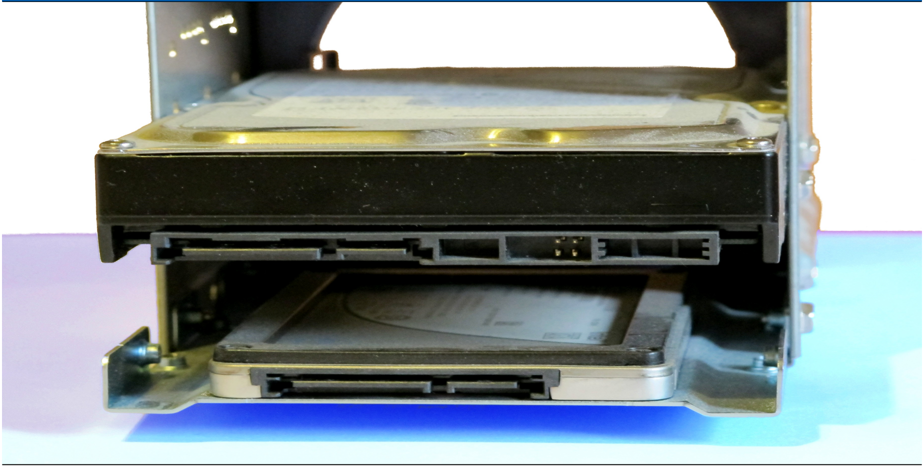
Making Cost-Based Query Optimization Asymmetry-Aware

Daniel Bausch, Ilia Petrov, and Alejandro Buchmann
{bausch, petrov, buchmann}@dvs.tu-darmstadt.de

TECHNISCHE UNIVERSITÄT DARMSTADT



Asymmetry in new storage devices

TECHNISCHE UNIVERSITÄT DARMSTADT

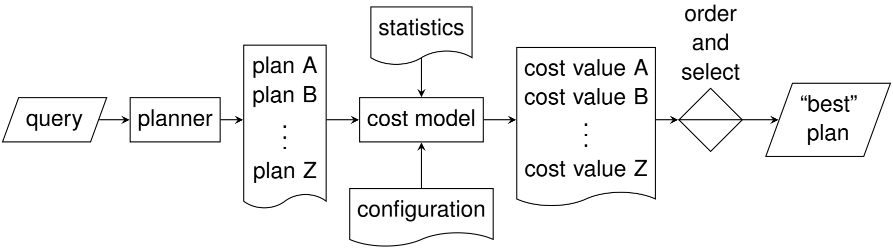
- Writing to Flash memory is slower than reading from it
- This also applies to emerging non-volatile memories (PCM, etc.)
- Small writes to random locations on Flash are even more slow
- Random reads from Flash are only $\frac{1}{3}$ slower than sequential reads[1]

[1] on Intel X25-E using full command queue

Cost-Based Query Optimization

- estimation of run-time before real execution (e.g. in PostgreSQL)
- model comprised of functions like

$$c(seqscan) = \underbrace{\mathbf{c_s}\,\|R\|_\mathrm{p}}_{\text{I/O cost}} + \underbrace{q_{R,0} + (\dot{\mathbf{c}}_\mathbf{cpu} + \dot{q}_R)\,\|R\|_\mathrm{t}}_{\text{CPU cost}}$$



Splitting Parameters

TECHNISCHE UNIVERSITÄT DARMSTADT

$$
\text{sequential page accesses} \quad \mathbf{c_s} \begin{cases} \nearrow \ \mathbf{c_{sr}} & \text{sequential page read accesses} \\ \searrow \ \mathbf{c_{sw}} & \text{sequential page write accesses} \end{cases}
$$

$$
\text{random page accesses} \quad \mathbf{c_r} \begin{cases} \nearrow \ \mathbf{c_{rr}} & \text{random page read accesses} \\ \searrow \ \mathbf{c_{rw}} & \text{random page write accesses} \end{cases}
$$

Cost functions for “pure load” algorithms

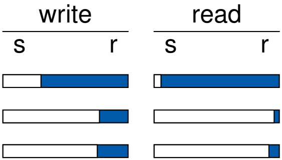
TECHNISCHE UNIVERSITÄT DARMSTADT

cost function	kind	original	replacement
sequential scan	read only	c_s	c_{sr}
index scan	read only	c_s	c_{sr}
		c_r	c_{rr}
bitmap scan	read only	c_s	c_{sr}
		c_r	c_{rr}
TID scan	read only	c_r	c_{rr} e
materialization	write only	c_s	c_{sw}
re-scan	read only	c_s	c_{sr}

