Hotspot mitigation for the masses

Fabien Hermenier, Aditya Ramesh, Abhinay Nagpal, Himanshu Nagpal, Ramesh Chandra

NUTANIX™

ACM Symposium On Cloud Computing 2019

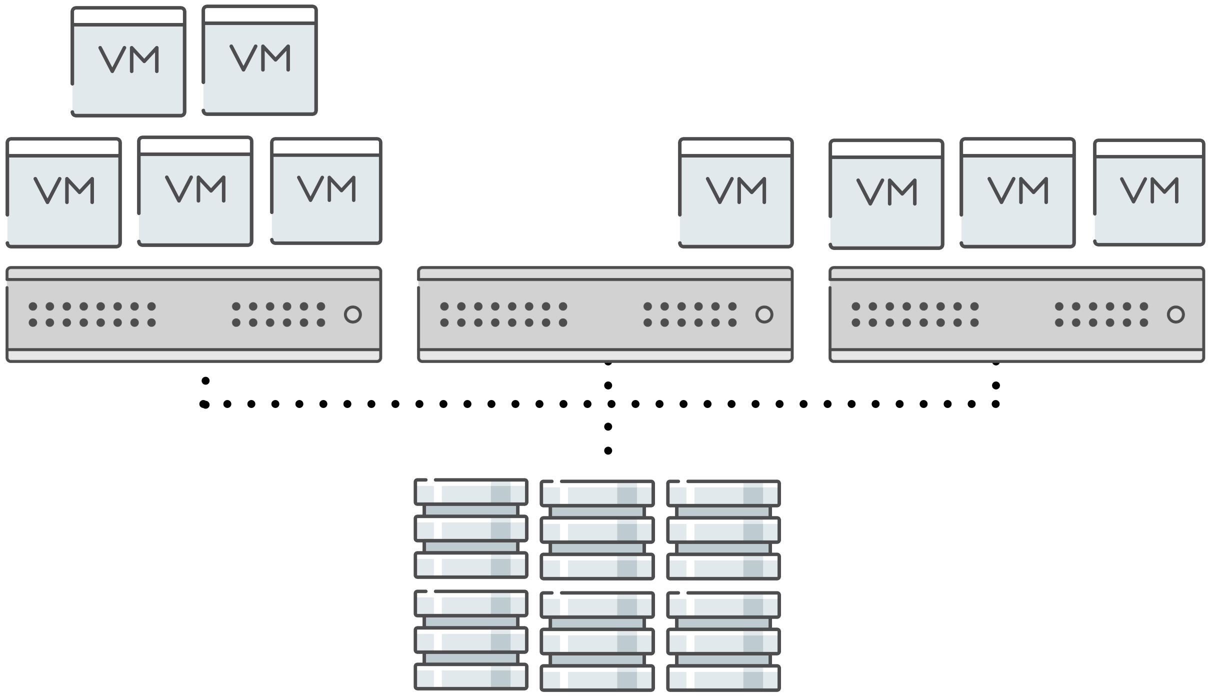
NUTANIX™

Entreprise cloud company

~ 15,000 customers worldwide

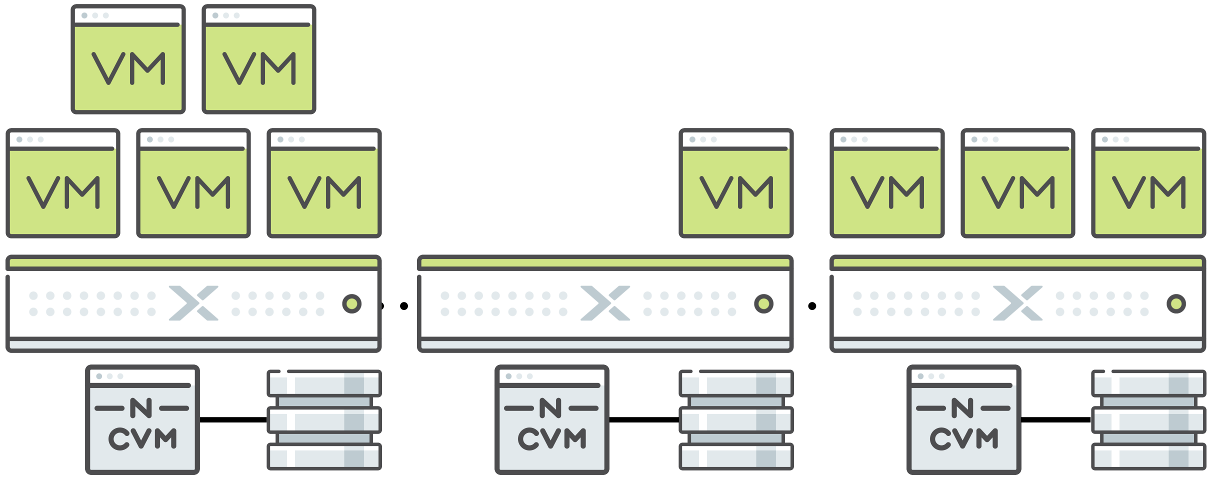
~ 40,000 private clouds deployments

Private clouds



From converged

SAN based, remote I/Os



to hyper-converged infrastructures (HCI)

