The Volcano Optimizer Generator: Extensibility and Efficient Search

Presentation: Mirna Limic
Discussion: Jerry Zhang

What is Volcano? Why develop it?

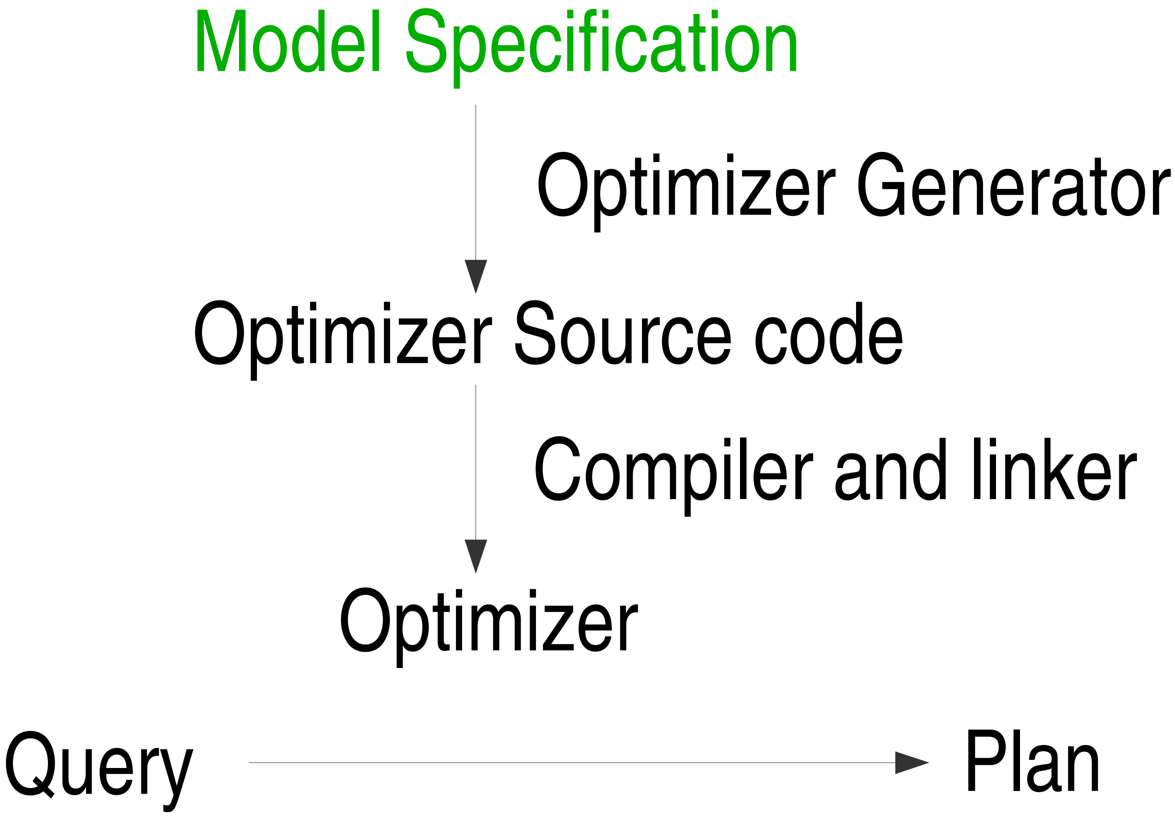
- An optimizer generator tuned towards object-oriented and scientific database systems. It can handle large data volumes.
- In short: Volcano is a package for “assembling” one's own efficient optimizer for the needs of an application.
- target = optimizer implementor

Definition: An architectural property of a program that allows its capabilities to expand is called *extensibility*. *(web.mit.edu/oki/learn/gloss.html)*

Where is extensibility in the Volcano Generator Model?

The Generator Paradigm

Where does extensibility come in the generator paradigm model?



Model Specification

Optimizer Generator

Optimizer Source code

Compiler and linker

Optimizer

Query

Plan

Design Principles of Opt. Gen.

