syntax = "proto3";

package tensorflow.data.model;

option cc\_enable\_arenas = true;

option go\_package = "github.com/tensorflow/tensorflow/tensorflow/go/core/framework/model\_go\_proto";

// Class of a node in the performance model.

enum NodeClass {

UNKNOWN = 0;

INTERLEAVE\_MANY = 1;

ASYNC\_INTERLEAVE\_MANY = 2;

KNOWN\_RATIO = 3;

ASYNC\_KNOWN\_RATIO = 4;

UNKNOWN\_RATIO = 5;

}

// Algorithm used for model autotuning optimization.

enum AutotuneAlgorithm {

DEFAULT = 0;

HILL\_CLIMB = 1;

GRADIENT\_DESCENT = 2;

MAX\_PARALLELISM = 3;

}

// Protocol buffer representing the data used by the autotuning modeling

// framework.

message ModelProto {

// General representation of a node in the model.

message Node {

// Unique node ID.

int64 id = 1;

// Human-readable name of the node.

string name = 2;

// An indication whether autotuning is enabled for this node.

bool autotune = 3;

// The number of bytes stored in this node's buffer.

int64 buffered\_bytes = 4;

// The number of elements stored in this node's buffer.

int64 buffered\_elements = 5;

// The number of bytes consumed by the node.

int64 bytes\_consumed = 6;

// The number of bytes produced by the node.

int64 bytes\_produced = 7;

// The number of elements produced by the node.

int64 num\_elements = 8;

// The aggregate processing time spent in this node.

int64 processing\_time = 9;

// An indication whether this node records metrics about produced and

// consumed elements.

bool record\_metrics = 10;

// Represents a node parameter.

message Parameter {

// Human-readable name of the parameter.

string name = 1;

// Identifies the model value of the parameter. This can be different from

// the actual value (e.g. during optimization search).

double value = 2;

// The actual value of the parameter.

double state\_value = 3;

// Minimum value of the parameter.

double min = 4;

// Maximum value of the parameter.

double max = 5;

// Identifies whether the parameter should participate in autotuning.

bool tunable = 6;

}

// Parameters of this node.

repeated Parameter parameters = 11;

// Statistic of inputs processing time history.

double input\_processing\_time\_sum = 12;

int64 input\_processing\_time\_count = 13;

// IDs of inputs of this node.

repeated int64 inputs = 14;

// Class of this node.

NodeClass node\_class = 15;

// Ratio of input to output elements. This is only used by KNOWN\_RATIO and

// ASYNC\_KNOWN\_RATIO nodes.

double ratio = 16;

// Ratio identifies how many parallelism calls are introduced by one

// buffered element. This is only used by ASYNC\_KNOWN\_RATIO nodes.

double memory\_ratio = 17;

}

// Map of node IDs to nodes of this model.

map<int64, Node> nodes = 1;

// ID of the output node of this model.

int64 output = 2;

// Counter for node IDs of this model.

int64 id\_counter = 3;

reserved 4;

// Contains parameters of the model autotuning optimization.

message OptimizationParams {

// Algorithm used for autotuning optimization.

AutotuneAlgorithm algorithm = 1;

// Number of available logical threads.

int64 cpu\_budget = 2;

// Amount of available memory in bytes.

int64 ram\_budget = 3;

// Time between two consecutive `GetNext` calls to the iterator represented

// by the output node.

double model\_input\_time = 4;

}

OptimizationParams optimization\_params = 5;

}