Distributed file-system favouring local I/Os,
one controller VM per node

602 private clouds

small clusters and beefy nodes fit SMB needs

~ 4 node clusters, 13 VMs per node
long tail distribution

oversubscribed cores

~ 1.31:1 vCPU/thread, up to 9:1

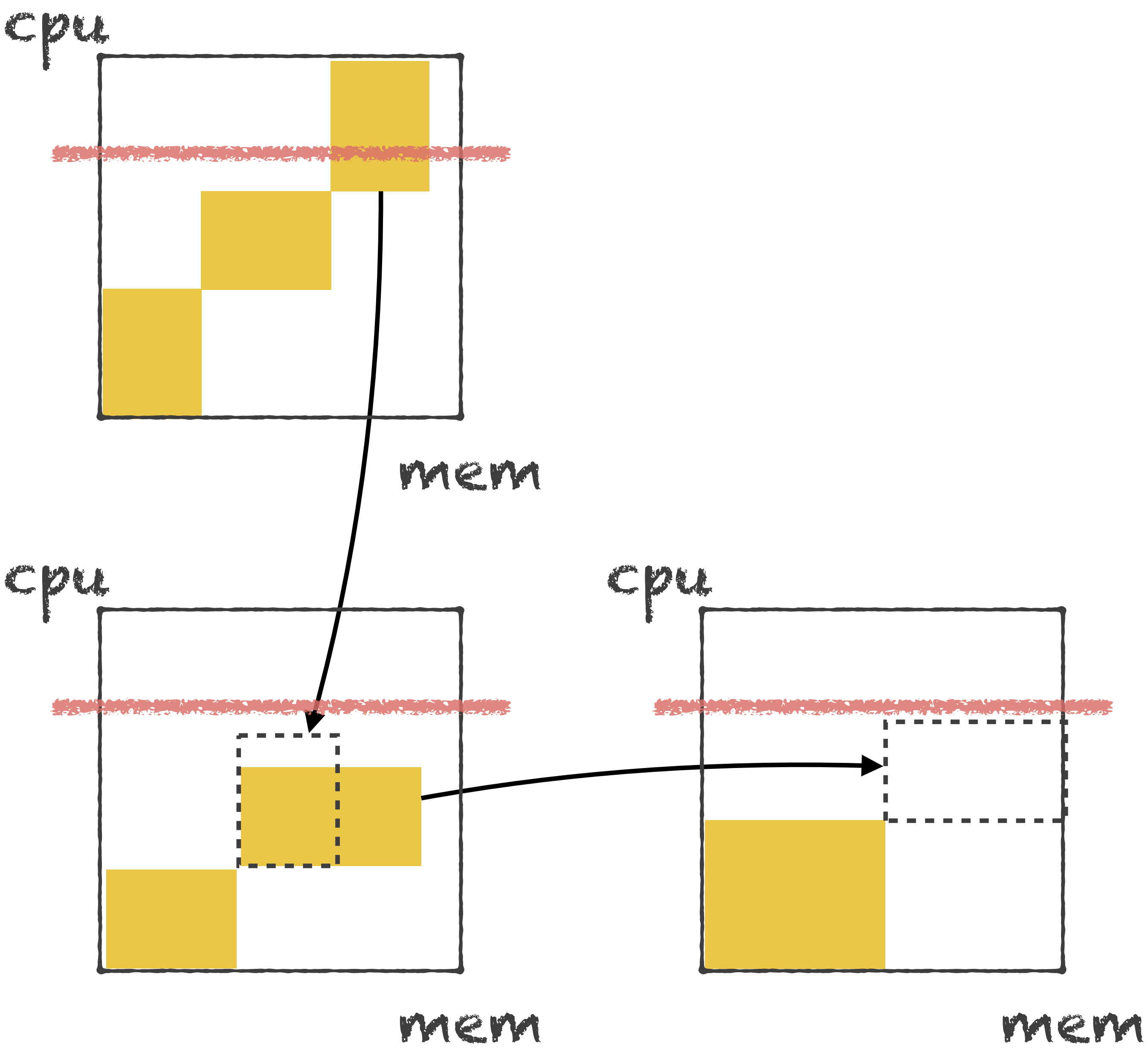
moderate load

~25% CPU, ~2% I/Os (dynamic allocation)
~44% memory (static allocation)

no relationship between dimensions

see the distributions in the paper

Acropolis Dynamic Scheduler (ADS)

