NUMA-Caffe: NUMA-Aware deep neural networks

[HPCaML – 2019]

Probir Roy, Shuaiwen Leon Song, Sriram Krishnamoorthy, Dipanjan Sengupta, *Xu Liu*

WILLIAM & MARY

CHARTERED 1693

Pacific Northwest

NATIONAL LABORATORY

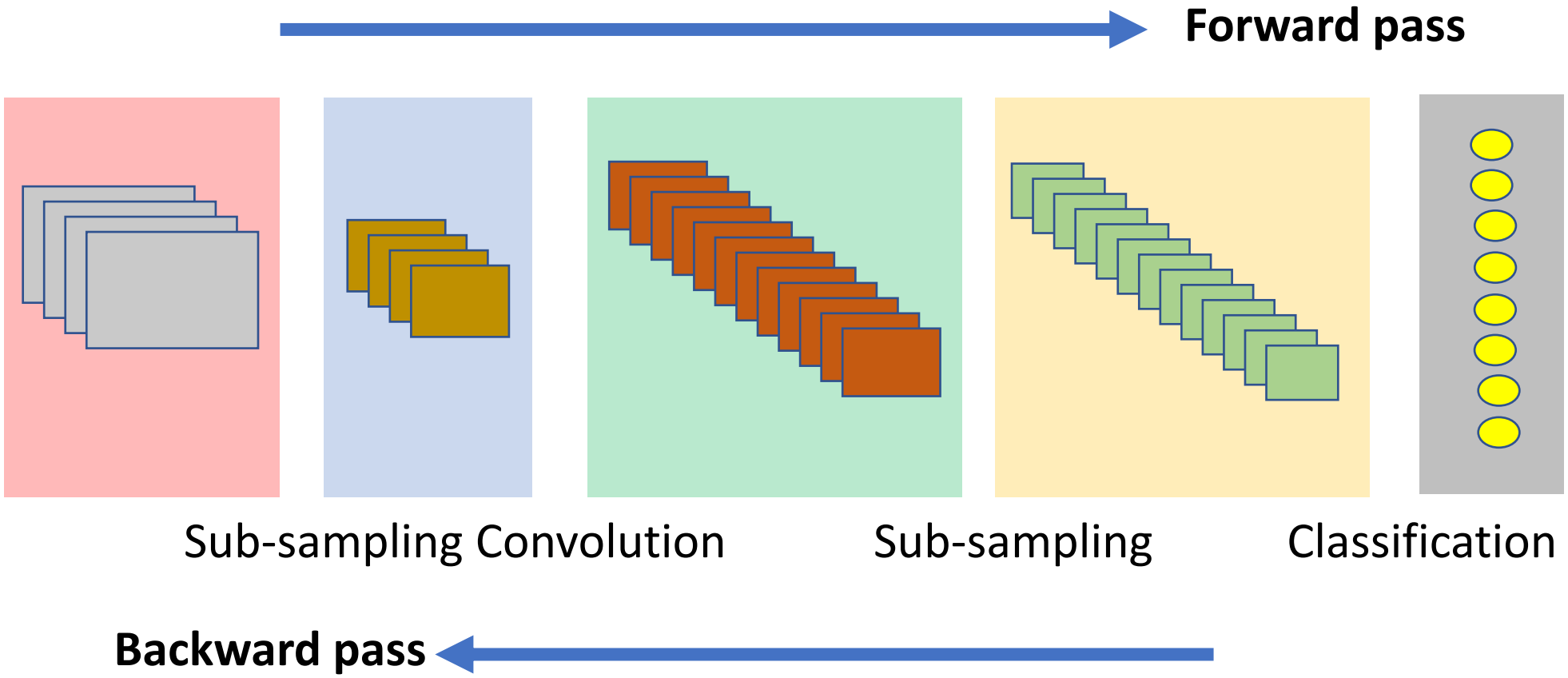
Harnessing power of multi-core and many-core

