Introducing Skew into the TPC-H Benchmark

Alain Crolotte

Ahmad Ghazal

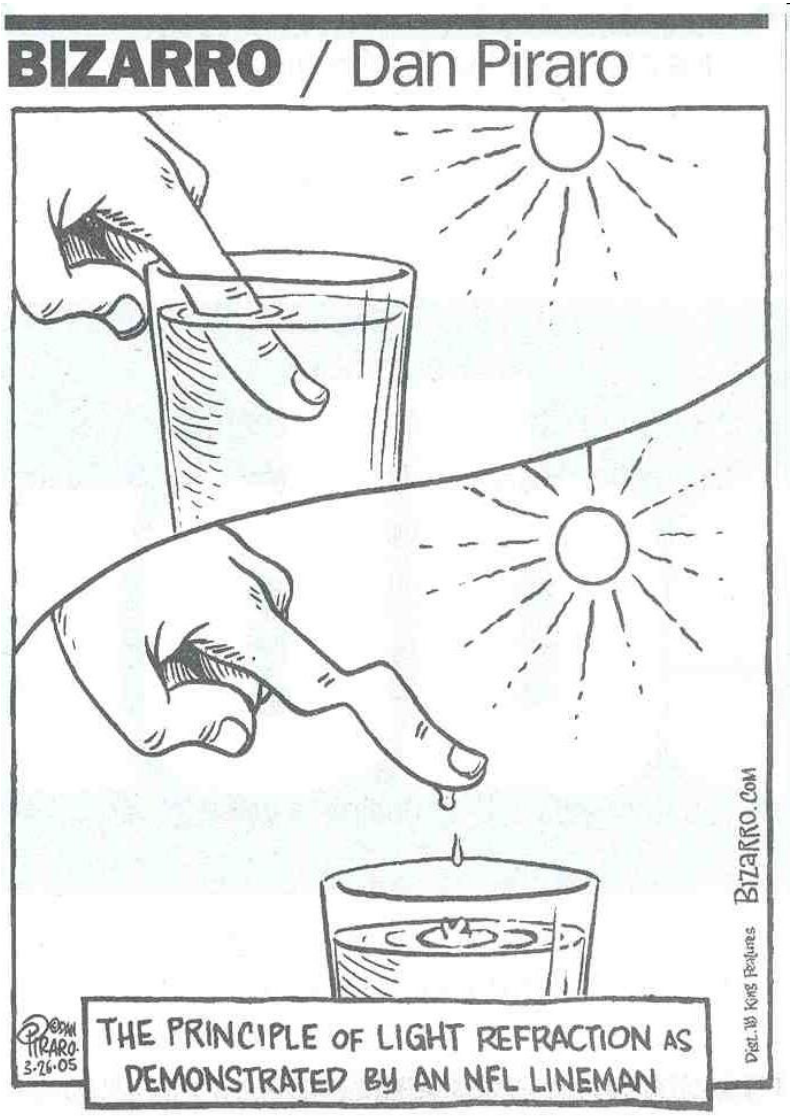
OUTLINE

- Problem Definition
- Potential Solutions
- Query Impact Analysis
- Final Proposal
- Conclusion

PROBLEM DEFINITION

- TPC-H is a well-known benchmark used by companies and universities for system sizing and research.
- Although skew is a know problem TPC-H distributions are uniform.
- How can we introduce skew in TPC-H without rewriting the entire benchmark?

