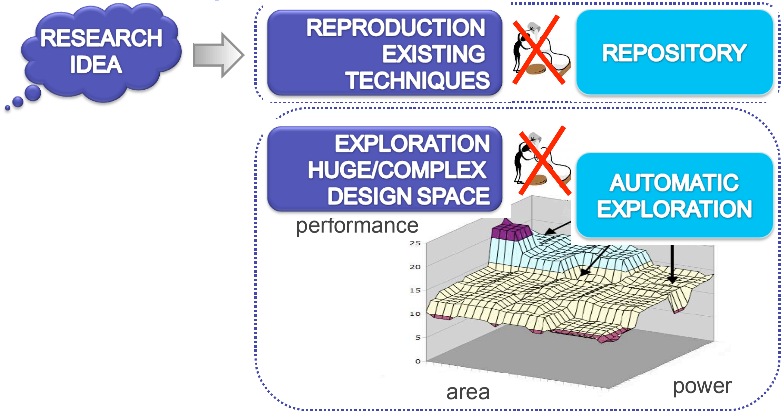


RESEARCH IDEA

REPRODUCTION EXISTING TECHNIQUES

EXPLORATION HUGE/COMPLEX DESIGN SPACE

Why is systematic comparison not (yet) common practice?



RESEARCH IDEA

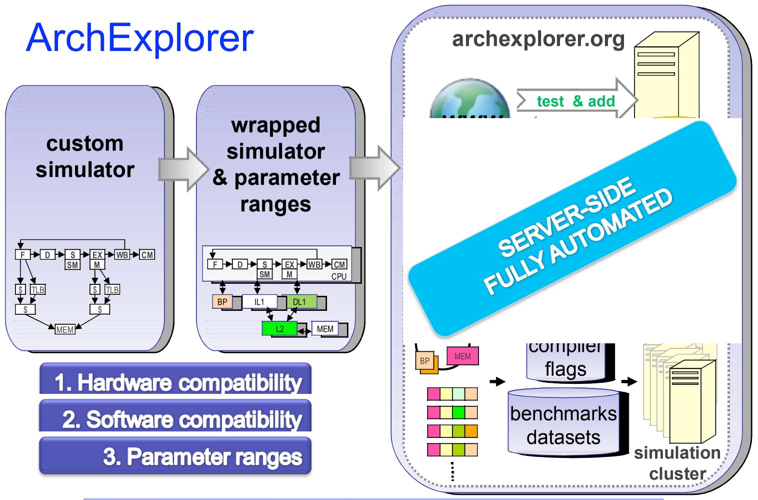
REPRODUCTION EXISTING TECHNIQUES

REPOSITORY

EXPLORATION HUGE/COMPLEX DESIGN SPACE

AUTOMATIC EXPLORATION

ArchExplorer



custom simulator

wrapped simulator & parameter ranges

archexplorer.org

test & add

SERVER-SIDE FULLY AUTOMATED

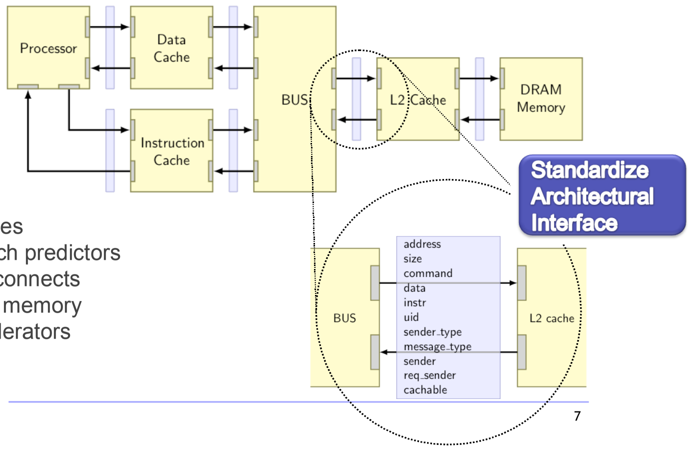
compiler flags

benchmarks datasets

simulation cluster

1. Hardware compatibility
2. Software compatibility
3. Parameter ranges

Step 1: Hardware compatibility



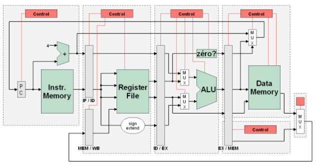
Standardize Architectural Interface

Caches
Branch predictors
Interconnects
Main memory
Accelerators
...

