syntax = "proto3";

package tensorflow;

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

option cc\_enable\_arenas = true;

option java\_outer\_classname = "GraphTransferInfoProto";

option java\_multiple\_files = true;

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

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

message GraphTransferNodeInput {

int32 node\_id = 1;

int32 output\_port = 2;

}

message GraphTransferNodeInfo {

string name = 1;

int32 node\_id = 2;

string type\_name = 3;

int32 soc\_op\_id = 4;

int32 padding\_id = 5;

int32 input\_count = 6;

int32 output\_count = 7;

}

message GraphTransferConstNodeInfo {

string name = 1;

int32 node\_id = 2;

repeated int64 shape = 3;

bytes data = 4;

DataType dtype = 5;

}

message GraphTransferNodeInputInfo {

int32 node\_id = 1;

repeated GraphTransferNodeInput node\_input = 2;

}

message GraphTransferNodeOutputInfo {

int32 node\_id = 1;

repeated int32 max\_byte\_size = 2;

}

message GraphTransferGraphInputNodeInfo {

string name = 1;

repeated int64 shape = 2;

DataType dtype = 3;

}

message GraphTransferGraphOutputNodeInfo {

string name = 1;

repeated int64 shape = 2;

DataType dtype = 3;

}

// Protocol buffer representing a handle to a tensorflow resource. Handles are

// not valid across executions, but can be serialized back and forth from within

// a single run.

message GraphTransferInfo {

enum Destination {

NOP = 0;

HEXAGON = 1;

}

repeated GraphTransferNodeInfo node\_info = 1;

repeated GraphTransferConstNodeInfo const\_node\_info = 2;

repeated GraphTransferNodeInputInfo node\_input\_info = 3;

repeated GraphTransferNodeOutputInfo node\_output\_info = 4;

// Input Node parameters of transferred graph

repeated GraphTransferGraphInputNodeInfo graph\_input\_node\_info = 5;

repeated GraphTransferGraphOutputNodeInfo graph\_output\_node\_info = 6;

// Destination of graph transfer

Destination destination = 7;

}