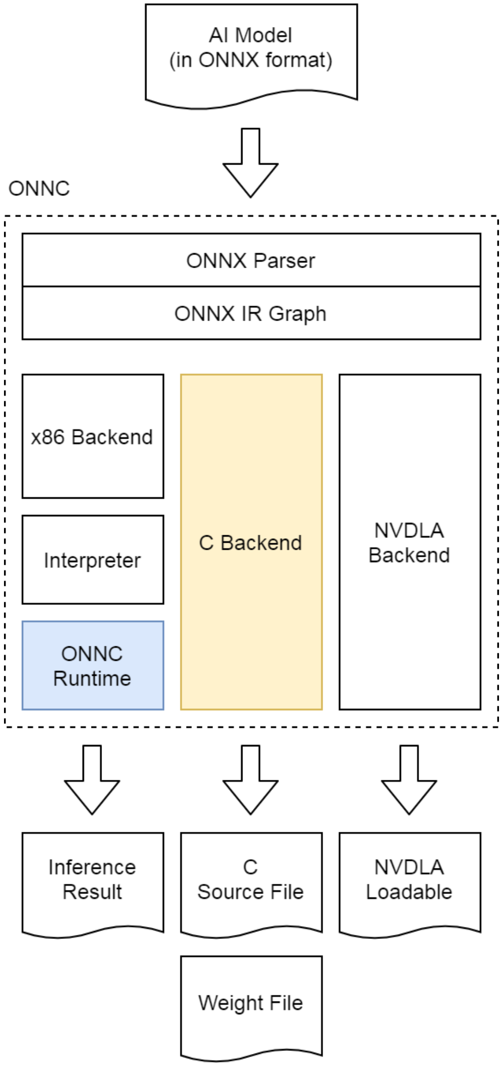
Skymizer | Introduction of ONNC C Backend

Po-Yen Chen (poyenc@skymizer.com)

a

ONNC Framework

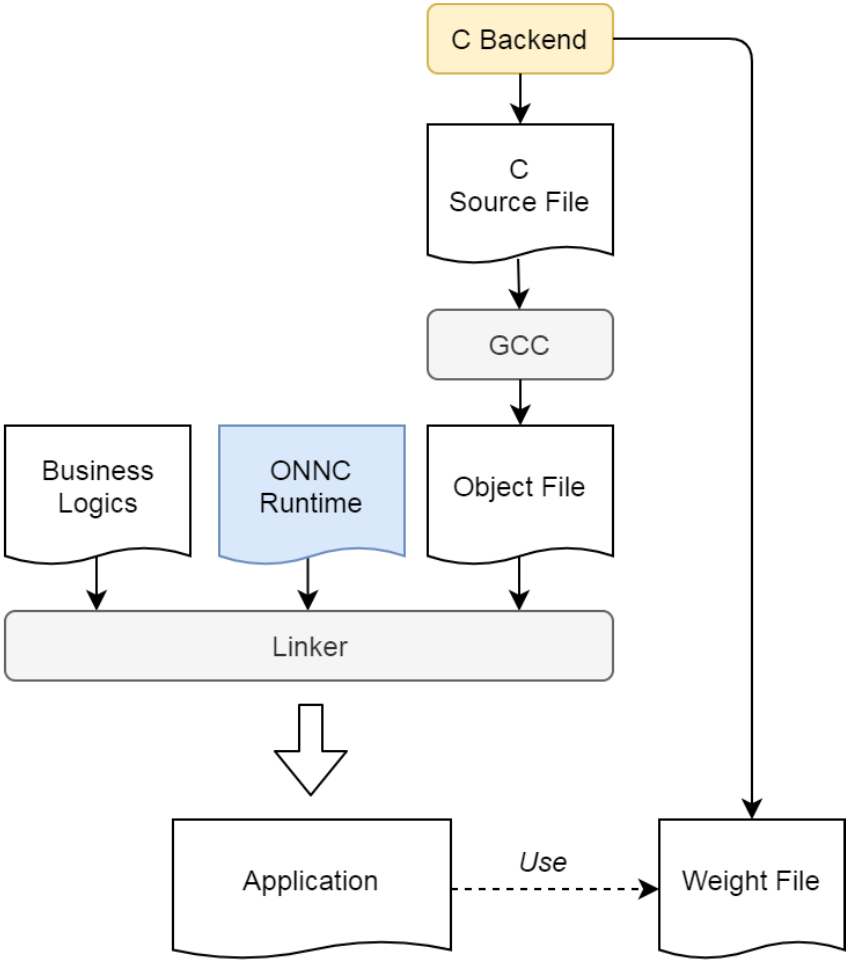
- ONNC Runtime is a model-inference library
- C Backend transforms a model into a C source file and a weight file
- Users can run model inference by using several of C functions