Top 500 ranking

Rank	Rmax Rpeak (PFLOPS)	Name	Model	Processor	Interconnect	Vendor	Site country, year	Operating system
1 —	143.500 200.795	Summit	Power System AC922	POWER9, Tesla V100	Infiniband EDR	IBM	Oak Ridge National Laboratory United States, 2018	Linux (RHEL)
2 ▲	94.640 125.436	Sierra	Power System S922LC	POWER9, Tesla V100	Infiniband EDR	IBM	Lawrence Livermore National Laboratory United States, 2018	Linux (RHEL)
3 ▼	93.015 125.436	Sunway TaihuLight	Sunway MPP	SW26010	Sunway[20]	NRCPC	National Supercomputing Center in Wuxi China, 2016[20]	Linux (Raise)
4 ▼	61.445 100.679	Tianhe-2A	TH-IVB-FEP	Xeon E5–2692 v2, Matrix-2000	TH Express-2	NUDT	National Supercomputing Center in Guangzhou China, 2013	Linux (Kylin)
5 ▲	21.230 27.154	Piz Daint	Cray XC50	Xeon E5-2690 v3, Tesla P100	Aries	Cray	Swiss National Supercomputing Centre Switzerland, 2016	Linux (CLE)
6 ▼	20.159 41.461	Trinity	Cray XC40	Xeon E5–2698 v3, Xeon Phi 7250	Aries	Cray	Los Alamos National Laboratory United States, 2015	Linux (CLE)
7 ▼	19.880 32.577	AI Bridging Cloud Infrastructure[21]	PRIMERGY CX2550 M4	Xeon Gold 6148, Tesla V100	Infiniband EDR	Fujitsu	National Institute of Advanced Industrial Science and Technology Japan, 2018	Linux
8 ▲	19.477 26.874	SuperMUC-NG[22]	ThinkSystem SD530	Xeon Platinum 8174	Intel Omni-Path	Lenovo	Leibniz Supercomputing Centre Germany, 2018	Linux (SLES)
9 ▼	17.590 27.113	Titan	Cray XK7	Opteron 6274, Tesla K20X	Gemini	Cray	Oak Ridge National Laboratory United States, 2012	Linux (CLE, SLES based)
10 ▼	17.173 20.133	Sequoia	Blue Gene/Q	A2	Custom	IBM	Lawrence Livermore National Laboratory United States, 2013	Linux (RHEL and CNK)

https://en.wikipedia.org/wiki/TOP500

Caffe workflow



Data-parallel convolution layer (BVLC-Caffe)

```
template <typename Dtype>
void ConvolutionLayer<Dtype>::Forward_cpu(const vector<Blob<Dtype>*>& bottom,
      const vector<Blob<Dtype>*>& top) {
  const Dtype* weight = this->blobs_[0]->cpu_data();
  for (int i = 0; i < bottom.size(); ++i) {
    const Dtype* bottom_data = bottom[i]->cpu_data();
    Dtype* top_data = top[i]->mutable_cpu_data();
    for (int n = 0; n < this->num_; ++n) {
      this->forward_cpu_gemm(bottom_data + n * this->bottom_dim_, weight,
          top_data + n * this->top_dim_);
      if (this->bias_term_) {
        const Dtype* bias = this->blobs_[1]->cpu_data();
        this->forward_cpu_bias(top_data + n * this->top_dim_, bias);
      }
    }
  }
}
```