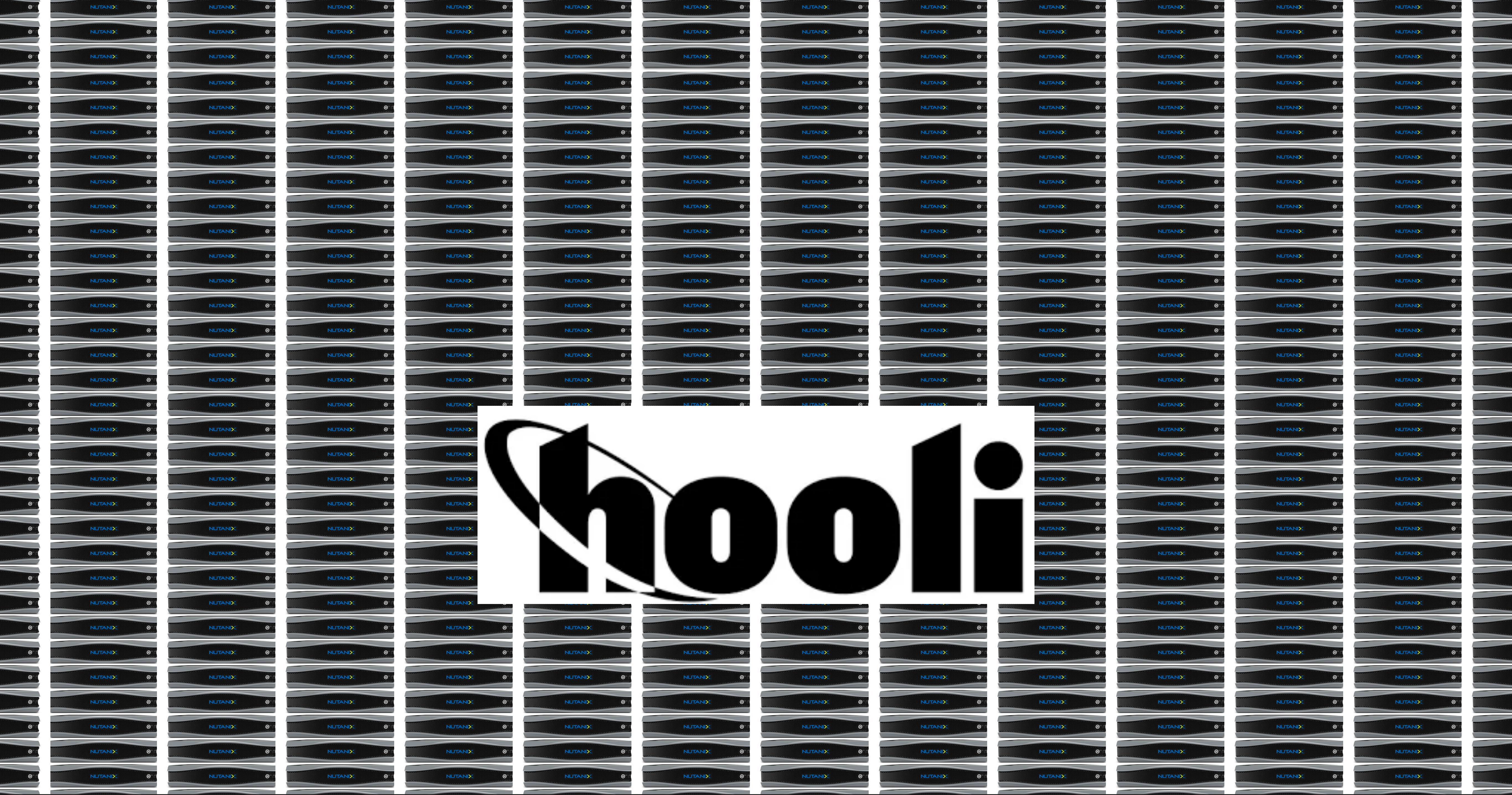


Fix hotspots induced by dynamic resources allocation

Cron based
Threshold based

NP-hard
No holy grail

Scheduler specialisation may alter its applicability



Doing great for the 1%

Doing ok for the 99%

Inside ADS

Exact approach on top of BtrPlace

Constraint programming backend CHOCO

Resource model

- Consumptions retrieved from monitoring system