Cost function of sort algorithm

$$c_{io}(sort) = \overbrace{2\,\|S\|_p \left\lceil log_m n\right\rceil \left(\tfrac{3}{4}\mathbf{c_s} + \tfrac{1}{4}\mathbf{c_r}\right)}^{\text{startup}}$$

blktrace stats



external sort of unordered data

external sort of ordered data

sort-merge join

$$c_{io}(sort)_{rw} = \underbrace{\|S\|_p \left(\mathbf{c_{sw}} + \left(\left\lceil log_m n\right\rceil - 1\right)\frac{\mathbf{c_{sw}}+\mathbf{c_{rw}}}{2} + \left\lceil log_m n\right\rceil \frac{\mathbf{c_{sr}}+\mathbf{c_{rr}}}{2}\right)}_{\text{startup}}$$

Cost function of hash join

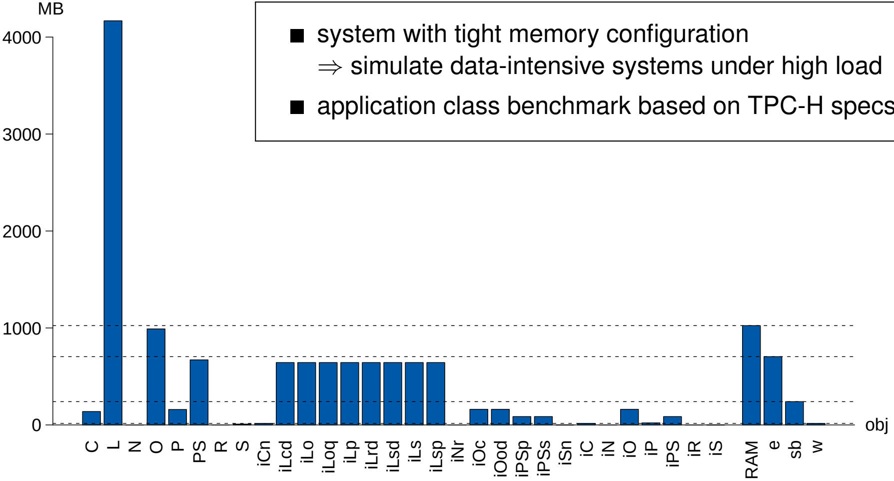
TECHNISCHE UNIVERSITÄT DARMSTADT

$$c_{\text{io}}(\textit{hashjoin}) = \overbrace{\|P_i\|_{\text{p}}\, \mathbf{c_s}}^{\text{startup}} + \left(\|P_i\|_{\text{p}} + 2\,\|P_o\|_{\text{p}}\right) \mathbf{c_s}$$

blktrace stats	write		read	
	s	r	s	r
hash join				

$$c_{\text{io}}(\textit{hashjoin})_{rw} = \underbrace{\|P_i\|_{\text{p}}\, \mathbf{c_{rw}}}_{\text{startup}} + \|P_i\|_{\text{p}}\, \mathbf{c_{sr}} + \|P_o\|_{\text{p}}\, (\mathbf{c_{rw}} + \mathbf{c_{sr}})$$

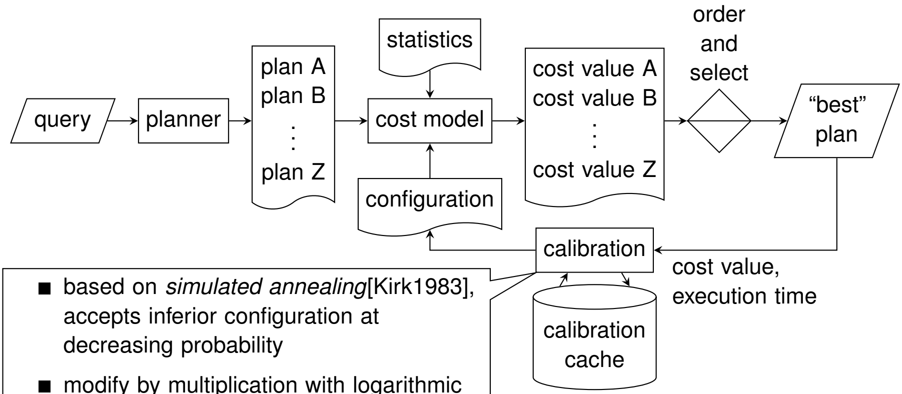
System and Load



- system with tight memory configuration
  $\Rightarrow$ simulate data-intensive systems under high load
- application class benchmark based on TPC-H specs