a

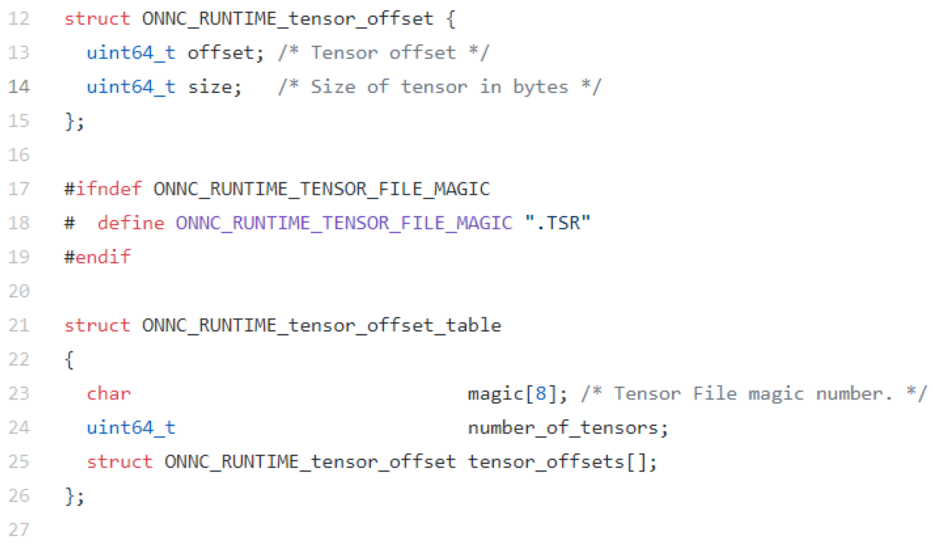
C Backend Workflow

- C Backend creates implementation of a single function: *model_main()*
- *model_main()* is the entry function for input model inference flow
- *model_main()* employ ONNC Runtime to compute results

