#include "tensorflow/core/framework/op.h"

#include "tensorflow/core/framework/shape\_inference.h"

#include "tensorflow/core/framework/op\_kernel.h"

#include "tensorflow/core/framework/common\_shape\_fns.h"

using namespace tensorflow;

REGISTER\_OP("NumIntraOpThreads")

.Input("x: float32")

.Output("num\_intra\_op\_threads: int32")

.SetShapeFn(tensorflow::shape\_inference::ScalarShape)

.Doc(R"doc(

A tensorflow OP that returns the number of threads in the intra\_op\_parallelism pool

This is not part of the Tensorflow API as of the date of writing this doc. Hence,

a tensorflow operation is the best resort.

Input

x: Dummy placeholder so that constant folding is not done by TF GraphOptimizer.

Please refer https://github.com/tensorflow/tensorflow/issues/22546 for more

details.

Output

num\_intra\_op\_threads: A scalar tensor corresponding to the number of threads in

the intra\_op\_parallelism pool

)doc");

class NumIntraOpThreads : public OpKernel {

public:

explicit NumIntraOpThreads(OpKernelConstruction\* context)

: OpKernel(context) {}

void Compute(OpKernelContext\* context) override {

int num\_intra\_op\_threads = context->device()->tensorflow\_cpu\_worker\_threads()->num\_threads;

Tensor\* output\_tensor = NULL;

OP\_REQUIRES\_OK(context, context->allocate\_output(0, TensorShape({}), &output\_tensor));

auto output\_flat = output\_tensor->flat<int32>();

output\_flat(0) = num\_intra\_op\_threads;

}

};

REGISTER\_KERNEL\_BUILDER(Name("NumIntraOpThreads").Device(DEVICE\_CPU), NumIntraOpThreads);