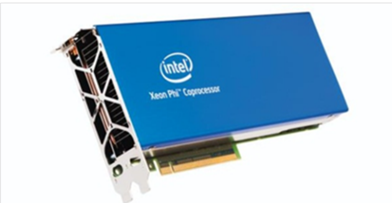
Performance Optimization of Deep Learning Frameworks on Modern Intel Architectures

ElMoustapha Ould-Ahmed-Vall, AG Ramesh, Vamsi Sripathi and Karthik Raman

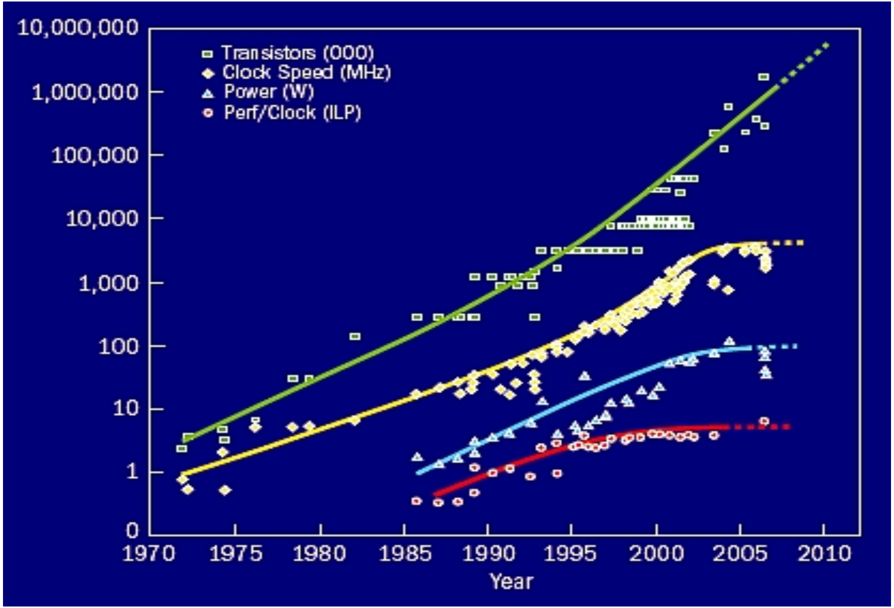
Representing the work of many at Intel



Agenda

- Optimization matters on modern architectures
- Intel's recent Xeon and Xeon Phi products
- Introduction to Deep Learning
- Optimizing DL frameworks on IA
  - Key challenges
  - Optimization techniques
  - Performance data
  - DL scaling

Moore’s Law Goes on!



Increasing clock speeds -> more cores + wider SIMD (Hierarchical parallelism)

Combined Amdahl's Law for Vector Multicores*

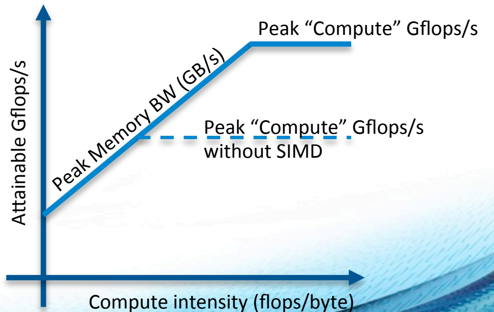
$$\boldsymbol{Speedup} = (1/Serial\downarrow frac + 1 - Serial\downarrow frac\ /\boldsymbol{NumCores}\ ) * (1/Scalar\downarrow frac + 1 - Scalar\downarrow frac\ /\boldsymbol{VectorLength}\ )$$

Goal: *Reduce **Serial Fraction*** and *Reduce **Scalar Fraction*** of Code

Ideal Speedup: **NumCores*VectorLength (requires zero scalar, zero serial work)**

<u>Compute Bound Performance</u>
Most kernels of ML codes are compute bound
i.e. raw FLOPS matter

<u>Roofline Model</u>
Gflops/s = min (Peak Gflops/s, Stream BW * flops/byte)