Iterating over images

BLAS parallel

Data-parallel convolution layer (Intel-Caffe)

```
template <typename Dtype>
void ConvolutionLayer<Dtype>::Forward_cpu(const vector<Blob<Dtype>*>& bottom,
    const vector<Blob<Dtype>*>& top) {
 const Dtype* weight = this->blobs_[0]->cpu_data();
 for (int i = 0; i < bottom.size(); ++i) {
  const Dtype* bottom_data = bottom[i]->cpu_data();
  Dtype* top_data = top[i]->mutable_cpu_data();
#ifdef _OPENMP
  #pragma omp parallel if(this->num_of_threads_ > 1) num_threads(this->num_of_threads_)
  {
   #pragma omp for
#endif
   for (int n = 0; n < this->num_; ++n) {
    this->forward_cpu_gemm(bottom_data + n*this->bottom_dim_,
                           weight,
                           top_data + n*this->top_dim_);
    if (this->bias_term_) {
     const Dtype* bias = this->blobs_[1]->cpu_data();
     this->forward_cpu_bias(top_data + n * this->top_dim_, bias);
    }
   }
...
```

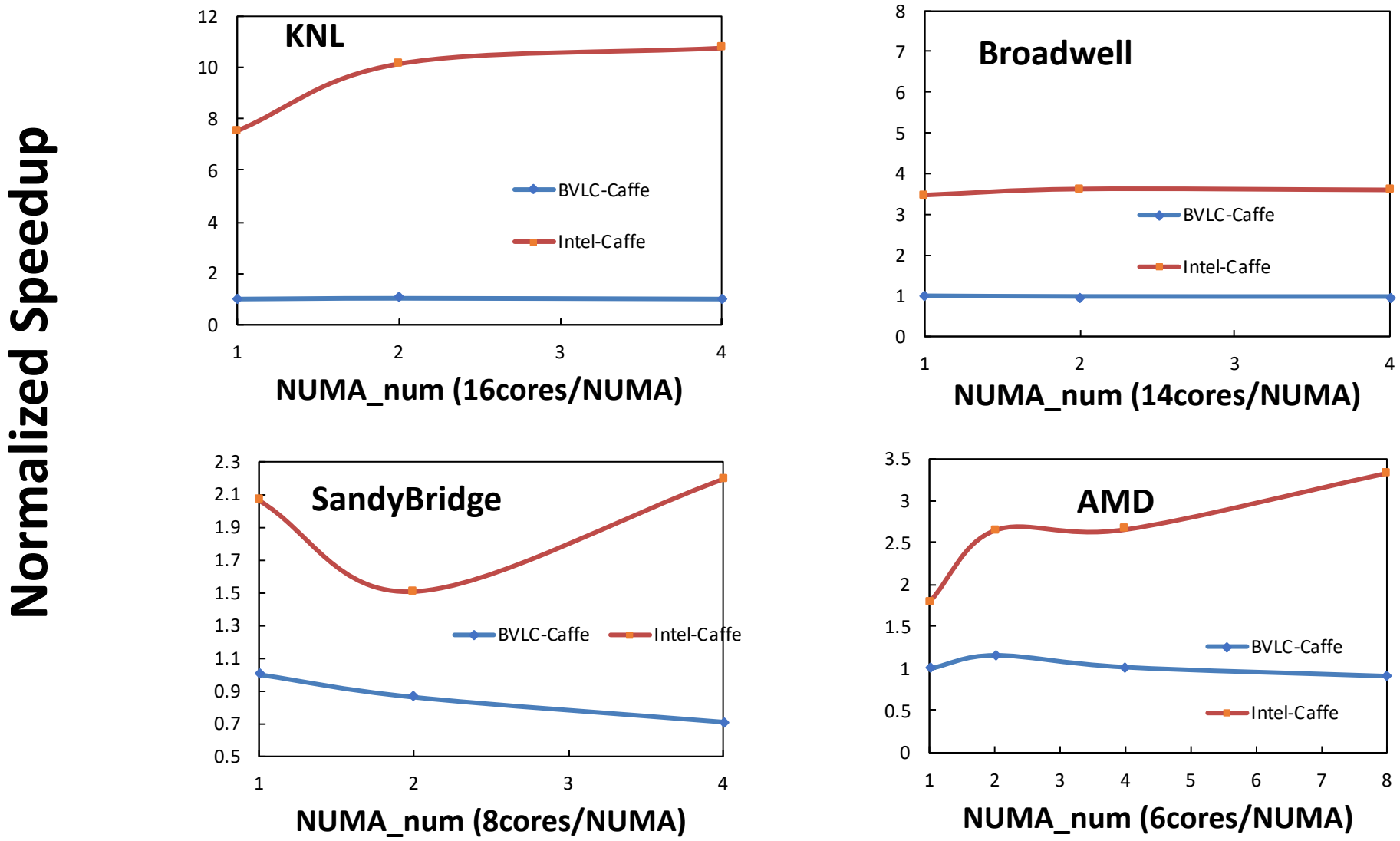
Coarse-grain parallel

Loop coalescing

Testbeds

Processor Type	# sockets	#cores/ socket	L1 data cache	L2 cache	LLC cache	Memory/Socket
Sandy Bridge: Intel Xeon1 E5-4650 2.70GHz	4	8	32 KB	256 KB	20 MB shared	64 GB
KNL: Intel Xeon Phi 7210 1.30GHz	4	16	32 KB	1024 KB	-	4 GB HBM 24 GB DDR4
Broadwell: Intel Xeon E7-4830v4 2.00GHz	4	14	32 KB	256 KB	35 MB shared	251 GB
AMD Opteron Processor 6168 800MHz	8	6	64 KB	512 KB	5 MB shared	16 GB

NUMA scalability of Caffe (Alexnet)



Memory access latency & bandwidth utilization (CIFAR-10)

# of NUMA	Avg DRAM latency	Scaling factor	Avg memory latency	Contribution of DRAM latency (%)
1 NUMA	13.6	1.00	69.2	19.66
2 NUMA	20.0	1.47	153.2	13.03
4 NUMA	64.5	4.75	173.7	37.15