Step 2: Software compatibility

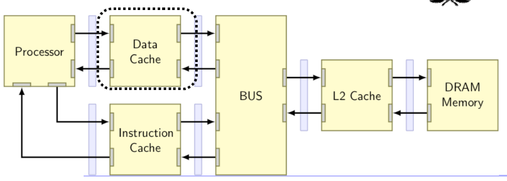
ISOLATE YOUR HARDWARE BLOCK

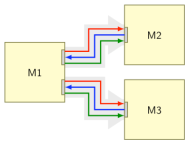
Centralized control vs. distributed control



WRAPPING IN SYSTEM-C + UNISIM

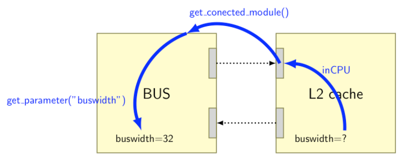
Models of computation





Step 3: Parameter ranges

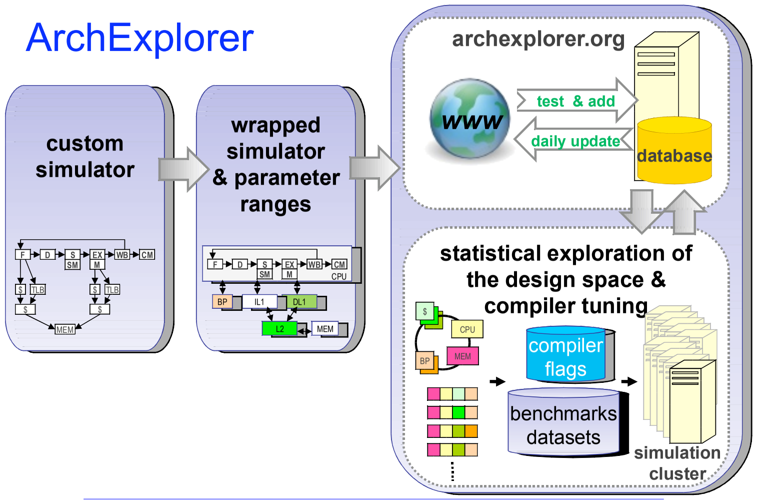
Self-Configuration and parameters validity



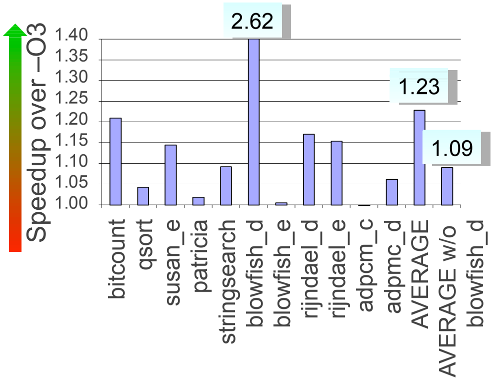
Min, max, range
Power-of-2
Positive
Complex relations

Associativity 1 2 4 8
CacheLines 64 256 1024 4096
LineSize 32 64
...