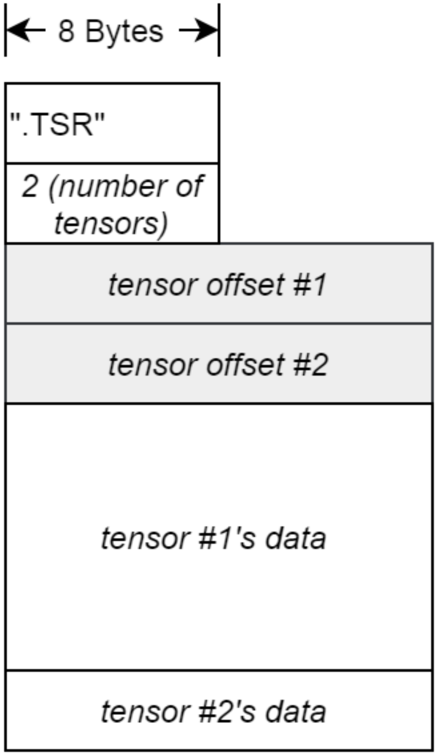


Weight file format

include/onnc/Runtime/onnc-runtime.h:12



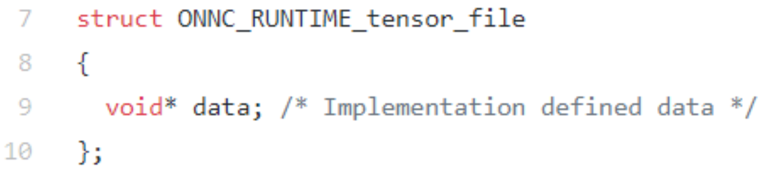
```
struct ONNC_RUNTIME_tensor_offset {
  uint64_t offset; /* Tensor offset */
  uint64_t size;   /* Size of tensor in bytes */
};

#ifndef ONNC_RUNTIME_TENSOR_FILE_MAGIC
#  define ONNC_RUNTIME_TENSOR_FILE_MAGIC ".TSR"
#endif

struct ONNC_RUNTIME_tensor_offset_table
{
  char                               magic[8]; /* Tensor File magic number. */
  uint64_t                           number_of_tensors;
  struct ONNC_RUNTIME_tensor_offset tensor_offsets[];
};

```



a

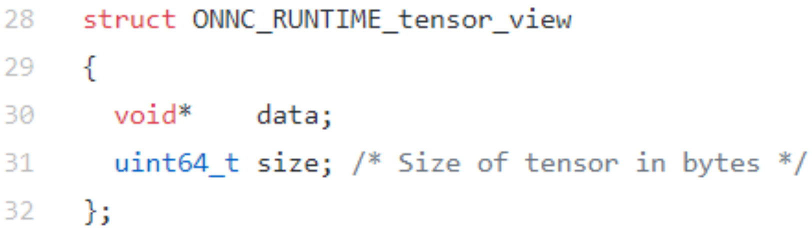
Tensor data format

include/onnc/Runtime/onnc-runtime.h:7

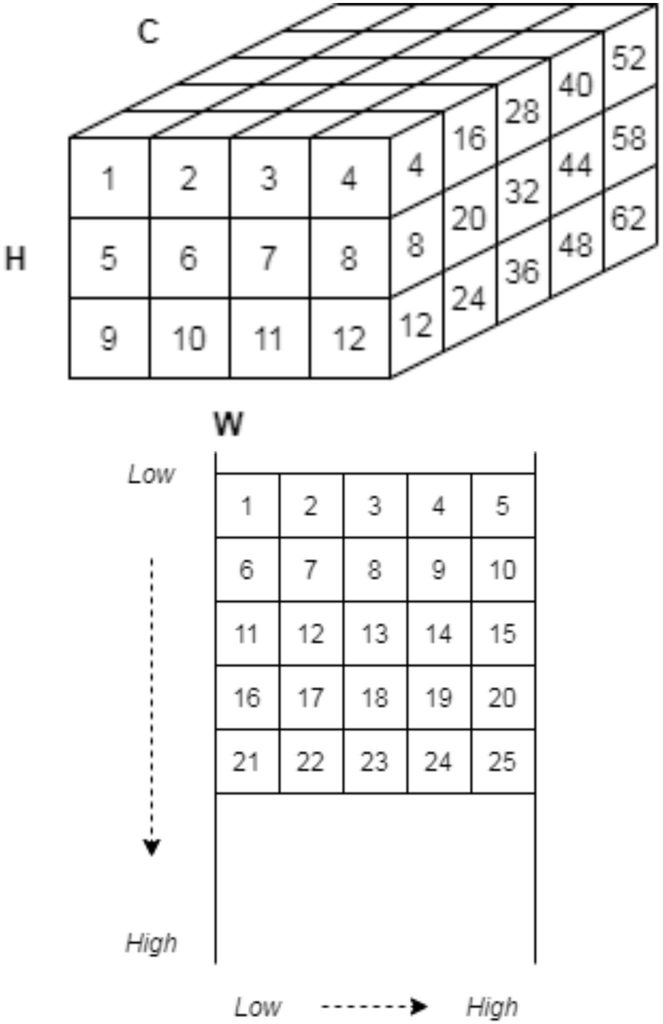


```
struct ONNC_RUNTIME_tensor_file
{
  void* data; /* Implementation defined data */
};
```