Memory Access Latency of CIFAR10 Training by IntelCaffe

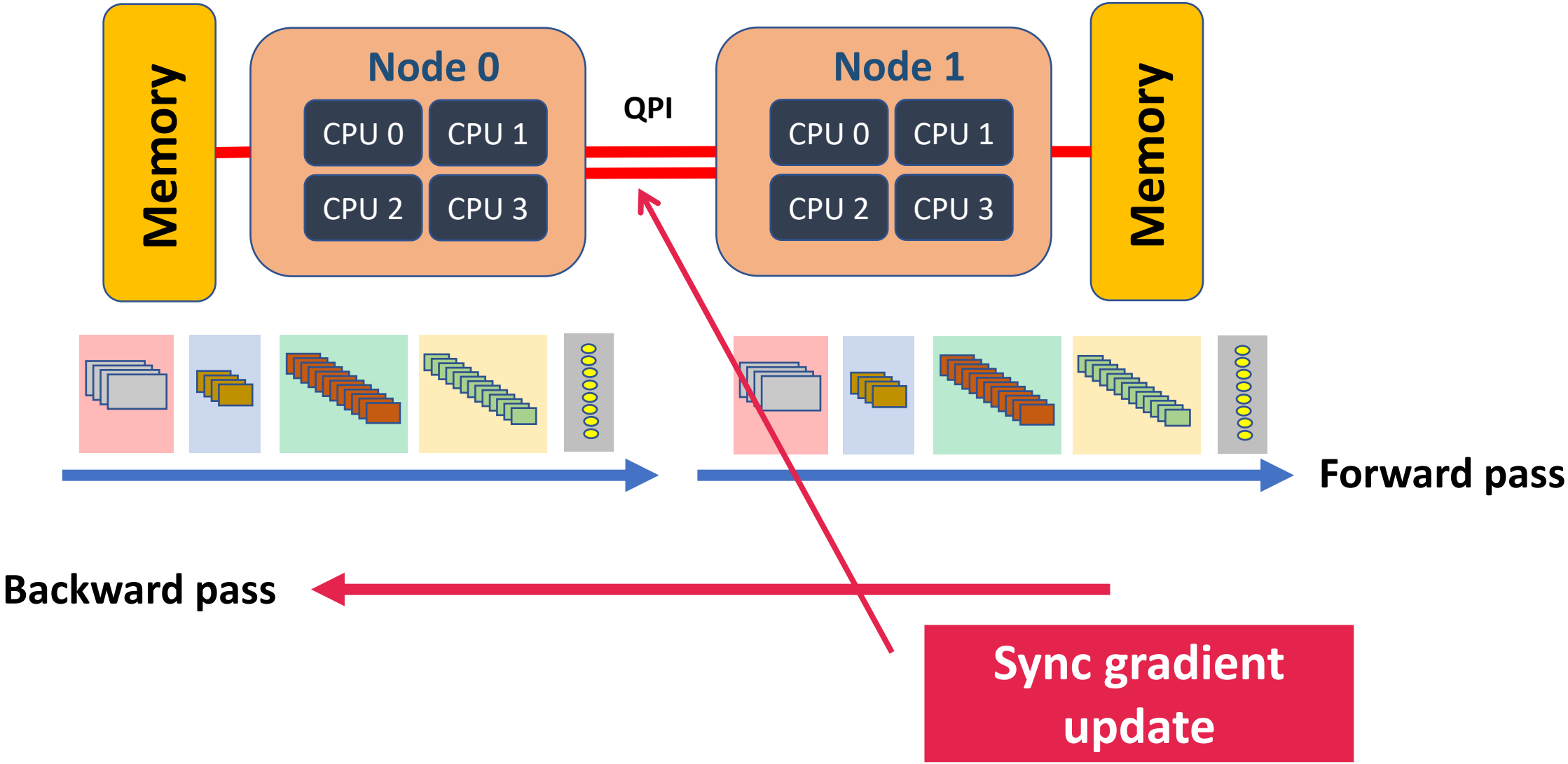
Sockets	Total GB/sec	Read GB/sec	Write GB/sec
Socket 0	12.6	5.8	6.08
Socket 1	0.73	0.34	0.39
Socket 2	1.02	0.57	0.45
Socket 3	0.65	0.3	0.35