Calibration

TECHNISCHE UNIVERSITÄT DARMSTADT

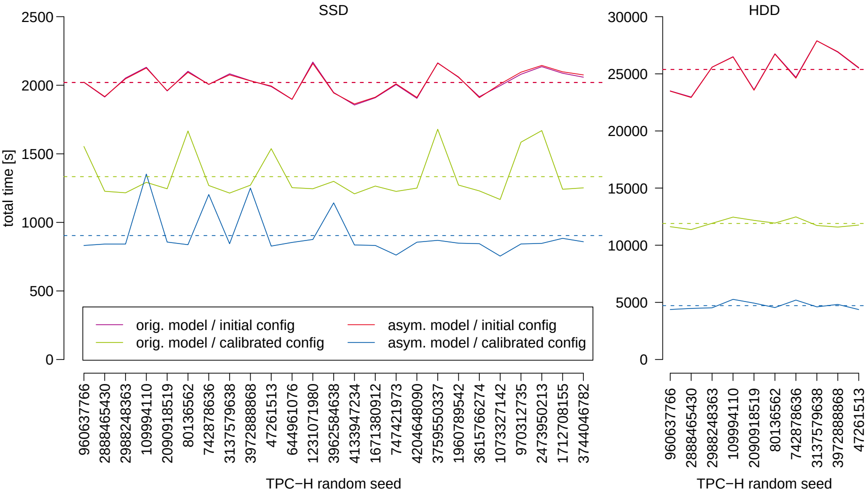


- based on *simulated annealing*[Kirk1983], accepts inferior configuration at decreasing probability
- modify by multiplication with logarithmic normally distributed random variable
- cooling cycle of 100 iterations, restarted 100 times

Found “Optimal” Settings

SSD		HDD	
original model	asymmetric model	original model	asymmetric model
$c_s = 1.00000$	$c_{sr} = 1.00000$	$c_s = 1.00000$	$c_{sr} = 1.00000$
	$c_{sw} = 49.91840$		$c_{sw} = 110.21139$
$c_r = 6.77405$	$c_{rr} = 5.62724$	$c_r = 29.04790$	$c_{rr} = 19.25494$
	$c_{rw} = 19.08421$		$c_{rw} = 20.18467$
$\dot{c}_{cpu} = 0.00121$	$\dot{c}_{cpu} = 0.00003$	$\dot{c}_{cpu} = 0.00280$	$\dot{c}_{cpu} = 0.00082$
$\dot{\hat{c}}_{cpu} = 0.03658$	$\dot{\hat{c}}_{cpu} = 0.01608$	$\dot{\hat{c}}_{cpu} = 0.03718$	$\dot{\hat{c}}_{cpu} = 0.00045$
$c_{op} = 0.00016$	$c_{op} = 0.00008$	$c_{op} = 0.00004$	$c_{op} = 0.00119$

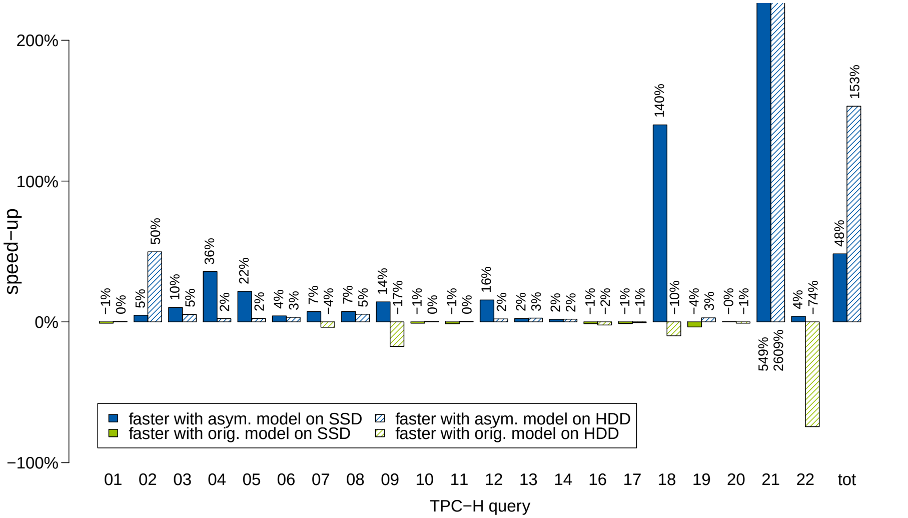
Comparison



SSD
HDD
total time [s]
2500
2000
1500
1000
500
0
30000
25000
20000
15000
10000
5000
0
orig. model / initial config
asym. model / initial config
orig. model / calibrated config
asym. model / calibrated config
960637766
2888465430
2988248363
109994110
2090918519
80136562
742878636
3137579638
3972888868
47261513
644961076
1231071980
3962584638
4133947234
1671380912
747421973
4204648090
3759550337
1960789542
3615766274
1073327142
970312735
2473950213
1712708155
3744046782
TPC−H random seed
TPC−H random seed