ArchExplorer



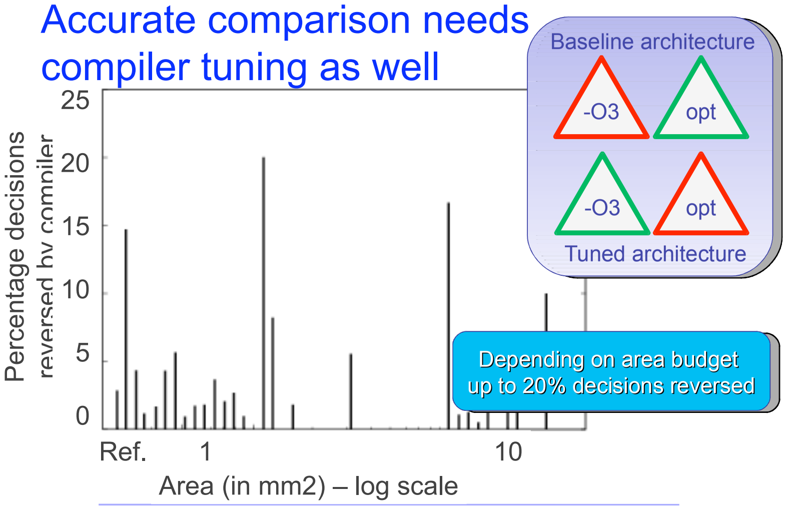
Accurate comparison needs compiler tuning as well



Best Compiler Flags

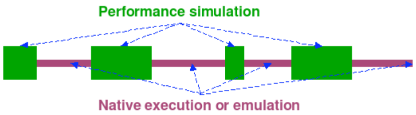
-O1 -falign-jumps=36 -fschedule-insns2 -fsingle-precision-constant
-O3 -falign-functions=56 -falign-jumps=51
-O1 -fno-ivopts -fno-tree-fre
-O0 -fsched-stalled-insns-dep=34 -fsched-stalled-insns=62 -falign-loops=63
-O1 -fno-rerun-cse-after-loop -fschedule-insns2
-O1 -fno-rerun-cse-after-loop -fschedule-insns2
-O1 -falign-functions=37 -falign-jumps=8 -falign-loops=40
-O1 -falign-functions=30 -falign-loops=60 –fregmove
-O1 -fsched-stalled-insns-dep=5 -fgcse-lm -finline-functions-ftree-pre
-O3 -fsched-stalled-insns-dep=54 -falign-loops=47
-O3 -fsched-stalled-insns-dep=5 -fno-tree-lrs

Accurate comparison needs compiler tuning as well



Depending on area budget up to 20% decisions reversed

Fast evaluation

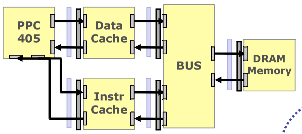


- Fast evaluation of one design point (fast simulation, e.g., sampling)
- Fast selection of design points (e.g., genetic algorithms; see MULTICUBE project also)

Case study

Architecture

PowerPC 405 memory sub-system



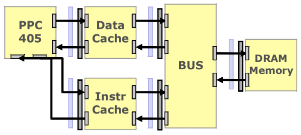
Module repository:
- Victim Cache
- Timekeeping Victim cache
- Stride Prefetcher
- Content-Directed Prefetcher
- Stride + Content Directed Prefetcher
- Tag Prefetcher
- Global History Prefetcher
- Skewed associtiative cache

Compiler

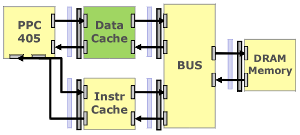
```
-O1 -falign-jumps=36 -fschedule-insns2
-fsingle-precision-constant
-O3 -falign-functions=56 -falign-jumps=51
-O1 -fno-ivopts -fno-tree-fre
-O0 -fsched-stalled-insns-dep=34 -fsched-
stalled-insns=62 -falign-loops=63
-O1 -fno-rerun-cse-after-loop -fschedule-
insns2
-O1 -fno-rerun-cse-after-loop -fschedule-
insns2
-O1 -falign-functions=37 -falign-jumps=8
-falign-loops=40
-O1 -falign-functions=30 -falign-loops=60 –
fregmove
-O1 -fsched-stalled-insns-dep=5 -fgcse-lm
-finline-functions-ftree-pre
-O3 -fsched-stalled-insns-dep=54 -falign-
loops=47
-O3 -fsched-stalled-insns-dep=5 -fno-tree-lrs
```