- Resource demand is a projection plus conditional scale-up
- Storage controller CPU usage as a proxy for I/O usage

Objective

- Minimise data movement
- *Tend* to balance

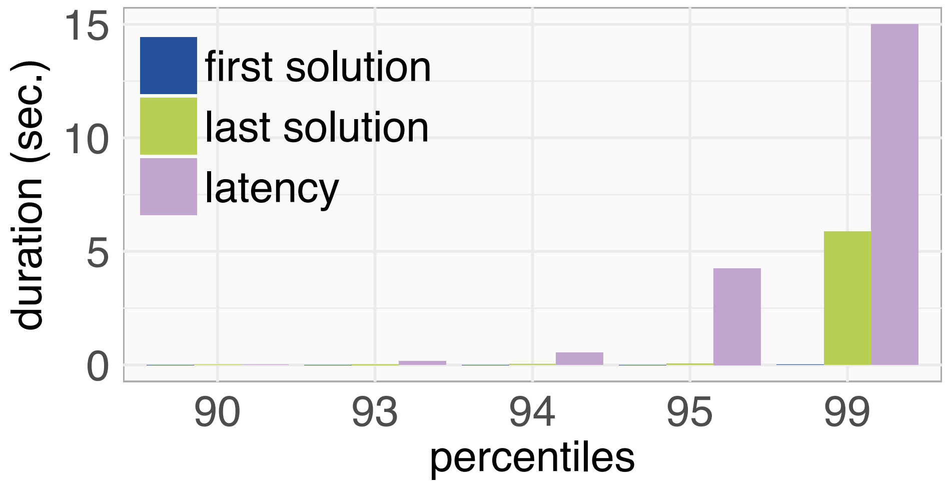
Actuation

- VM migrations (up to 2 in parallel)
- Admin notification upon no solutions

Lessons learnt

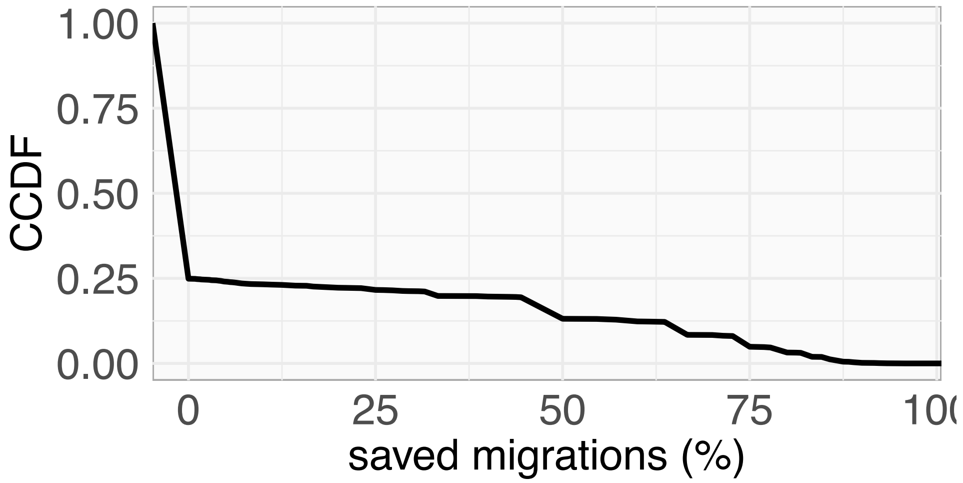
Looking at 2,668 clusters that called ADS at least once