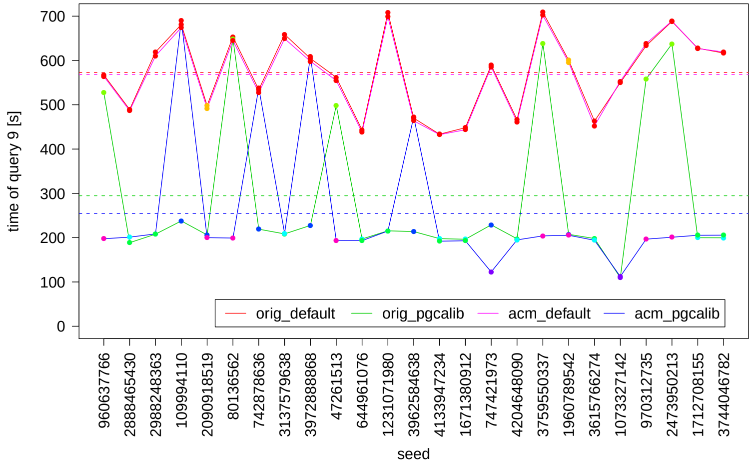
Individual Query Speed-Up

TECHNISCHE UNIVERSITÄT DARMSTADT

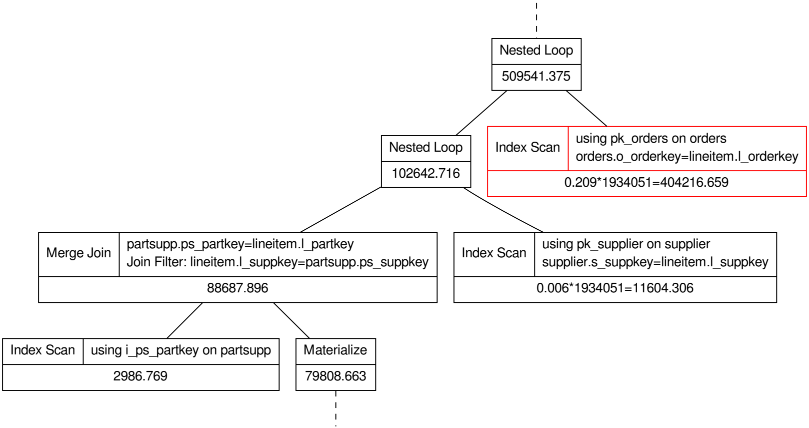


Discussion – Cause for Peaks ⇒ Query 9

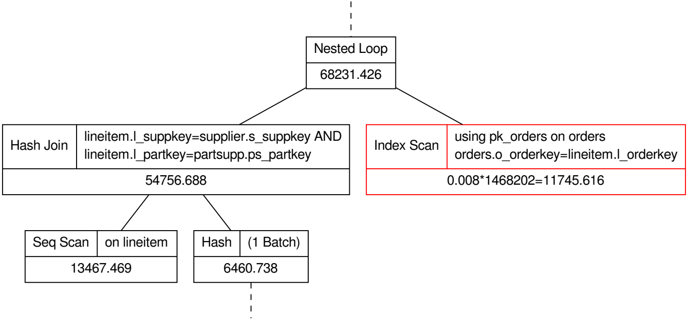
TECHNISCHE UNIVERSITÄT DARMSTADT



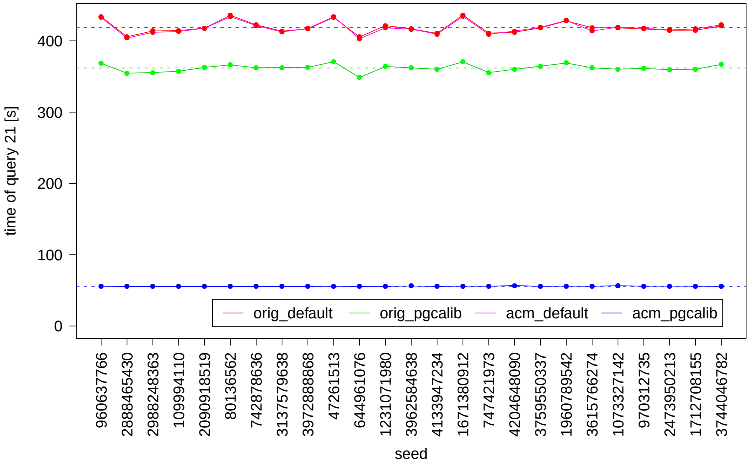
Discussion – Query 9 – the slow plan



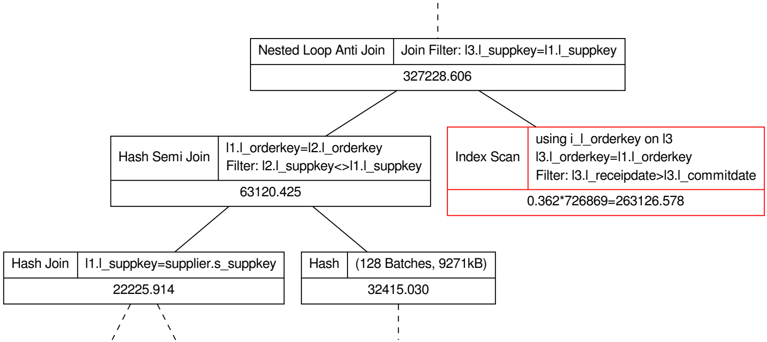
Discussion – Query 9 – the fast plan



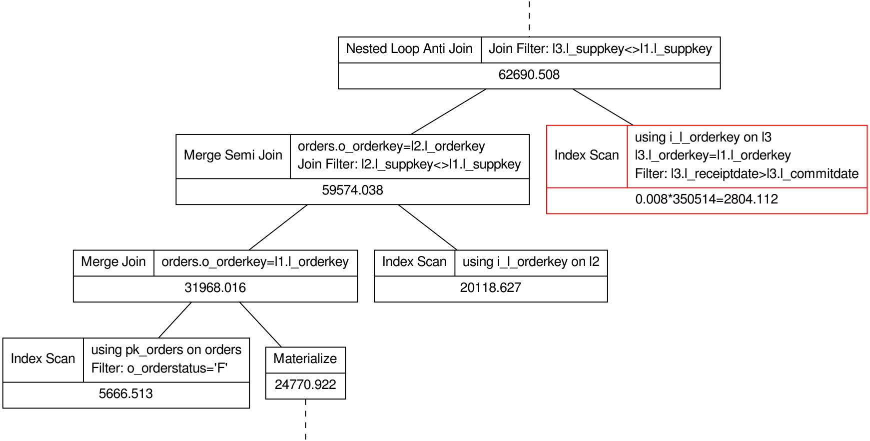
Discussion – Biggest Improvement ⇒ Query 21



Discussion – Query 21 – the slow plan



Discussion – Query 21 – the fast plan



Conclusion

TECHNISCHE UNIVERSITÄT DARMSTADT

- Derived an asymmetry-aware model from facts and observable behavior
  - Storage properties
  - Algorithm access behavior
- Comparison shows improved performance on application-class benchmark
  - Average speed-up at 48%
  - Individual query speed-up by up to 549%
- Calibration is attracted by strong effects
  - TPC-H features properties not respected in PostgreSQL's optimizer
  - Available degrees of freedom may got abused to compensate deficiencies
- Additional experiments are required to decide upon original hypothesis
  - Using a special load (i.e. a custom micro-benchmark)
  - Focussing on optimization problems in which asymmetry matters explicitely
- Calibration may be a useful tool find optimal configurations in general
  - Typical queries need to be combined in a repeatable benchmark
  - Can cope with any number of variables

Bibliography

TECHNISCHE UNIVERSITÄT DARMSTADT

[Knut1973] Donald E. Knuth.
The Art of Computer Programming, Volume III: Sorting and Searching.
Addison-Wesley, 1973.

[Kirk1983] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi.
Optimization by Simulated Annealing.
In *Science*, vol. 220, no. 4598, pages 671–680, 1983.

[YoBC2009] Terry Yoshii, Christian Black, and Sudip Chahal.
Solid-state drives in the enterprise: A proof of concept.
Intel white paper, Intel Corporation, March 2009.