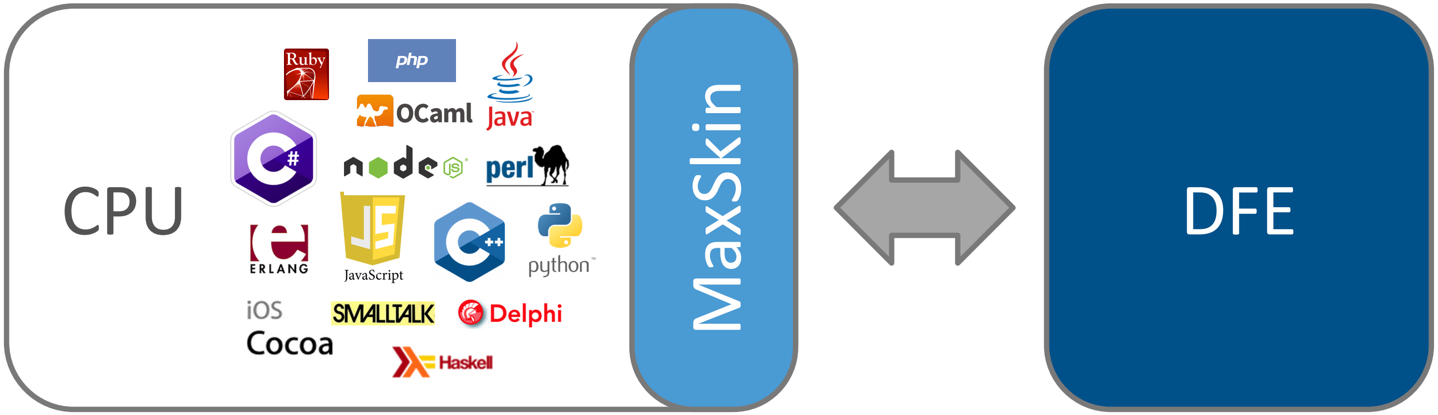
MaxSkins

MAXELER Technologies
MAXIMUM PERFORMANCE COMPUTING

September 2015

Multilingual DFE with MaxSkin

- DFEs can now be taught to speak almost any language
- Regardless of which language you like to program in, you can call a Maxeler Dataflow Engine in it



Simple & Straightforward

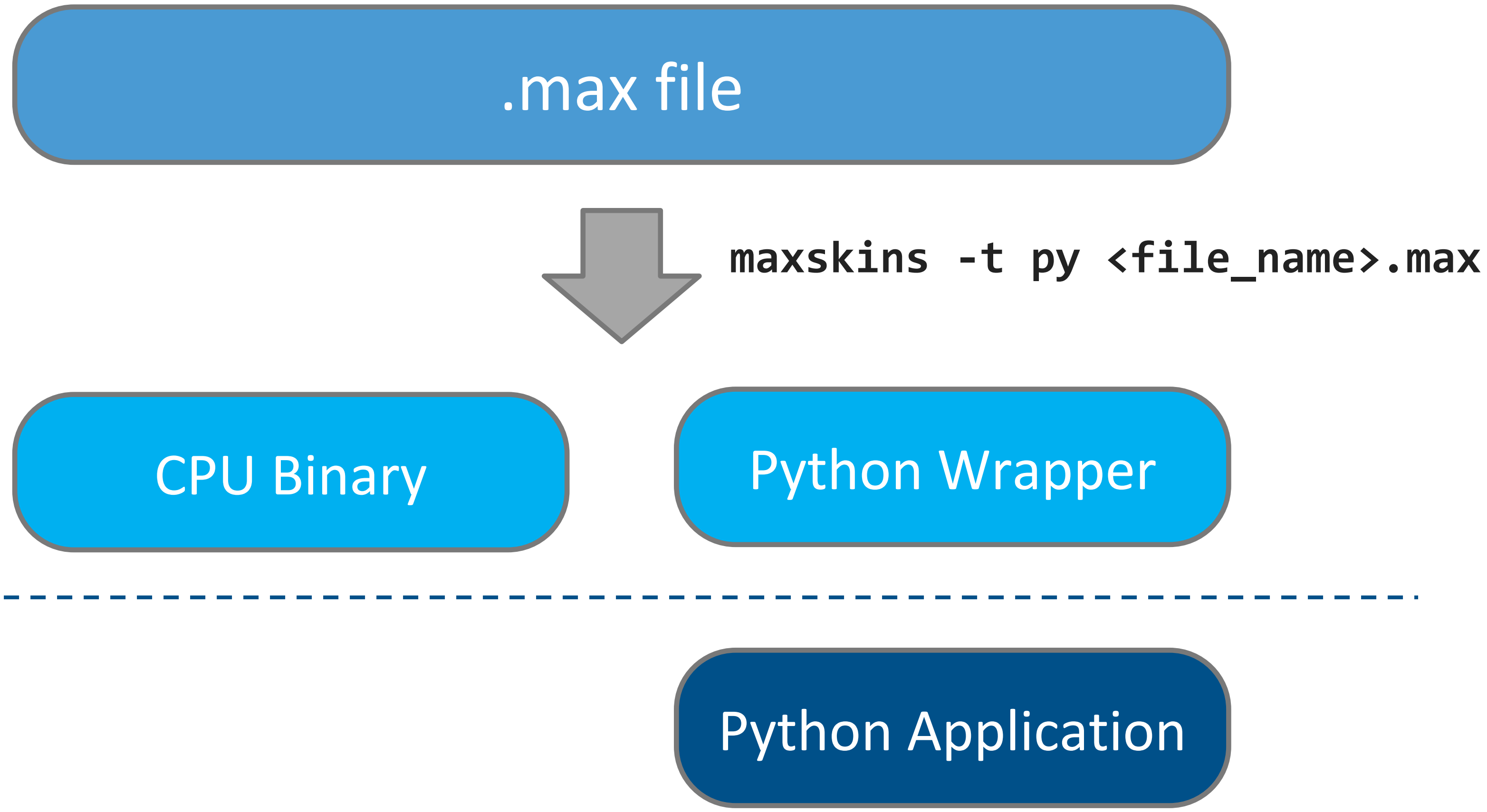
- Allows RPC access to DFEs from many different programming languages
- Automatic generation of wrappers