Working with an exact approach

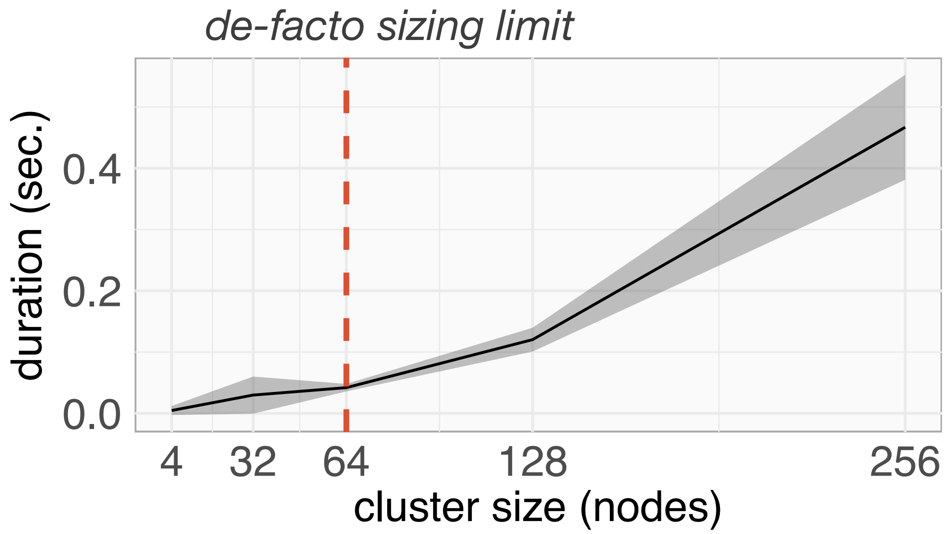


Service latency is good enough

0.5% undecidable problems



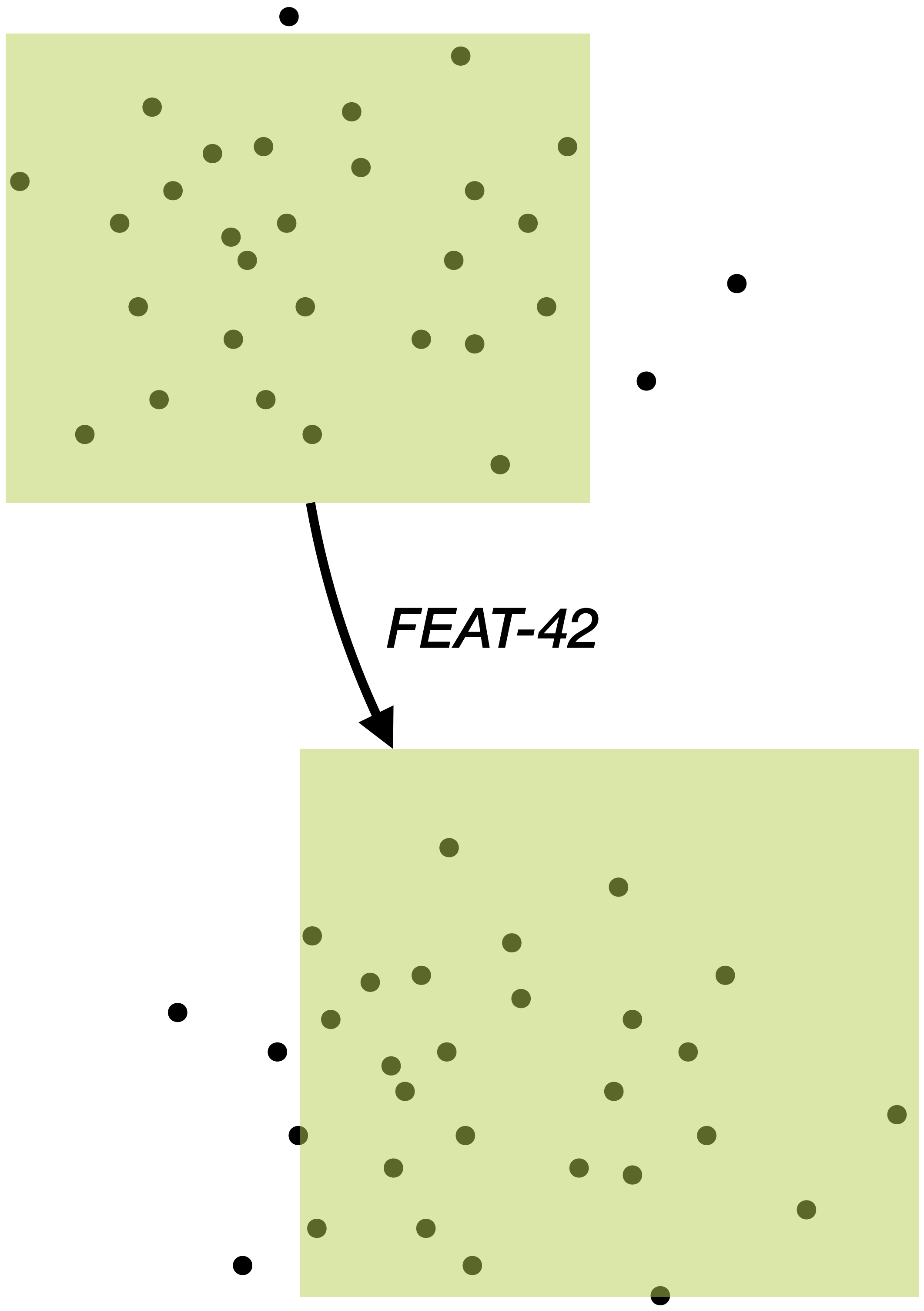
Continuous search helps yield better mitigation plans



