syntax = "proto3";

package tensorflow;

import "tensorflow/core/framework/tensor\_shape.proto";

import "tensorflow/core/framework/types.proto";

option cc\_enable\_arenas = true;

option java\_outer\_classname = "CostGraphProtos";

option java\_multiple\_files = true;

option java\_package = "org.tensorflow.framework";

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

message CostGraphDef {

message Node {

// The name of the node. Names are globally unique.

string name = 1;

// The device of the node. Can be empty if the node is mapped to the

// default partition or partitioning hasn't been run yet.

string device = 2;

// The id of the node. Node ids are only unique inside a partition.

int32 id = 3;

// Inputs of this node. They must be executed before this node can be

// executed. An input is a particular output of another node, specified

// by the node id and the output index.

message InputInfo {

int32 preceding\_node = 1;

int32 preceding\_port = 2;

}

repeated InputInfo input\_info = 4;

// Outputs of this node.

message OutputInfo {

int64 size = 1;

// If >= 0, the output is an alias of an input. Note that an alias input

// may itself be an alias. The algorithm will therefore need to follow

// those pointers.

int64 alias\_input\_port = 2;

TensorShapeProto shape = 3;

DataType dtype = 4;

}

repeated OutputInfo output\_info = 5;

// Temporary memory used by this node.

int64 temporary\_memory\_size = 6;

// Persistent memory used by this node.

int64 persistent\_memory\_size = 12;

int64 host\_temp\_memory\_size = 10 [deprecated = true];

int64 device\_temp\_memory\_size = 11 [deprecated = true];

int64 device\_persistent\_memory\_size = 16 [deprecated = true];

// Estimate of the computational cost of this node, in microseconds.

int64 compute\_cost = 9;

// Analytical estimate of the computational cost of this node, in

// microseconds.

int64 compute\_time = 14;

// Analytical estimate of the memory access cost of this node, in

// microseconds.

int64 memory\_time = 15;

// If true, the output is permanent: it can't be discarded, because this

// node is part of the "final output". Nodes may depend on final nodes.

bool is\_final = 7;

// Ids of the control inputs for this node.

repeated int32 control\_input = 8;

// Are the costs inaccurate?

bool inaccurate = 17;

}

repeated Node node = 1;

// Total cost of this graph, typically used for balancing decisions.

message AggregatedCost {

// Aggregated cost value.

float cost = 1;

// Aggregated cost dimension (e.g. 'memory', 'compute', 'network').

string dimension = 2;

}

repeated AggregatedCost cost = 2;

}