include/onnc/Runtime/onnc-runtime.h:28



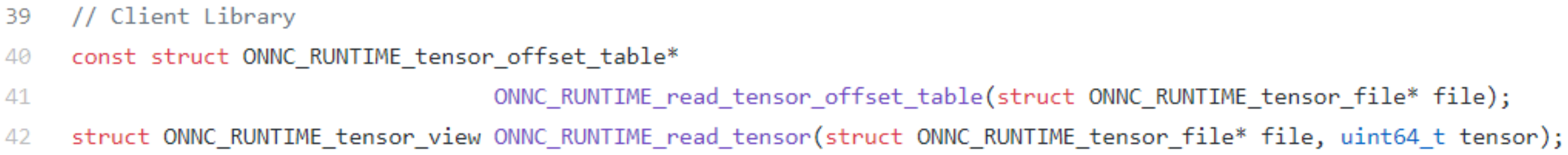
```
struct ONNC_RUNTIME_tensor_view
{
  void*     data;
  uint64_t size; /* Size of tensor in bytes */
};
```



a

Tensor access API (user defined)

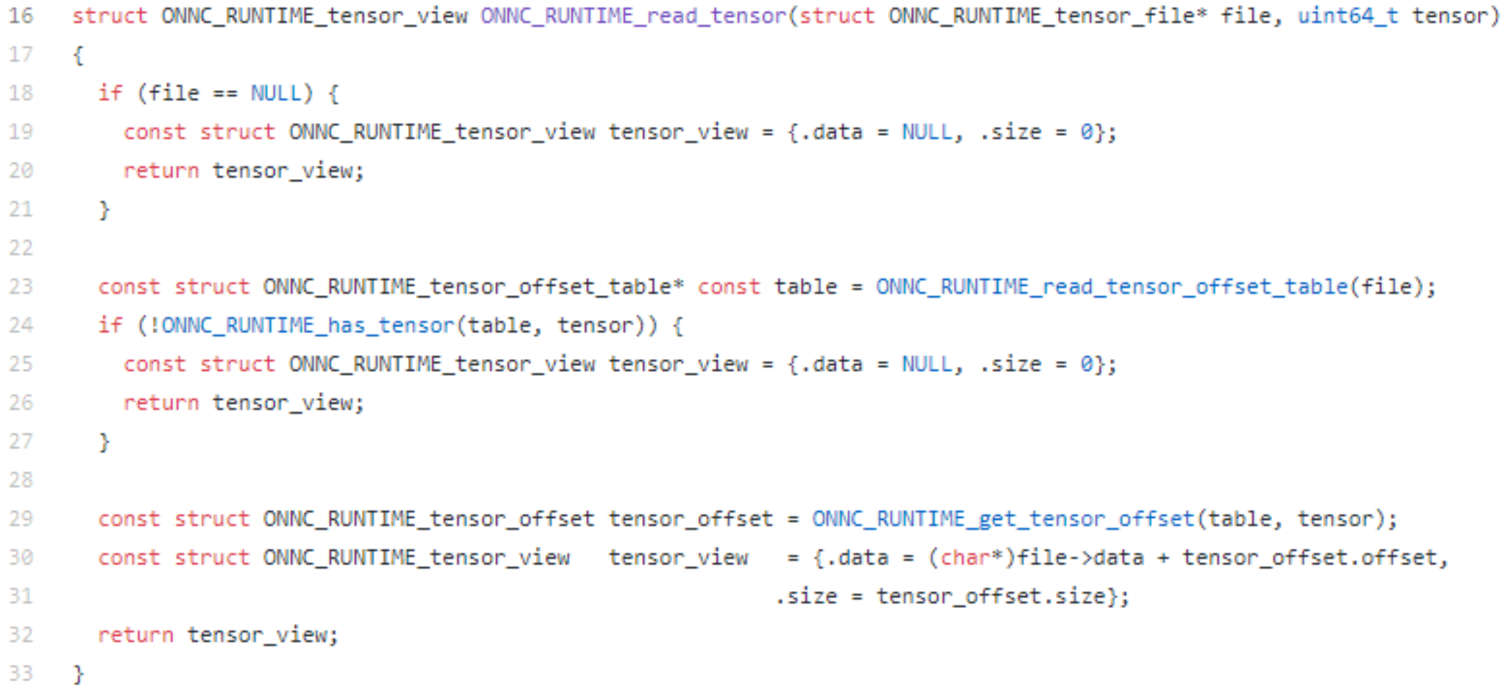
include/onnc/Runtime/onnc-runtime.h:39



```
// Client Library
const struct ONNC_RUNTIME_tensor_offset_table*
                                      ONNC_RUNTIME_read_tensor_offset_table(struct ONNC_RUNTIME_tensor_file* file);
struct ONNC_RUNTIME_tensor_view ONNC_RUNTIME_read_tensor(struct ONNC_RUNTIME_tensor_file* file, uint64_t tensor);
```

