Deployment of Query Plans on Multicores

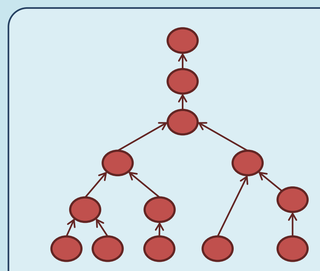
Jana Giceva, Gustavo Alonso, Timothy Roscoe
Department of Computer Science, ETH Zurich

Tim Harris
Oracle Labs

Systems @ ETH Zurich

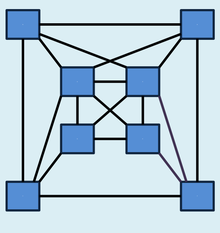
1. Problem Definition

Complex query plans
- Large number of operators
- Complex data-dependencies



Query plans

User requirements



Multicore machines

Manycore systems
- Complex architectures
- Asymmetric interconnect topology

Performance

Efficient resource utilization

User requirements for
- Good and stable performance
- Machine under-utilization avoidance

2. Approaches

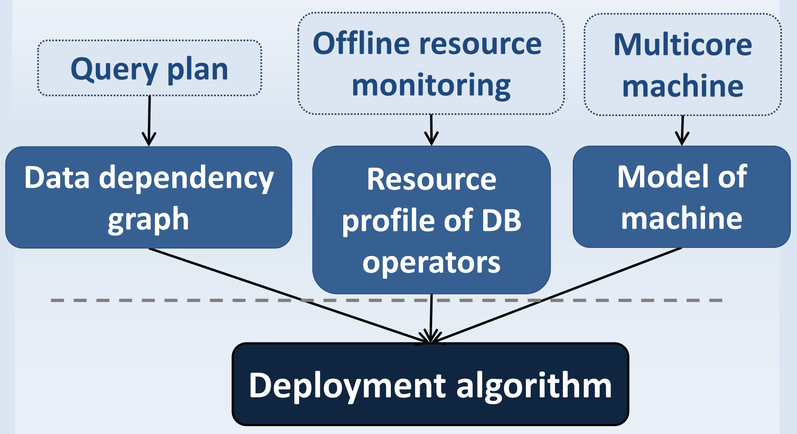
Until now:

1. Default operating system scheduling

2. Operator-per-core deployment

New:

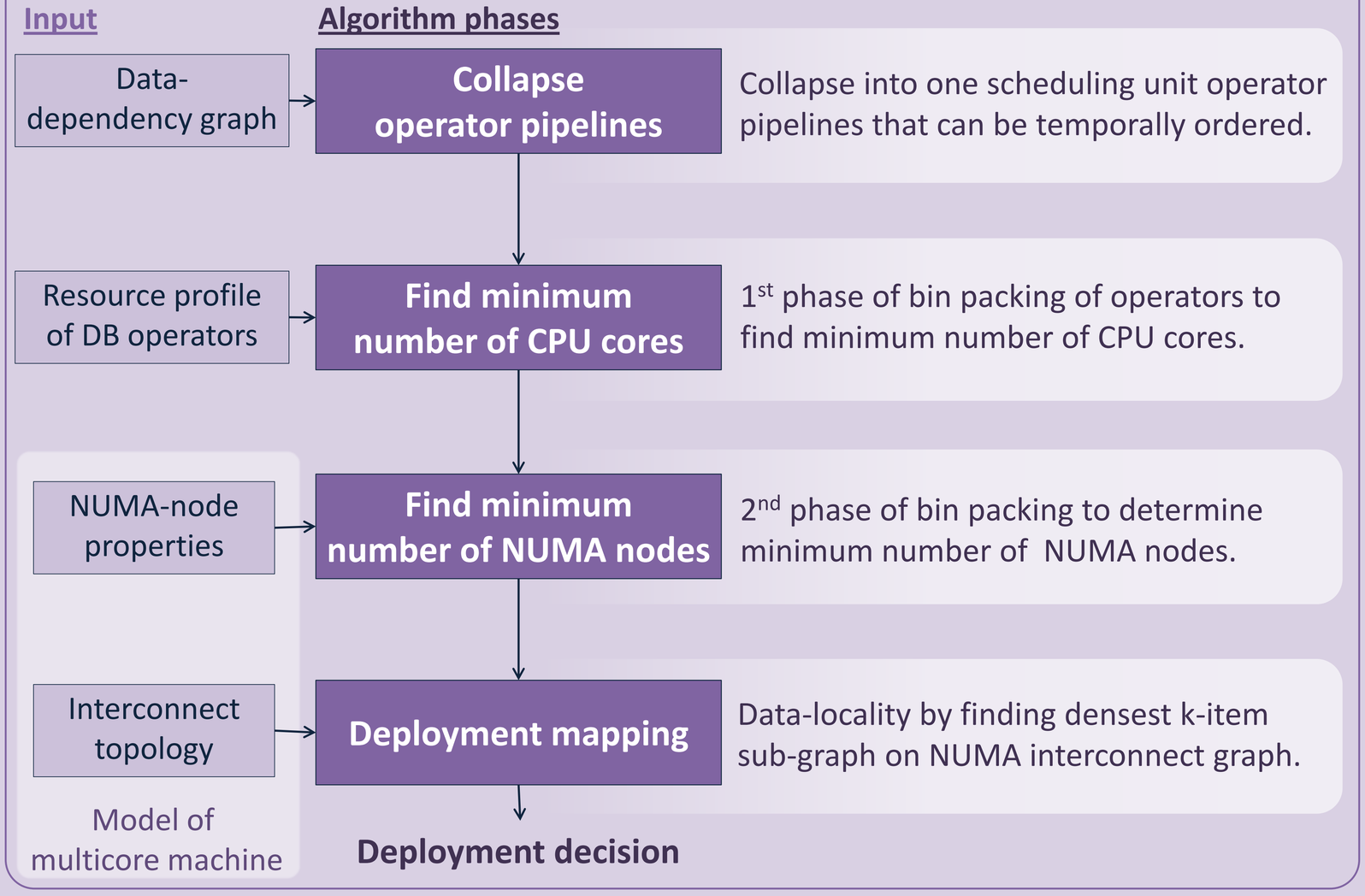
3. Our deployment algorithm



Minimize resource usage, provide NUMA-aware deployment & enhance data-locality.

GOAL: Deployment that minimizes resource footprint without affecting performance

3. Algorithm Phases



4. Evaluation

Setup:
- Query plan: SharedDB's TPC-W with 44 relational operators
- Dataset size: 20 GB
- Machine: AMD Magnycours

Results:

Approaches	# Cores	Throughput [WIPS] Average	Stdev	Latency percentiles [ms] 50th	90th	99th
Default OS	48	317	31	8	72	82
Operator-per-core	44	425	54	14	22	36
Deployment algorithm	6	428	32	15	23	36

Analysis:

1. Default Operating System scheduling
- ✗ Poorer performance
- ✗ Reduced stability and predictability
- ✗ Uses the whole machine

2. Operator-per-core deployment
- ✓ Good performance
- ✓ Stable and predictable
- ✗ Over-provisions the system

3. Our deployment algorithm
- ✓ Good performance
- ✓ Stable and predictable
- ✓ Minimizes resource requirement

Comparable performance with 1/7th of resources

Part of this work has been funded through a donation from Oracle Labs.
Jana Giceva is supported in part by European Google PhD Fellowship.

ETH zürich