Average Bandwidth Utilization of CIFAR10 Training by IntelCaffe

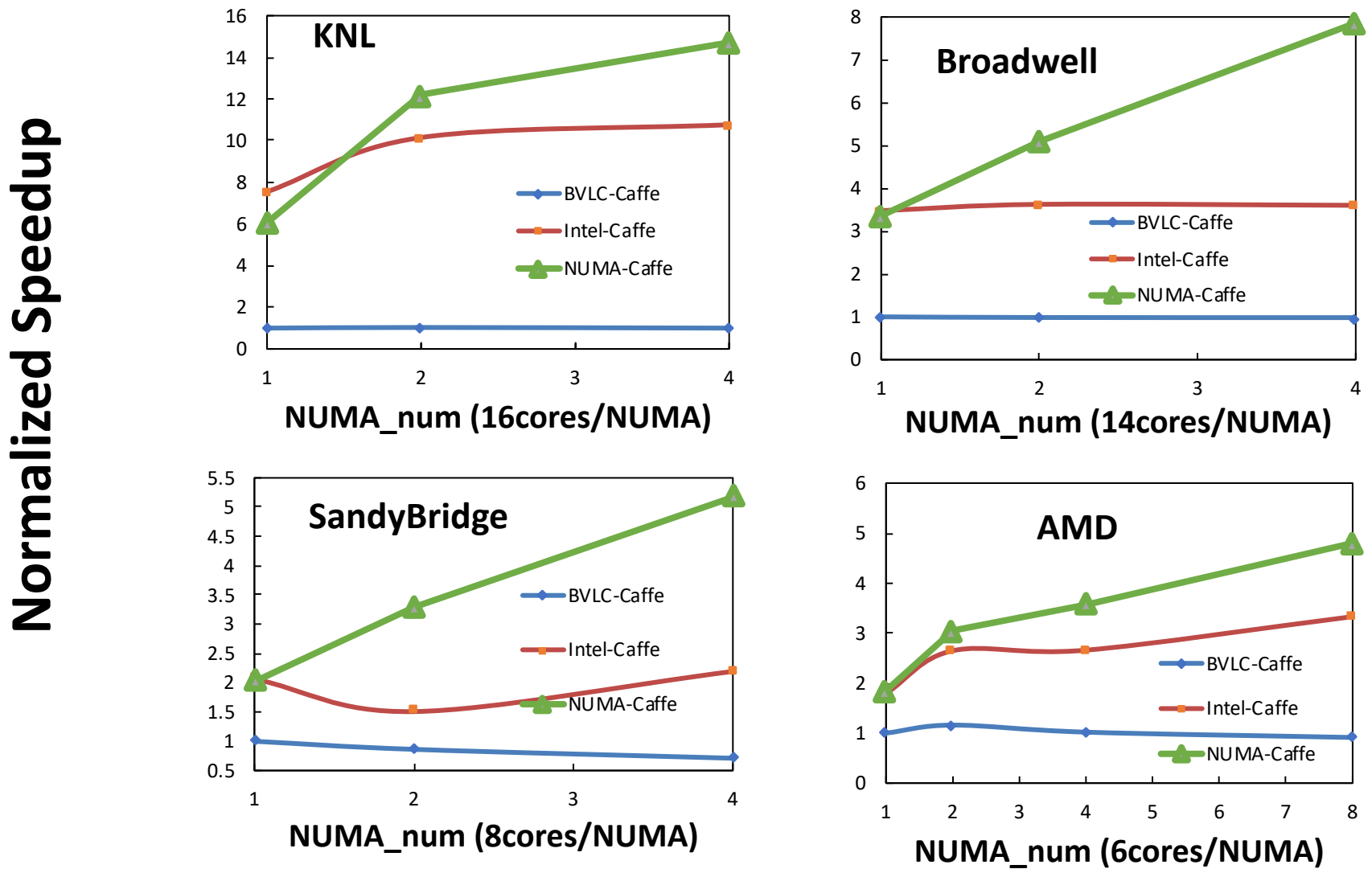
Layer level breakdown node-load miss (CIFAR-10)

Layer	Function	%Node-load miss
Convolution layer	weight_cpu_gemm	36.1%
	forward_cpu_gemm	11.4%
	Im2col_cpu	5.7%
	Backward_cpu_gemm	4.0%
Pooling layer	Backward_cpu	3.3%
	Forward_cpu	2.0%
ReLU layer	Backward_cpu	3.7%

NUMA-Caffe workflow



NUMA-Caffe scalability (Alexnet)