A possible implementation

example/runtime/src/client-lib.c

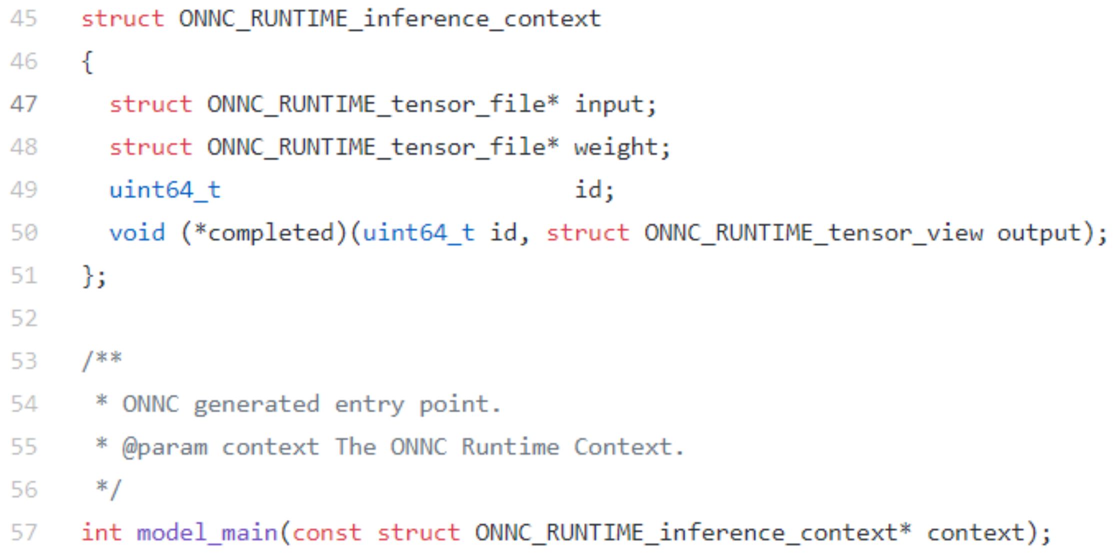


```
struct ONNC_RUNTIME_tensor_view ONNC_RUNTIME_read_tensor(struct ONNC_RUNTIME_tensor_file* file, uint64_t tensor)
{
  if (file == NULL) {
    const struct ONNC_RUNTIME_tensor_view tensor_view = {.data = NULL, .size = 0};
    return tensor_view;
  }

  const struct ONNC_RUNTIME_tensor_offset_table* const table = ONNC_RUNTIME_read_tensor_offset_table(file);
  if (!ONNC_RUNTIME_has_tensor(table, tensor)) {
    const struct ONNC_RUNTIME_tensor_view tensor_view = {.data = NULL, .size = 0};
    return tensor_view;
  }

  const struct ONNC_RUNTIME_tensor_offset tensor_offset = ONNC_RUNTIME_get_tensor_offset(table, tensor);
  const struct ONNC_RUNTIME_tensor_view   tensor_view   = {.data = (char*)file->data + tensor_offset.offset,
                                                           .size = tensor_offset.size};
  return tensor_view;
}
```