Ruby
php
OCaml
Java
C#
node JS
perl
ERLANG
JS
JavaScript
C++
python™
iOS
Cocoa
SMALLTALK
Delphi
Haskell

Based on Apache Thrift

- Thrift originally developed at Facebook
- Open sourced in 2007
- Proven solution
- Widely used by likes of Facebook, Evernote, last.fm and Siemens

MaxSkins | Generating Skins



.max file

```
maxskins -t py <file_name>.max
```

CPU Binary

Python Wrapper

Python Application

python™ example

Python example: Correlation

```
def correlate(data, size_timeseries, num_timeseries, correlations):
        # Make socket
        transport = TSocket.TSocket('localhost', 9090)

        # Buffering is critical. Raw sockets are very slow
        transport = TTransport.TBufferedTransport(transport)

        # Wrap in a protocol
        protocol = TBinaryProtocol.TBinaryProtocol(transport)

        # Create a client to use the protocol encoder
        client = correlationService.Client(protocol)

        # Connect!
        transport.open()
```

```
        # Precalculations and preparation of data
        num_timesteps = size_timeseries
        window_size = float(size_timeseries)

        num_bursts = calc_num_bursts(num_timeseries)
        loop_length = client.correlation_get_CorrelationKernel_loopLength()

        precalculations = []
        data_pairs = []

        burst_size = 384 # for anything other than ISCA this should be 384
        in_mem_load = [0] * (num_bursts * burst_size)

        prepare_data_for_dfe(data, size_timeseries, num_timeseries, num_timesteps,
                             window_size, precalculations, data_pairs)
```

...

...

```
# Allocate and send input streams to server
address_loop_length = client.malloc_int32_t(1)
client.send_data_int32_t(address_loop_length, [loop_length])

address_in_mem_load = client.malloc_int32_t(num_bursts * burst_size)
client.send_data_int32_t(address_in_mem_load, in_mem_load)

address_precalculations = client.malloc_double(
    2 * num_timeseries * num_timesteps)
client.send_data_double(address_precalculations, precalculations)

address_data_pairs = client.malloc_double(
    2 * num_timeseries * num_timesteps)
client.send_data_double(address_data_pairs, data_pairs)
```

```
# Allocate memory for output stream on server
address_out_correlation = client.malloc_double(
    num_timesteps * loop_length * correlation_numTopScores *
    correlation_numPipes + num_bursts * 48)
address_out_indices = client.malloc_int32_t(
    2 * num_timesteps * loop_length * correlation_numTopScores *
    correlation_numPipes)

client.correlation_loadLMem(
    num_bursts, address_loop_length, address_in_mem_load)
print 'LMem initialized!'
```

...

…

```
#Executing correlation action
 client.correlation(
     num_bursts,                  # scalar input
     num_timesteps,               # scalar input
     num_timeseries,              # scalar input
     1,                           # scalar input
     window_size,                 # scalar input
     address_precalculations,     # streaming input
     address_data_pairs,          # streaming input
     address_out_correlation,     # streaming output
     address_out_indices)         # streaming output
```

```
# Get output stream from server
out_correlation = client.receive_data_double(
    addr_out_correlation, num_timesteps * loop_len *
    correlation_numTopScores * correlation_numPipes +
                                     num_bursts * 48)
out_indices = client.receive_data_int32_t(
    addr_out_indices, 2 * num_timesteps * loop_len *
    correlation_numTopScores * correlation_numPipes)
loop_len = client.receive_data_int32_t(addr_loop_len, 1)
```

```
# Free allocated memory for streams on server
client.free_server(address_in_mem_load)
client.free_server(address_precalculations)
client.free_server(address_data_pairs)
client.free_server(address_out_correlation)
client.free_server(address_out_indices)
```