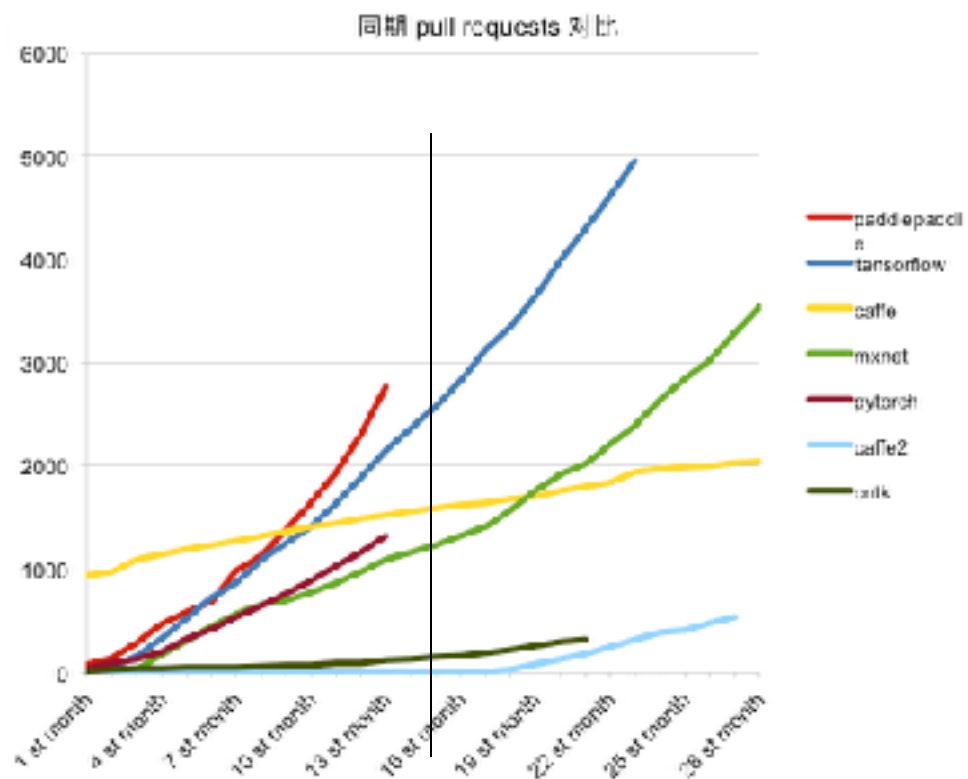
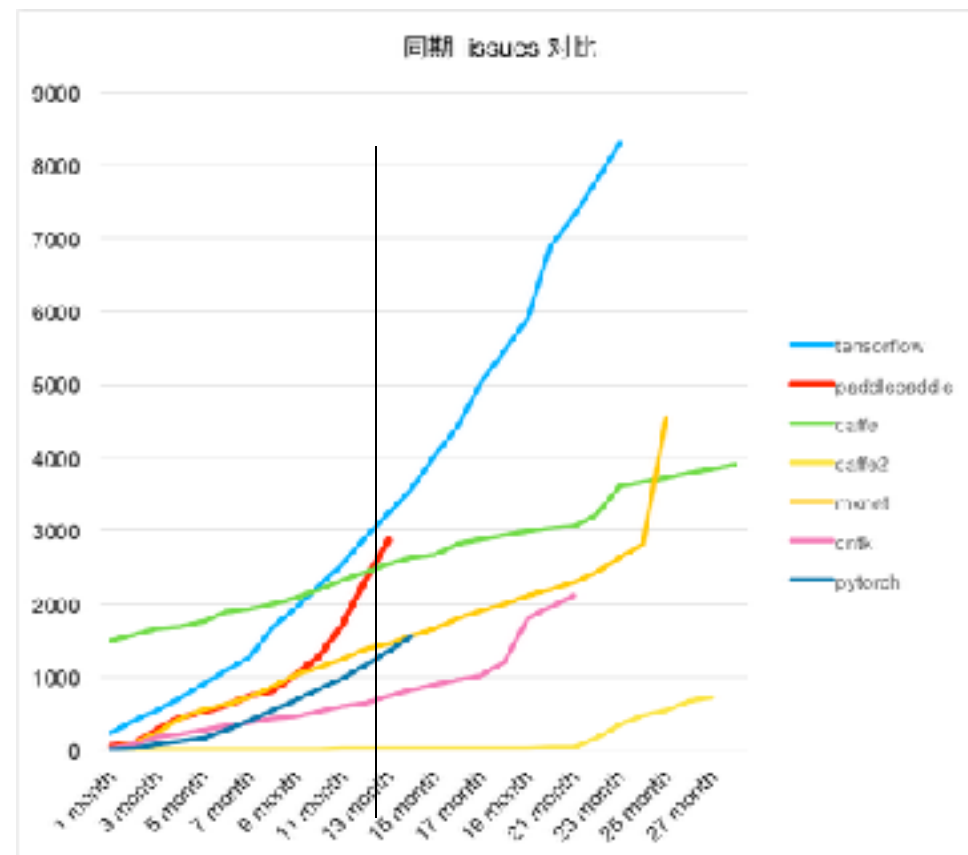


