syntax = "proto3";

package tensorflow.data.experimental;

import "tensorflow/core/protobuf/data\_service.proto";

option go\_package = "github.com/tensorflow/tensorflow/tensorflow/go/core/protobuf/for\_core\_protos\_go\_proto";

// Configuration for a tf.data service DispatchServer.

// Next id: 10

message DispatcherConfig {

// The port for the dispatcher to bind to. A value of 0 indicates that the

// dispatcher may bind to any available port.

int64 port = 1;

// The protocol for the dispatcher to use when connecting to workers.

string protocol = 2;

// A work directory to use for storing dispatcher state, and for recovering

// during restarts. The empty string indicates not to use any work directory.

string work\_dir = 3;

// Whether to run in fault tolerant mode, where dispatcher state is saved

// across restarts. Requires that `work\_dir` is nonempty.

bool fault\_tolerant\_mode = 4;

// (Optional.) If the job uses auto-sharding, it needs to specify a fixed list

// of worker addresses that will register with the dispatcher. The worker

// addresses should be in the format "host" or "host:port", where "port" is an

// integer, named port, or %port% to match any port.

repeated string worker\_addresses = 7;

// (Optional.) tf.data service deployment mode. Supported values are "REMOTE",

// "COLOCATED", and "HYBRID". If unspecified, it is assumed to be "REMOTE".

DeploymentMode deployment\_mode = 9;

// How often the dispatcher should scan through to delete old and unused

// jobs. A value of 0 indicates that the decision should be left up to the

// runtime.

int64 job\_gc\_check\_interval\_ms = 5;

// How long a job needs to be unused before it becomes a candidate for garbage

// collection. A value of -1 indicates that jobs should never be garbage

// collected. A value of 0 indicates that the decision should be left up to

// the runtime.

int64 job\_gc\_timeout\_ms = 6;

// How long to wait before garbage-collecting a client that hasn't

// heartbeated to the dispatcher. A value of 0 indicates that the timeout

// should be left to the runtime.

int64 client\_timeout\_ms = 8;

}

// Configuration for a tf.data service WorkerServer.

// Next id: 11

message WorkerConfig {

// The port for the worker to bind to. A value of 0 indicates that the

// worker may bind to any available port.

int64 port = 1;

// The protocol for the worker to use when connecting to the dispatcher.

string protocol = 2;

// The address of the dispatcher to register with.

string dispatcher\_address = 3;

// The address of the worker server. The substring "%port%", if specified,

// will be replaced with the worker's bound port. This is useful when the port

// is set to `0`.

string worker\_address = 4;

// Tags attached to the worker. This allows reading from selected workers.

// For example, by applying a "COLOCATED" tag, tf.data service is able to read

// from the local tf.data worker if one exists, then from off-TF-host workers,

// to avoid cross-TF-host reads.

repeated string worker\_tags = 10;

// How often the worker should heartbeat to the master. A value of 0 indicates

// that the decision should be left up to the runtime.

int64 heartbeat\_interval\_ms = 5;

// How long to retry requests to the dispatcher before giving up and reporting

// an error. A value of 0 indicates that the decision should be left up to the

// runtime.

int64 dispatcher\_timeout\_ms = 6;

// The protocol for the worker to use when transferring data to clients.

string data\_transfer\_protocol = 7;

// The data transfer address of the worker server. The substring "%port%", if

// specified, will be replaced with the worker's bound port. This is useful

// when the port is set to `0`.

string data\_transfer\_address = 8;

// When shutting down a worker, how long to wait for the gRPC server to

// process the final requests. This is used to achieve clean shutdown in unit

// tests.

int64 shutdown\_quiet\_period\_ms = 9;

}