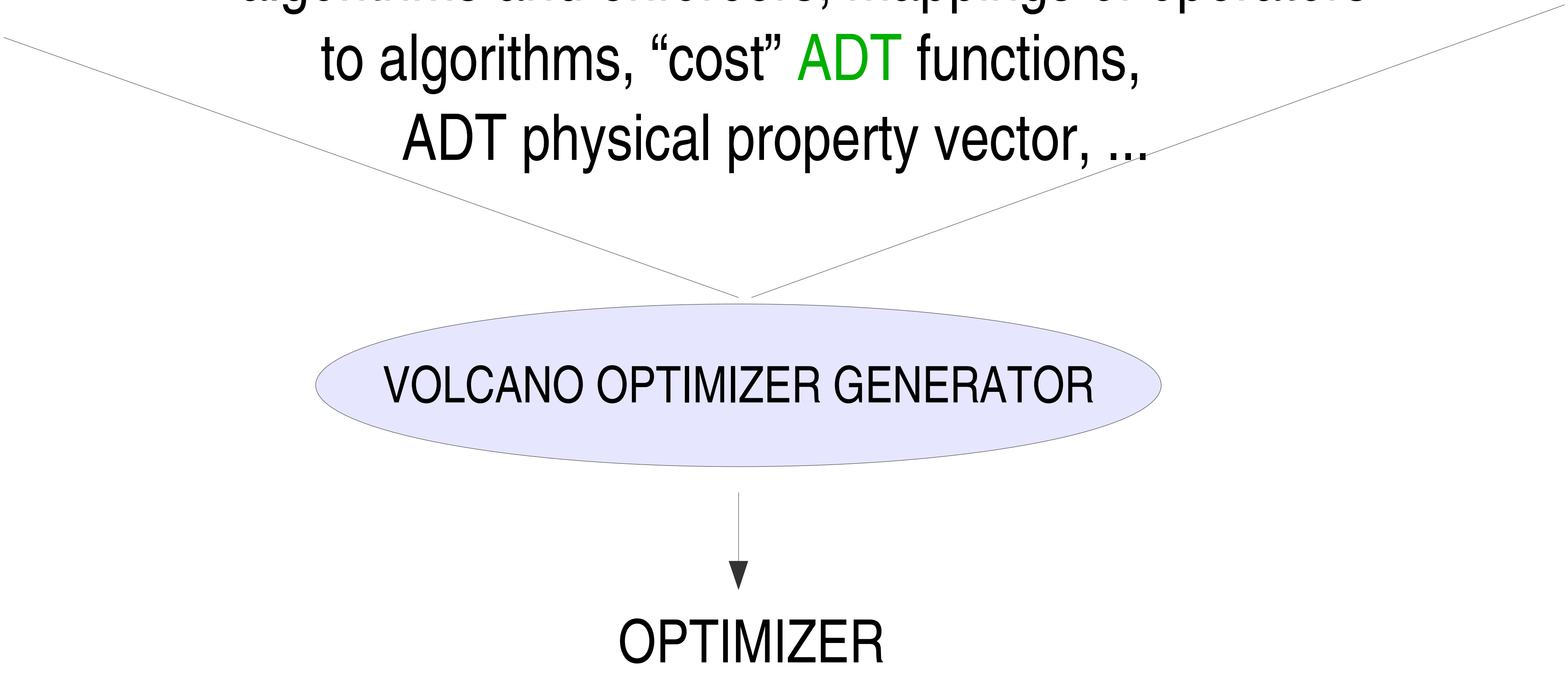
relational algebra – logical for the set of algebraic operators, physical for the set of algorithms

rules over relational algebra – used in equivalence transformations, allow for modularity

rule compilation rather than interpretation

dynamic programming used to find the most efficient plan

How do we make use of extensibility?



By specifying the model using:

logical operators, algebraic transformation rules,

algorithms and enforcers, mappings of operators

to algorithms, “cost” ADT functions,

ADT physical property vector, ...

VOLCANO OPTIMIZER GENERATOR

OPTIMIZER

ADT is Abstract Data Type

The Search Engine - Terms

Results of an algebra expression are described using properties

- logical – e.g. types in schema, a type's expected size
- physical – e.g. sort order, uniqueness

An enforcer is a physical algebra operator that ensures one or two physical properties. It does not correspond to any operator in the logical algebra.

Search Engine – Terms, cont'd

plan is used for two things so we differentiate:

execution plan: a set of algorithms,
their inputs,
the ordering in which algorithms are executed
the total cost of executing the algorithms

decision plan: a way (plan) of doing something

So How Does It Work?

Hash table containing expressions and equivalence classes.

A row contains the following

a logical expression	equivalence classes	
	equivalent logical expressions	physical expressions i.e. execution plans

Decision plan:

Based on the input: query exp., phys. prop., cost limit do in order

-Read the execution plan by keying on the logical expression, where the execution plan matches the physical properties supplied, and its cost is less than the cost limit

-If no such execution plan exists do one of the following:

- use an equivalent logical expression
- use an algorithm
- use enforcer to change the physical properties

Volcano vs. Starburst

- Volcano can do general algebraic queries, Starburst can do only SPJ queries
- Starburst optimizer evaluates all alternative query evaluation plans (QEP's) to find the cheapest, Volcano evaluates only those it needs to.
- Starburst has a hierarchy of intermediate levels, Volcano uses an algebraic approach (Which one is easier to understand?)

Goals of the new optimizer generator

- make the new generator compatible with the existing query execution software, and as stand-alone tool
- make it more efficient than its predecessor (EXODUS) in optimization time and memory requirements
- allow for the definition of one's own physical properties (e.g. sort order)
- use heuristics and specification of the data model when searching for the optimal plan of execution of a query
- generate optimization plans for incompletely specified queries