syntax = "proto3";

package tensorflow;

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

option cc\_enable\_arenas = true;

option java\_outer\_classname = "LogMemoryProtos";

option java\_multiple\_files = true;

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

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

message MemoryLogStep {

// Process-unique step id.

int64 step\_id = 1;

// Handle describing the feeds and fetches of the step.

string handle = 2;

}

message MemoryLogTensorAllocation {

// Process-unique step id.

int64 step\_id = 1;

// Name of the kernel making the allocation as set in GraphDef,

// e.g., "affine2/weights/Assign".

string kernel\_name = 2;

// Allocated tensor details.

TensorDescription tensor = 3;

}

message MemoryLogTensorDeallocation {

// Id of the tensor buffer being deallocated, used to match to a

// corresponding allocation.

int64 allocation\_id = 1;

// Name of the allocator used.

string allocator\_name = 2;

}

message MemoryLogTensorOutput {

// Process-unique step id.

int64 step\_id = 1;

// Name of the kernel producing an output as set in GraphDef, e.g.,

// "affine2/weights/Assign".

string kernel\_name = 2;

// Index of the output being set.

int32 index = 3;

// Output tensor details.

TensorDescription tensor = 4;

}

message MemoryLogRawAllocation {

// Process-unique step id.

int64 step\_id = 1;

// Name of the operation making the allocation.

string operation = 2;

// Number of bytes in the allocation.

int64 num\_bytes = 3;

// Address of the allocation.

uint64 ptr = 4;

// Id of the tensor buffer being allocated, used to match to a

// corresponding deallocation.

int64 allocation\_id = 5;

// Name of the allocator used.

string allocator\_name = 6;

}

message MemoryLogRawDeallocation {

// Process-unique step id.

int64 step\_id = 1;

// Name of the operation making the deallocation.

string operation = 2;

// Id of the tensor buffer being deallocated, used to match to a

// corresponding allocation.

int64 allocation\_id = 3;

// Name of the allocator used.

string allocator\_name = 4;

// True if the deallocation is queued and will be performed later,

// e.g. for GPU lazy freeing of buffers.

bool deferred = 5;

}