Multicore OS Benchmarks: We Can Do Better

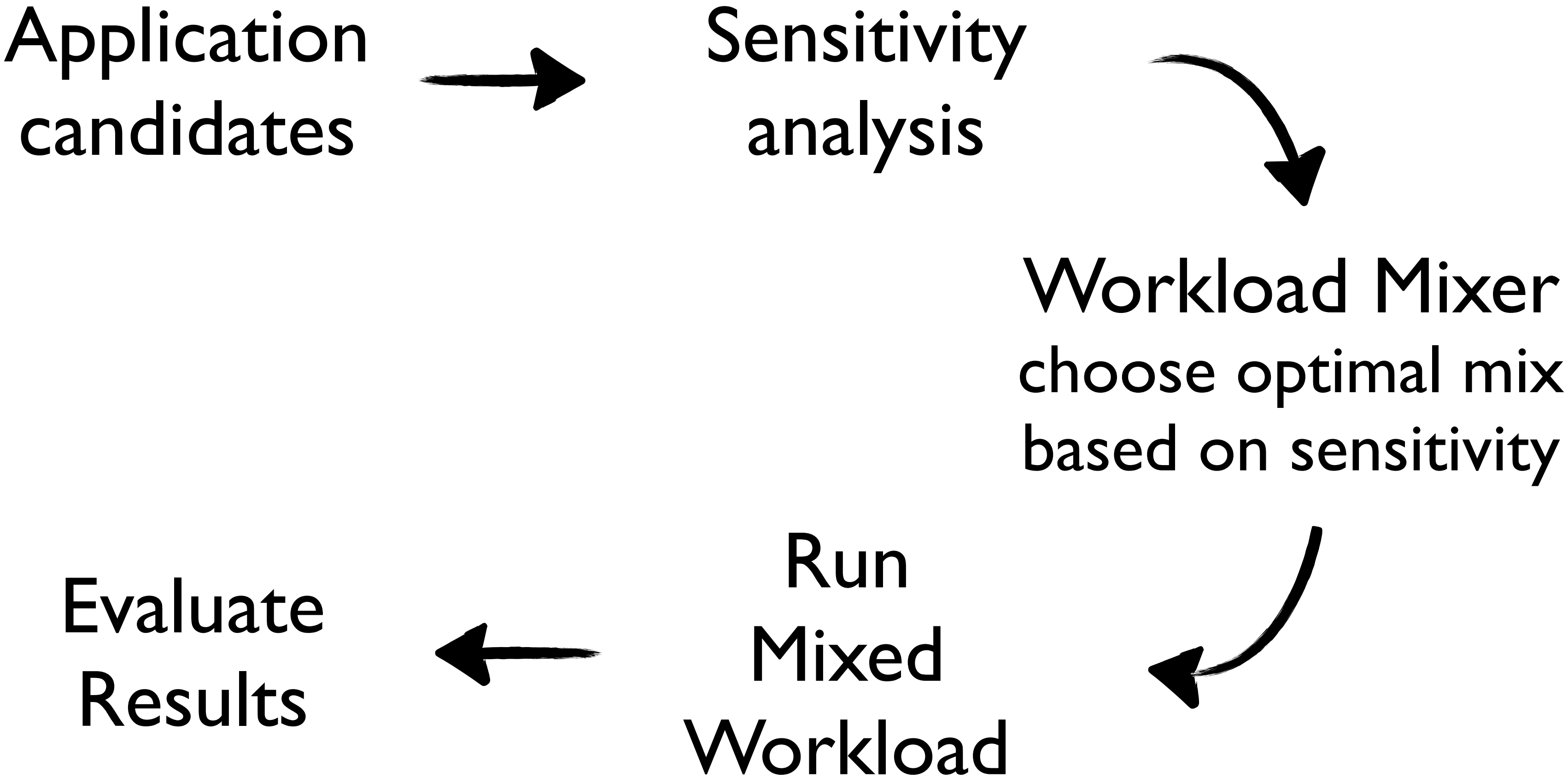
Ihor Kuz*, Zachary Anderson, Pravin Shinde,
Timothy Roscoe
Systems Group, ETH Zurich
* NICTA Australia

Multicore OS benchmarks do not evaluate performance isolation between independent apps.

Mixing Workloads

- Mix must:
  - use system resources
  - not overcommit resources
  - be sensitive to availability of the resources

Mixer Overview



Application candidates

Sensitivity analysis

Workload Mixer
choose optimal mix based on sensitivity

Run Mixed Workload

Evaluate Results

Application Candidates

- Variants
  - Application parameters
  - Resource constraints

- Example:
  - game1: low gfx
  - game2: high gfx

app	CPU	cache	mem	disk	net	score
game1	0.25	0.25	0.25	0.1	0.1	0.25
...	...	...	...	...	...	...
webb1	0.25	0.25	0.1	0.0	0.5	0.2
...	...	...	...	...	...	...
antivN	0.1	0.1	0.1	0.8	0.0	0.8

Sensitivity Analysis

How much resource availability affects goodness score

bmark	CPU	cache	mem	disk	net
game	0.8	0.8	0.6	0.4	0.1
webb	0.8	0.7	0.5	0.1	0.5
antiv	0.2	0.5	0.4	0.8	0.0

Mixing

- Optimization problem based on
  - resource usage
  - resource sensitivity
- 2 parts:
  - Choose application variants that use resources they are most sensitive to
  - Constraint: no resource overcommitted
- ILP problem

Run Mixed Workload

- Mix:

mix = game1, game1,
webb1, antivn

CPU	cach	mem	disk	net
85%	85%	70%	100%	70%

- Running the mix
  - unmixed, mixed
- Benchmark result: unmixed - mixed

Evaluate Results

- Low performance difference: Good!
- Comparison between Operating Systems
  - run different optimal mix for each OS
  - compare results
  - ➡how well each OS manages optimal mix
- OS or hardware platform?

Conclusion

- Current status: microbenchmarks
- Real applications
- Bursty applications
- Dynamic workloads
- Extend not Replace