NUMA-Caffe achieves 2.4× speedup over Intel-Caffe

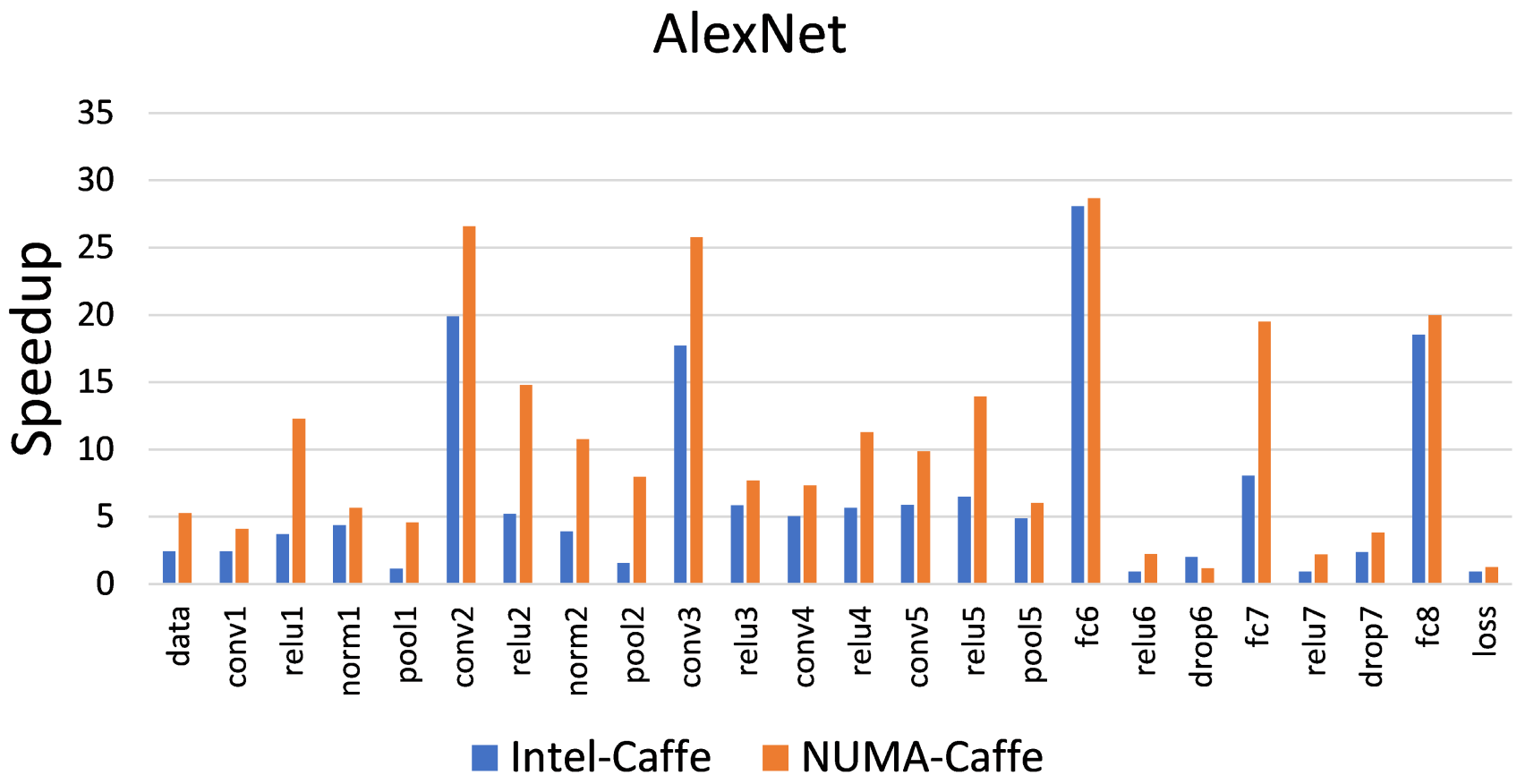
Package bandwidth utilization of NUMA-Caffe

Sockets	Total GB/sec	Read GB/sec	Write GB/sec
Socket 0	3.69	1.99	1.70
Socket 1	3.37	1.76	1.61
Socket 2	3.8	1.96	1.83
Socket 3	3.4	1.76	1.64