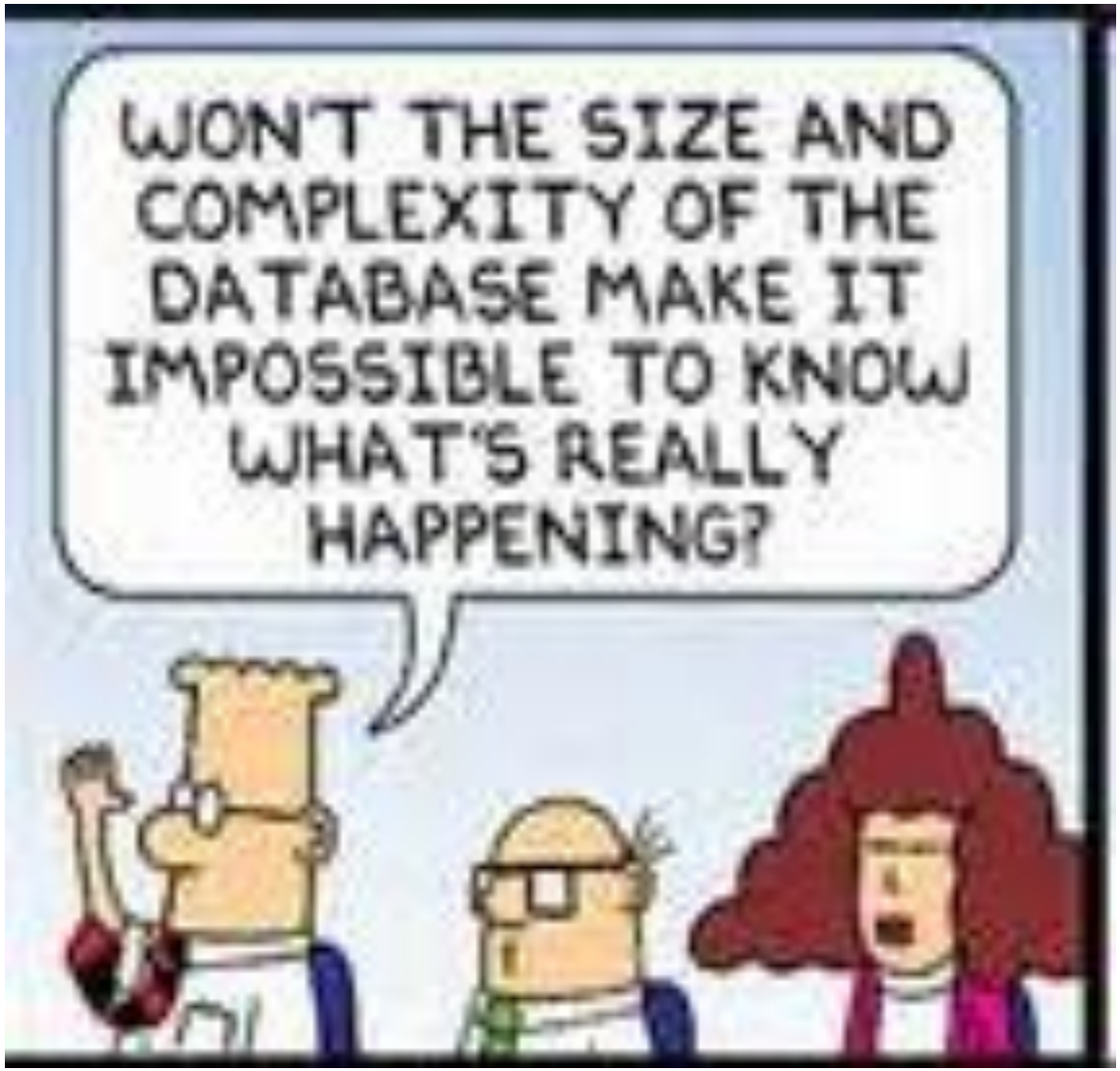
SKEW IN DBMS


BIZARRO / Dan Piraro
THE PRINCIPLE OF LIGHT REFRACTION AS DEMONSTRATED BY AN NFL LINEMAN
DAN PIRARO 3-26-05
DIST. BY KING FEATURES
BIZARRO.COM

POTENTIAL SOLUTIONS

- Changing some or all distributions according to known statistical laws
- Change a small number of tables that have an impact limited to a few queries: e.g. Nation

CHANGE ALL DISTRIBUTIONS



CHANGING NATION

- Currently 5 regions each containing 5 nations each with equal population
- Changing the nation populations will change the region populations
- Joins from nation to supplier and customer will be skewed
- Queries with WHERE clauses involving a specific nation or region will be affected
- Nation changes for supplier and customer must keep the total numbers constant

USING CENSUS DATA

nation-key	nation name	actual 2000 population	original #customers	#customers skewed	original #suppliers	#suppliers skewed
0	ALGERIA	32854000	5996505	1245313	399554	82867
1	ARGENTINA	38747000	5997462	1467640	399809	97991
2	BRAZIL	170000000	6001105	6434837	399867	429159
20	SAUDI ARABIA	24573000	5998452	929786	400246	62114
21	VIETNAM	70000000	6003717	2648881	400471	175911
22	RUSSIA	147000000	6000916	5566052	399995	370747
23	UNITED KINGDOM	58459000	6000497	2211755	399662	147395
24	UNITED STATES	281421906	5999998	10650785	399769	711709
	TOTAL	3962235727	150000000	150000000	10000000	10000000