a

Context and model_main()

include/onnc/Runtime/onnc-runtime.h:45

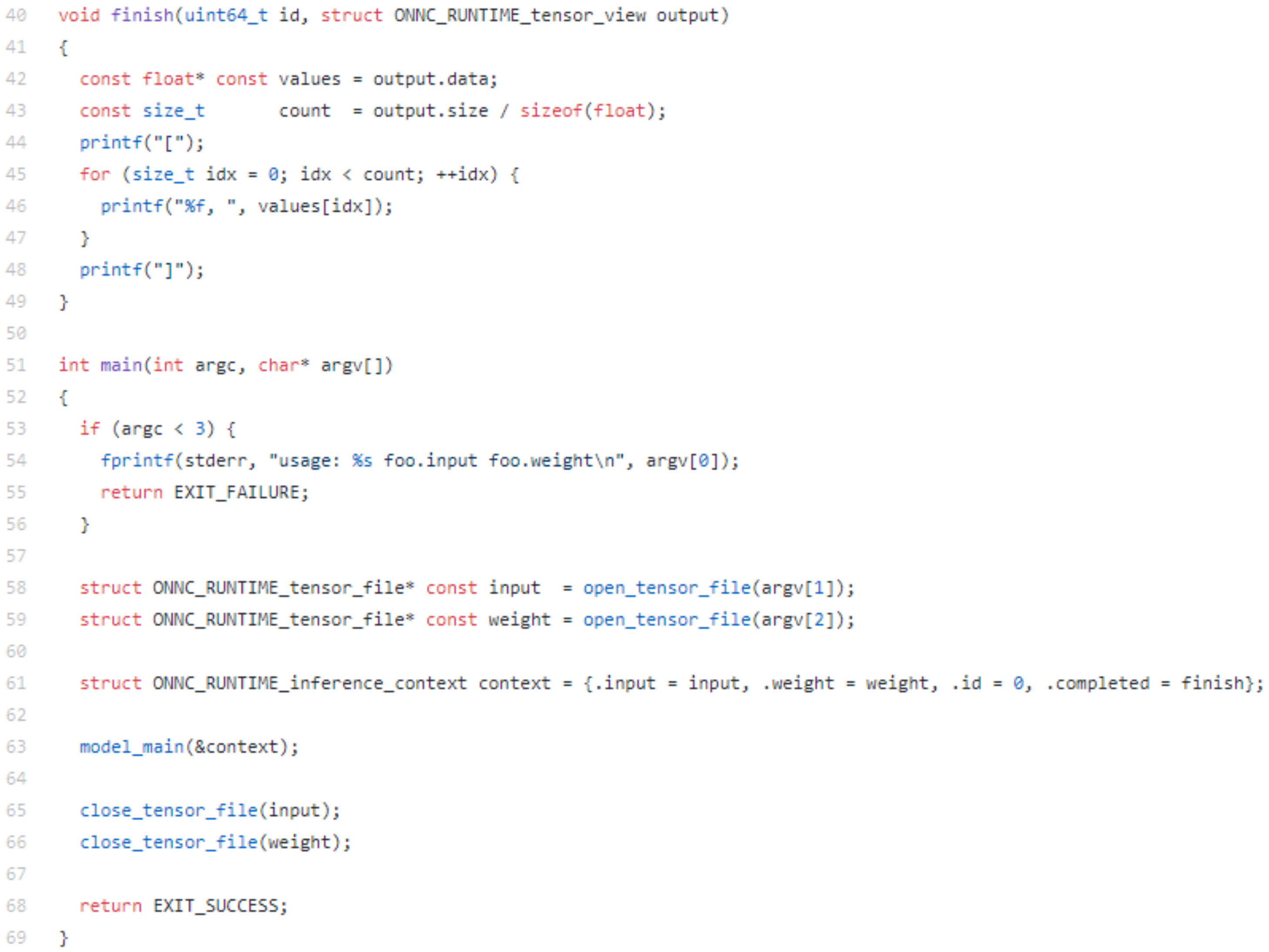


```
struct ONNC_RUNTIME_inference_context
{
  struct ONNC_RUNTIME_tensor_file* input;
  struct ONNC_RUNTIME_tensor_file* weight;
  uint64_t                         id;
  void (*completed)(uint64_t id, struct ONNC_RUNTIME_tensor_view output);
};

/**
 * ONNC generated entry point.
 * @param context The ONNC Runtime Context.
 */
int model_main(const struct ONNC_RUNTIME_inference_context* context);
```