Community Activities



Development activities measured by pull requests

- Each curve shows # **pull requests (PR)** of a DL system
- Curves are horizontally moved so it looks like they were open sourced on the same day.
- When they had been opened for 13 months, PaddlePaddle ranked the top.
- The right ends of these curves shows the current situation. PaddlePaddle has the steepest slope (grows the fastest).



Users' activity measured by issues

- Each curve shows # issues of a DL system
- Curves are horizontally moved.
- In the 13th month, PaddlePaddle ranks the second.
- PaddlePaddle is the second steepest at the right end.

From Framework to Compiler

If-then-else

```
import paddle as pd
# block 0: global block
x = minibatch([10, 20, 30])
y = var(1)
z = minibatch([10, 20, 30])
cond = larger_than(x, 15)

ie = pd.ifelse()
with ie.true_block(): # block 1
    d = pd.layer.add(x, y)
    ie.output(d, pd.layer.softmax(d))
with ie.false_block(): # block 2
    d = pd.layer.fc(z)
    ie.output(d, d+1)
o1, o2 = ie(cond)

namespace pd = paddle;
// block 0: global block
int x = 10;
int y = 1;
int z = 10;
bool cond = false;

int o1, o2;
if (cond) { // block 1
    int d = x + y;
    o1 = z; o2 = pd::layer::softmax(z);
} else { // block 2
    int d = pd::layer::fc(z);
    o1 = d; o2 = d+1;
}
```