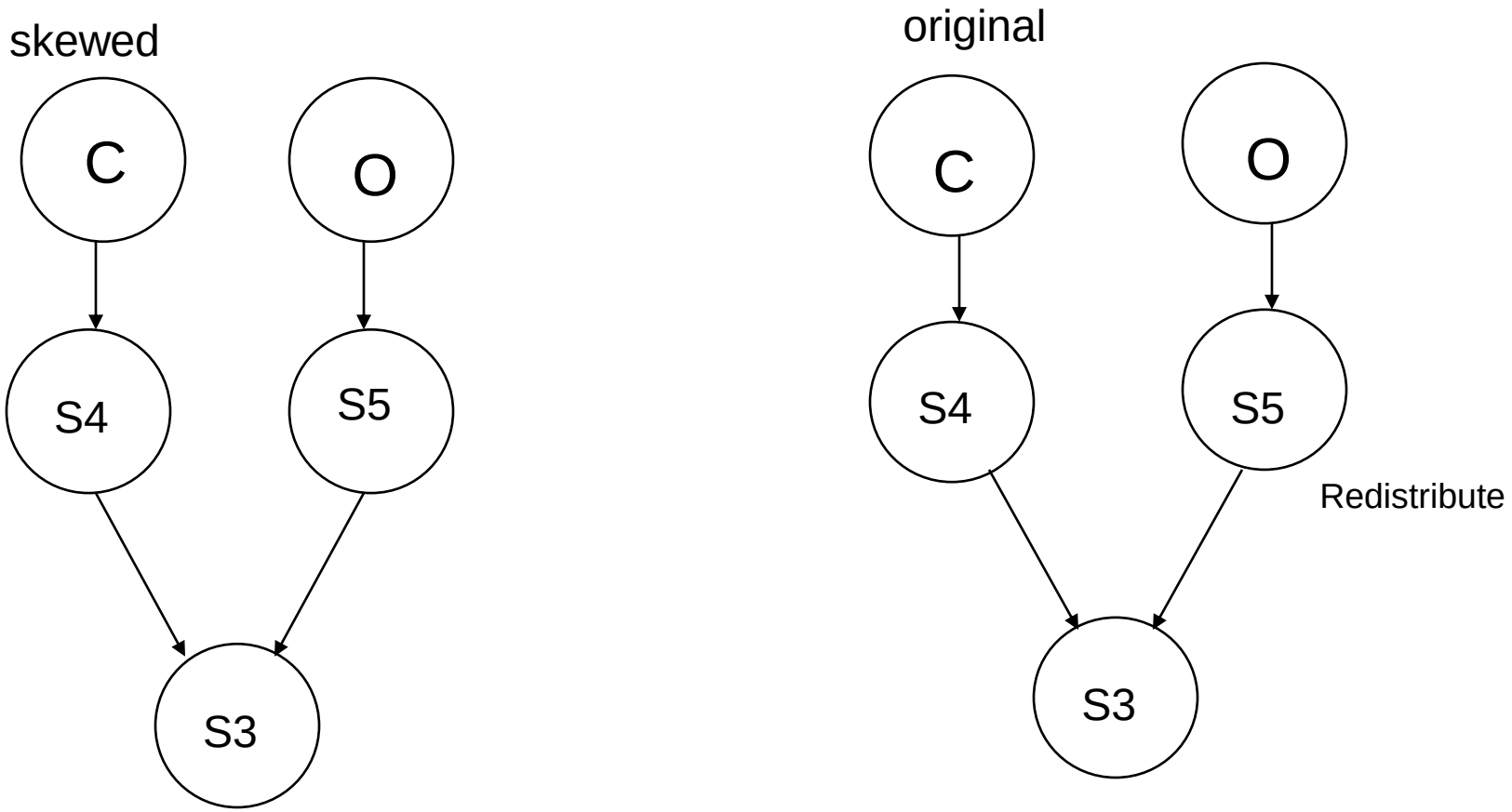
TPC-H QUERY RULES

- Multi-user test uses WHERE clause values that are random but create “equal” work
- For queries involving the supplier or customer table and a particular nation unequal populations create a problem
- Solution: equal populations in 2 separate groups: “large” nations and “small” nations

ALTERNATIVE 2

nation-key	nation name	original #customers	#customers skewed	original #suppliers	#suppliers skewed
0	ALGERIA	5996505	1127162	399554	75093
1	ARGENTINA	5997462	1129929	399809	75398
2	BRAZIL	6001105	1127874	399867	74755
11	IRAQ	6000601	1125231	399914	74940
12	JAPAN	5998850	1127436	399558	75545
13	JORDAN	6000889	11275061	399629	750316
14	KENYA	5997420	11280163	399302	752856
21	VIETNAM	6003717	11278040	400471	752970
22	RUSSIA	6000916	11275496	399995	753193
23	UNITED KINGDOM	6000497	11283291	399662	753027
24	UNITED STATES	5999998	11280056	399769	751789
	TOTAL	150000000	150000000	10000000	10000000

EFFECT OF SKEW - EXAMPLE




```
SELECT c_mktsegment , SUM(o_totalprice) FROM orders, customer
WHERE c_custkey = o_custkey
AND o_orderdate BETWEEN DATE '1997-12-31' -  INTERVAL '1' YEAR AND DATE '1997-12-31'
AND c_nationkey IN (<parm1>,<parm2>,<parm3>)
GROUP BY c_mktsegment;
```

IMPACT ANALYSIS

Queries Involving the Skewed Tables			
CUSTOMER		SUPPLIER	
via nation	via region	via nation	via region
7, 8, 20, 21	5, 8	11, 7	2, 5

A BETTER ALTERNATIVE

- Divide up nations into “small” and “large” each nation within a group with a constant size
- Each region has 3 “small” nations and 2 “large” nations
- Region sizes are now constant
- For each query impacted by nation (7, 8, 11, 20, 21) duplicate the queries into q.v1 and q.v2
- Version 1 uses a “small” nation while version 2 uses a “large” nation

CONCLUSION

- Overall impact on the benchmark spec relatively small
- Significant amount of dbgen and qgen work
- Proposed solution can be instrumented easily in the lab via insert-select and manual changes in queries