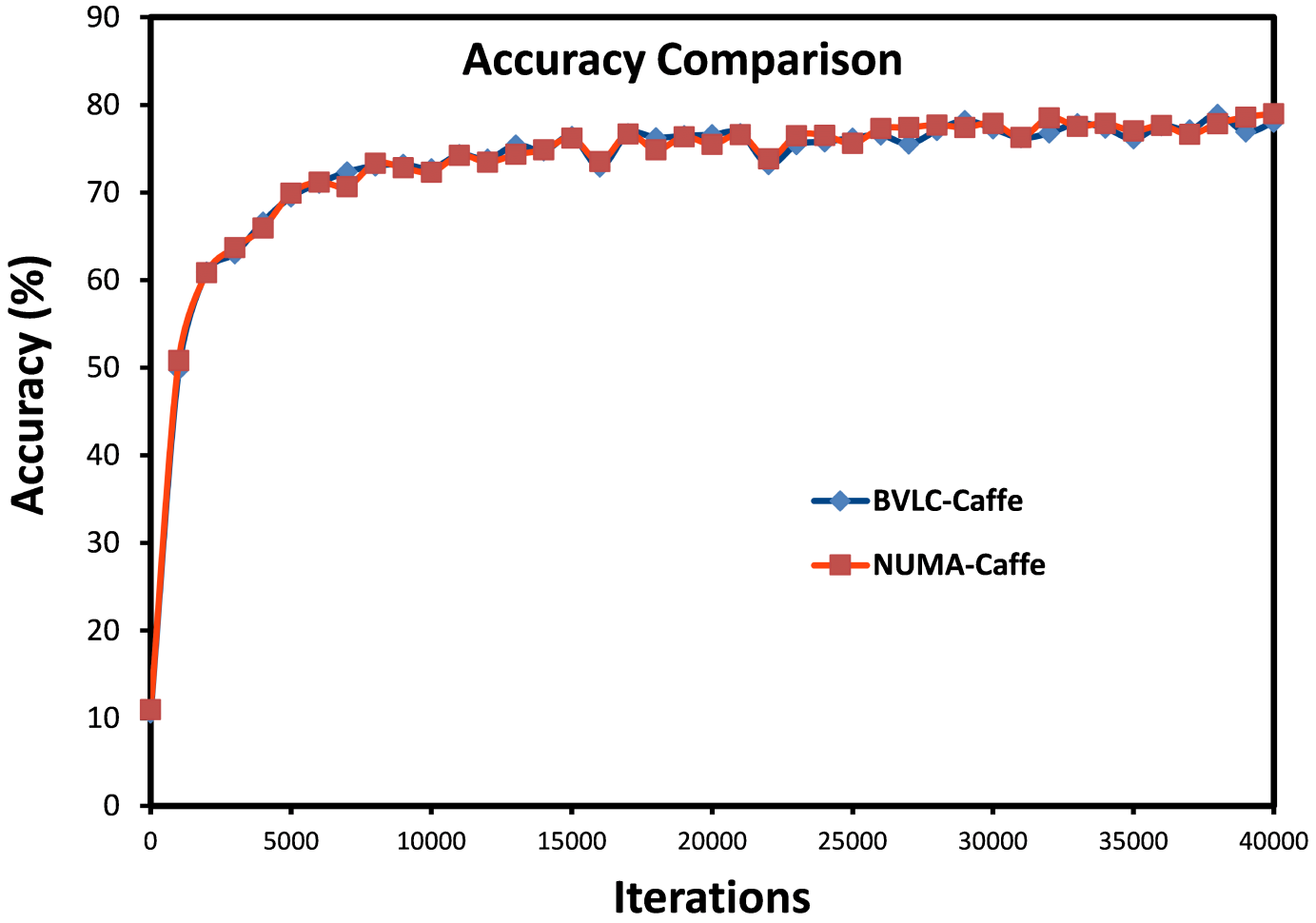
Average Bandwidth Utilization of CIFAR10 Training by NUMA-Caffe

NUMA-Caffe achieves 7× less memory access latency compared to Intel-Caffe

Layer-level speedup of AlexNet



Empirical validation of accuracy (CIFAR-10)



Conclusion

- Systematic study of NUMA-scalability challenges of Caffe
  - Variants of Caffe: BVLC-Caffe and Intel-Caffe
  - Three networks: CIFAR-10, AlexNet, GoogLeNet
  - Testbeds: Intel KNL, Sandy Bridge, Broadwell, AMD Opteron
- Root-cause analysis of NUMA bottleneck
- Proposed NUMA-Caffe
- Evaluate performance and accuracy