Scale beyond sizing limits

In the paper: engineering particularities

Looking for workload agnostic optimisations



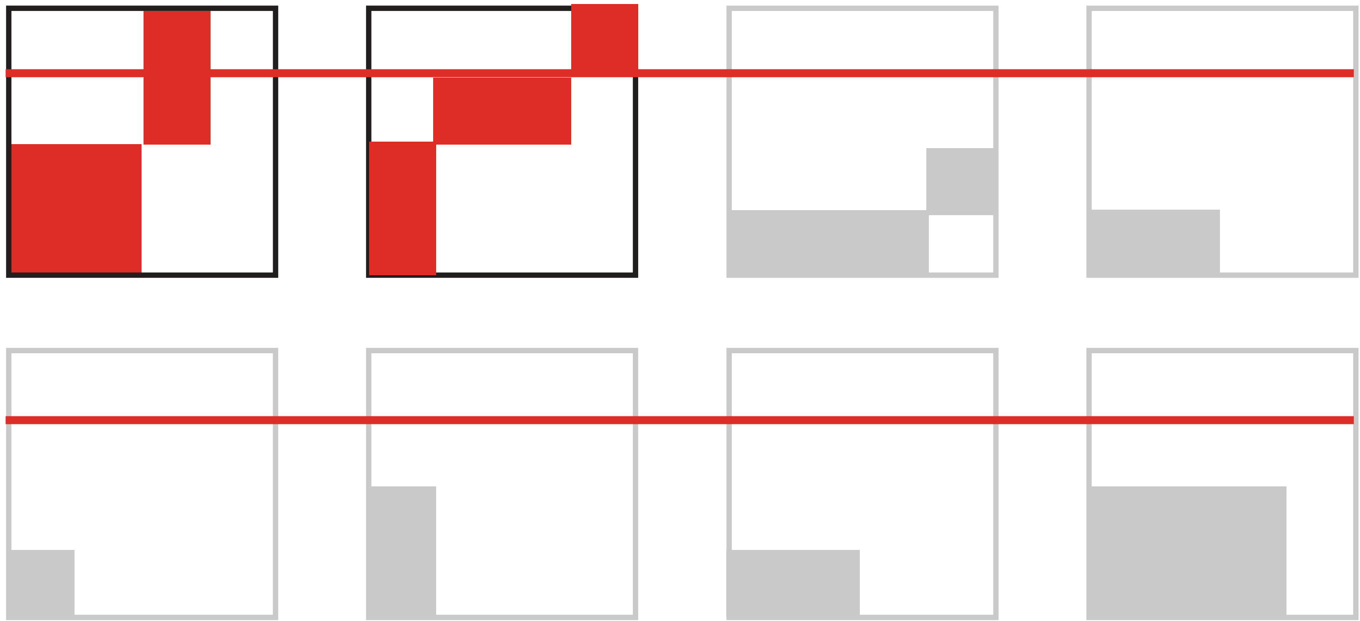
Still NP-hard, still no holy grail

Optimise to reduce undecidable rate, migrations

Beware of false quick wins

The dataset bias dilemma

Local search to reduce the problem size



Low overall load, local hotspots.

Manage only supposed mis-placed VMs

Pin “well placed VM”

Available in BtrPlace

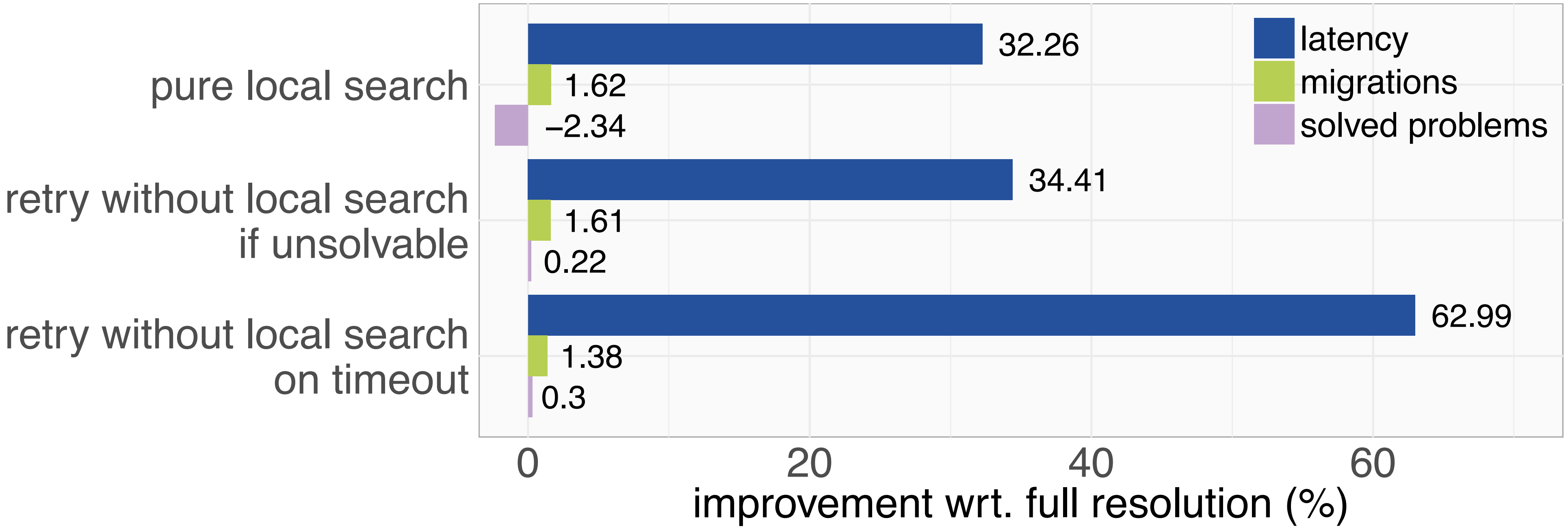
Enabled in ADS 1.0 during the prototyping phase

Local search considered useful and harmful

Over-filtering issues reported

Moved to a 2-phases resolution

- Local search enabled, then disabled if needed
- Trigger reconsidered over time



Practical effectiveness

Complex to analyse without a/b testing

The success rate is a consequence of subjective modelling choices

How many clusters in a clean state after a call to ADS ?

73.28%

if ADS issues a plan

12.24%

If unsolvable

Conclusion

It is about supporting diverse workload

Incremental improvements from observation
small wins matter

Not all enhancements are safe

Trading quality for capability

It is not about developing a new feature,
it is about checking its side effects

Tools and knowledge bases are crucial

Exhibit and characterise outliers
Tests changes to detect regressions