Design space dimension

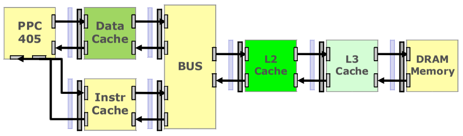
Parametric: 10^6 points



Structural: 10^9 points

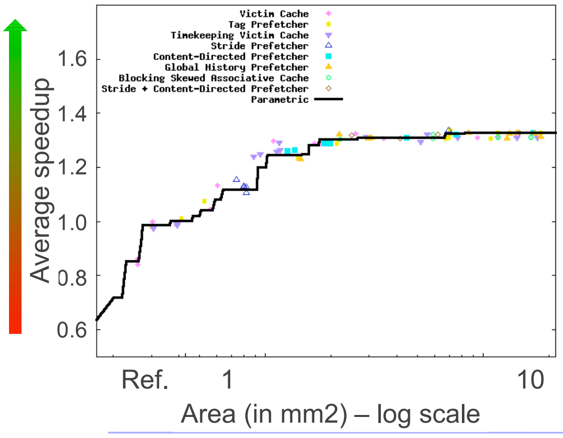


Hierarchies up to 4 levels: 10^24 points



Joint architecture + compiler space: 10^27 points

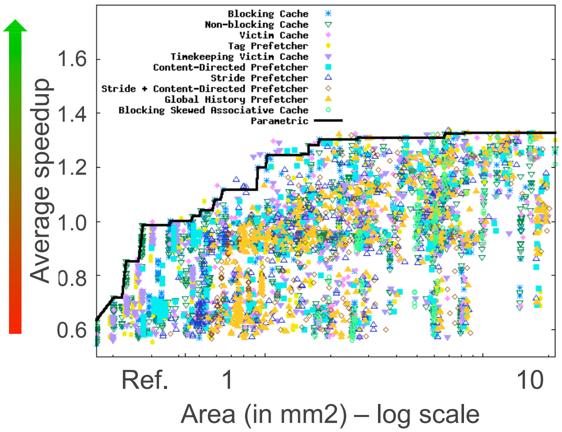
Best memory sub-system per area



CONCLUSIONS:

1. Contrast to Gracia-Pérez et al. [MICRO 2004]
2. No clear winner
3. Close to tuned parametric cache

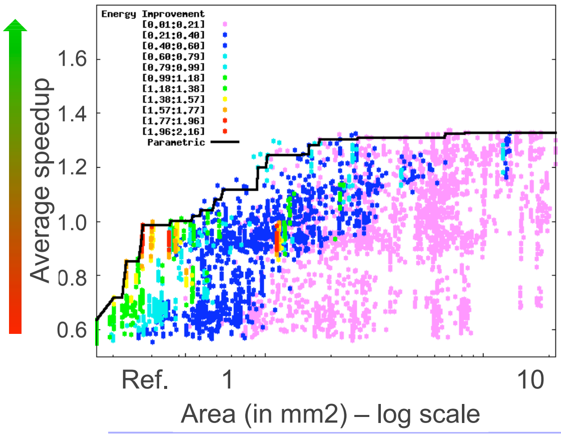
Best memory sub-system per area



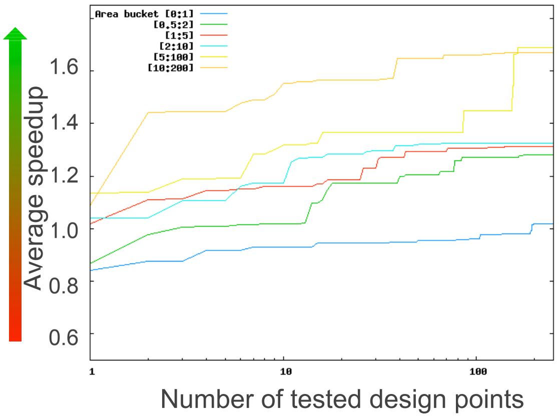
CONCLUSIONS:

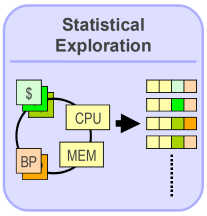
1. Contrast to Gracia-Pérez et al. [MICRO 2004]
2. No clear winner
3. Close to tuned parametric cache

Speedup and energy improvement



Exploration scalability

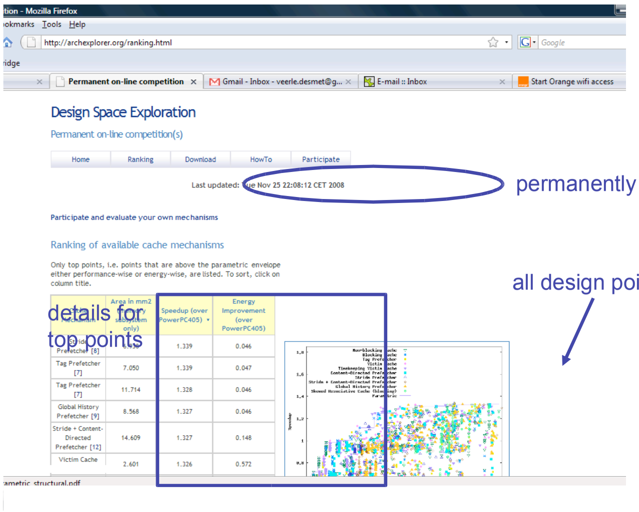




- Permanently ranks all designs
  - per area bucket
  - speedup or power
  - assigning higher probability to better points
- Picking point w.r.t distribution
- Mutations+crossover
- Natural selection

Check out this website:

ARCHEXPLORER.ORG



tion - Mozilla Firefox
okmarks Tools Help
http://archexplorer.org/ranking.html
Google
ridge
Permanent on-line competition
Gmail - Inbox - veerle.desmet@g...
E-mail :: Inbox
Start Orange wifi access

Design Space Exploration

Permanent on-line competition(s)

Last updated: Tue Nov 25 22:08:12 CET 2008

Participate and evaluate your own mechanisms

Ranking of available cache mechanisms

Only top points, i.e. points that are above the parametric envelope either performance-wise or energy-wise, are listed. To sort, click on column title.

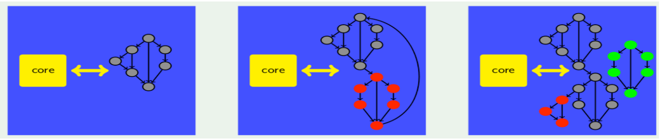
Cache Mechanism	Area in mm2 (memory subsystem only)	Speedup (over PowerPC405) ▾	Energy Improvement (over PowerPC405)
Stride Prefetcher [8]	[illegible]	1.339	0.046
Tag Prefetcher [7]	7.050	1.339	0.047
Tag Prefetcher [7]	11.714	1.328	0.046
Global History Prefetcher [9]	8.568	1.327	0.046
Stride + Content-Directed Prefetcher [12]	14.609	1.327	0.148
Victim Cache	2.601	1.326	0.572

permanently updated

all design points

details for top points

Next step: explorable customization



- Customization “on-demand”
- Automatically generate & aggregate accelerators

Reconciling Specialization and Flexibility with Compound Circuits, S. Yehia, H. Berry, S. Girbal, O. Temam, HPCA 2009, Raleigh, North Carolina

Conclusions & future work

- ArchExplorer.org
  - Facilitates fair quantitative comparison of research ideas
    - repository
    - automatic, joint compiler/hardware design space exploration
- Permanent open competition(s)
    - Data Prefetching Competition (DPC)
- Future work:
    - more micro-architecture exploration
    - more system-level exploration
    - explorable customization