Loop / RNN

```
# block 0
x = sequence([10, 20, 30])
m = var(0)
W = var(0.314, param=true)
U = var(0.375, param=true)

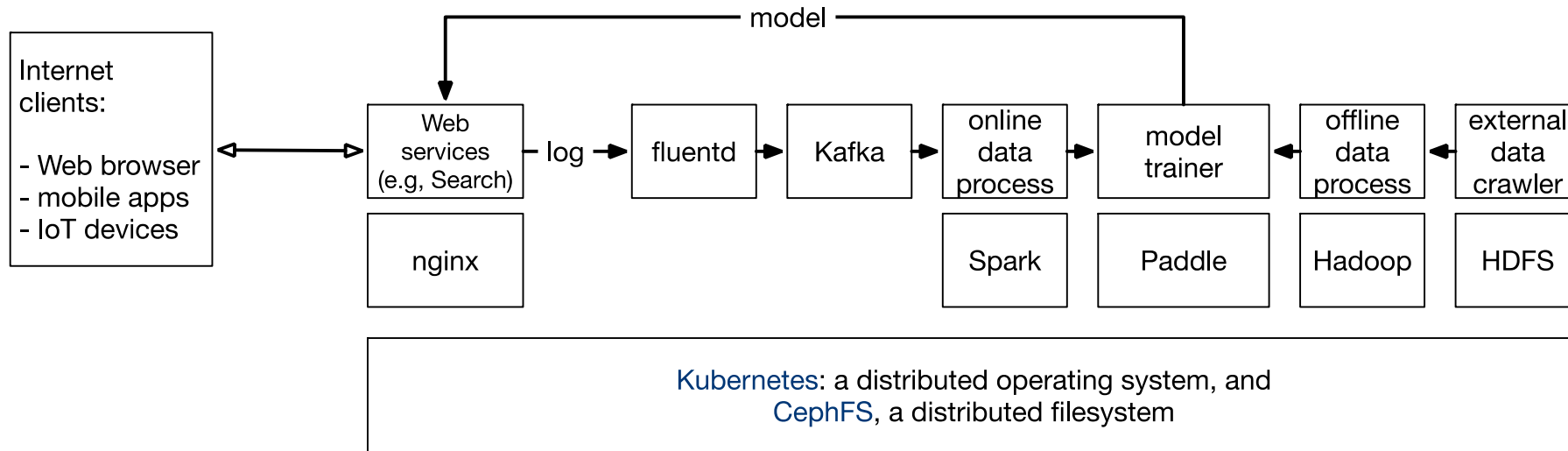
rnn = pd.rnn()
with rnn.step(): # block 1
    x_ = rnn.step_input(x)
    h = rnn.memory(init = m)
    hh = rnn.previous_memory(h)
    a = layer.fc(W, x_)
    b = layer.fc(U, hh)
    s = pd.add(a, b)
    act = pd.sigmoid(s)
    rnn.update_memory(h, act)
    rnn.output(a, b)
o1, o2 = rnn()

// block 0
int* x = [10, 20, 30];
int* m = {0};
int* W = {0.314};
int* U = {0.375};

int len = sizeof(x) / sizeof(x[0])
int mem[len + 1];
int o1[len + 1];
int o2[len + 1];
for (int i = 1; i <= len; ++i) { // block 1
    int x = x[i-1];
    if (i == 1) mem[0] = m;
    int* hh = &(mem[i-1]);
    int a = W * x;
    int b = U * *hh;
    int s = fc_out + hidden_out;
    int act = sigmoid(sum);
    mem[i] = act;
    o1[i] = act; o2[i] = hidden_out;
}
```

- The new PaddlePaddle API makes PaddlePaddle programs almost line-to-line correspondence to a C++/Java program.
- This makes it possible to develop a new DL programming language that brings users out of Python.
- TensorFlow's graph-of-operators describes models, PaddlePaddle's nested-block describe procedures.
- We can use graph compilers (XLA & NNVM) by converting nested-blocks into graphs, or
- invent new highly efficient compiler that converts from nested-blocks into CUDA code directly.

Paddle on Cloud



- The key of a cloud business is to minimum the cost.
- So we run all kinds of jobs on the same set of nodes to make full use of CPU, GPU, memory, network and disk I/O.
- Higher priority jobs (e.g., Web services) often preempt processes in lower priority jobs (e.g., AI).
- PaddlePaddle jobs works with a varying number of processes.

- <http://blog.kubernetes.io/2017/02/run-deep-learning-with-paddlepaddle-on-kubernetes.html>
- <https://www.infoworld.com/article/3167608/artificial-intelligence/baidus-deep-learning-framework-adopts-kubernetes.html>
- <https://www.technologybreakingnews.com/2017/02/baidus-deep-learning-framework-adopts-kubernetes/>
- <https://blog.blackducksoftware.com/paddlepaddle-dives-evolution-of-deep-learning>
- <http://opensourceforu.com/2017/02/baidus-paddlepaddle-deep-learning-framework-gets-kubernetes-support/>
- <https://hackernoon.com/gpus-kubernetes-for-deep-learning-part-1-3-d8eebe0dd6fe>
- <https://sweetcode.io/how-kubernetes-is-sparking-innovation-in-the-docker-ecosystem/>