a

User Application Example

example/runtime/src/client-app.c



```c
void finish(uint64_t id, struct ONNC_RUNTIME_tensor_view output)
{
  const float* const values = output.data;
  const size_t       count  = output.size / sizeof(float);
  printf("[");
  for (size_t idx = 0; idx < count; ++idx) {
    printf("%f, ", values[idx]);
  }
  printf("]");
}

int main(int argc, char* argv[])
{
  if (argc < 3) {
    fprintf(stderr, "usage: %s foo.input foo.weight\n", argv[0]);
    return EXIT_FAILURE;
  }

  struct ONNC_RUNTIME_tensor_file* const input  = open_tensor_file(argv[1]);
  struct ONNC_RUNTIME_tensor_file* const weight = open_tensor_file(argv[2]);

  struct ONNC_RUNTIME_inference_context context = {.input = input, .weight = weight, .id = 0, .completed = finish};

  model_main(&context);

  close_tensor_file(input);
  close_tensor_file(weight);

  return EXIT_SUCCESS;
}
```

a

How to build a user application

(Inside onnc-community docker container)

1. Prepare ONNC Runtime library and onnx tensor convertor **pb2t**

```
$ sudo cp /onnc/onnc-umbrella/build-normal/lib/Runtime/libonnc-rt.a /usr/local/lib
# go to build directory and sync source files
$ cd /onnc/onnc-umbrella/build-normal && ssync
# build convertor and install it into PATH
$ make pb2t && make install
```

2. Build example application and run model inference
example/runtime

```
# prepare build directory
$ cd /onnc/onnc/example/runtime
$ mkdir build && cd build
# prepare service library
$ onnc -mquadruple clang /models/bvlc_alexnet/model.onnx -o ./test.c
# prepare sample input tensor file
$ pb2t /models/bvlc_alexnet/test_data_set_0/input_0.pb ./test.input
$ cp ./test.c ../src/onnc-runtime-service.c
# configure & build example project
$ cmake .. && make
$ ./example/inference test.input test.weight
[0.000043, 0.000046, 0.000024, 0.000011, 0.000114, 0.000469, …
…
…, 0.000035, 0.000148, 0.000964, 0.000134, 0.001431, 0.000448, ]
```

a

References

- ONNC C-backend Tutorial

https://github.com/ONNC/onnc/blob/master/docs/ONNC-C-Backend-Guide.md

Skymizer Taiwan Inc.

CONTACT US

E-mail sales@skymizer.com **Tel** +886 2 8797 8337

HQ 12F-2, No.408, Ruiguang Rd., Neihu Dist., Taipei City 11492, Taiwan
BR Center of Innovative Incubator, National Tsing Hua University,
Hsinchu Taiwan

skymizer

Boost deep learning accelerator
with compiler technology

https://skymizer.com