Overview of Current Generation of Intel Xeon and Xeon Phi Products

Current Intel® Xeon Platforms

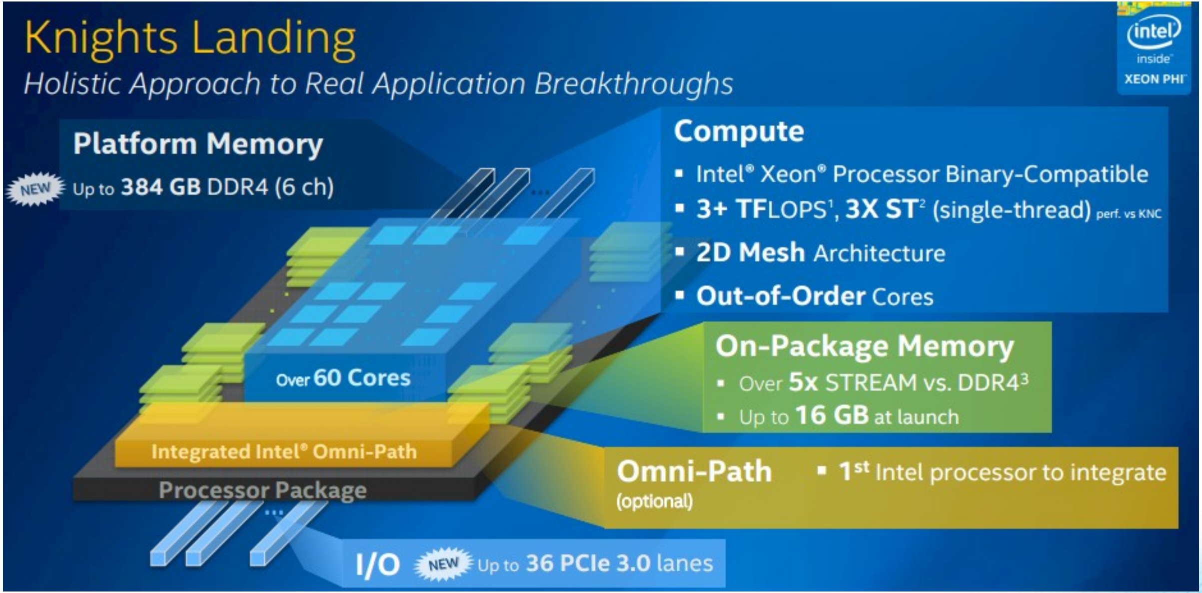
45nm Process Technology	32nm Process Technology		22nm Process Technology		14nm Process Technology
Nehalem	Westmere	Sandy Bridge	Ivy Bridge	Haswell	Broadwell
NEW Intel® Microarchitecture (Nehalem)	Intel Microarchitecture (Nehalem)	NEW Intel Microarchitecture (Sandy Bridge)	Intel Microarchitecture (Sandy Bridge)	NEW Intel Microarchitecture (Haswell)	Intel Microarchitecture (Haswell)
TOCK	*TICK*	*TOCK*	*TICK*	*TOCK*	*TICK*

Latest released – Broadwell (14nm process)

- Intel's foundation of HPC and ML performance
- Suited for full scope of workloads
- Industry leading performance/watt for serial & highly parallel workloads.
- Upto 22 cores / socket (Broadwell-EP) (w/ Hyper-Threading technology)

Software optimization helps maximize benefit and adoption of new features

2nd Generation Intel® Xeon Phi™ Platform



Intel® AVX Technology

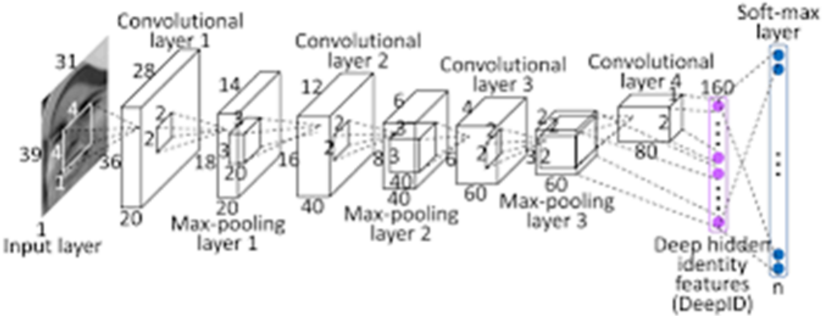
SNB/IVB	HSW/BDW	SKX & KNL
256b AVX1 Flops/Cycle: 16 SP / 8 DP	256b AVX2 Flops/Cycle: 32SP / 16 DP (FMA)	512b AVX512 Flops/Cycle: 64SP / 32 DP (FMA)

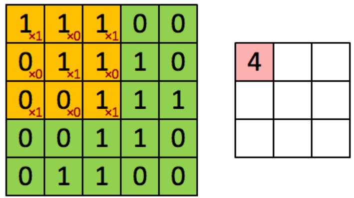
AVX	AVX2
256-bit basic FP	Float16 (IVB 2012)
16 registers	256-bit FP FMA
NDS (and AVX128)	256-bit integer
Improved blend	PERMD
MASKMOV	Gather
Implicit unaligned	

AVX512
512-bit FP/Integer
32 registers
8 mask registers
Embedded rounding
Embedded broadcast
Scalar/SSE/AVX "promotions"
Native media additions
HPC additions
Transcendental support
Gather/Scatter

Overview of Deep Learning and DL Frameworks

Deep Learning – Convolutional Neural Network

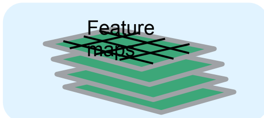


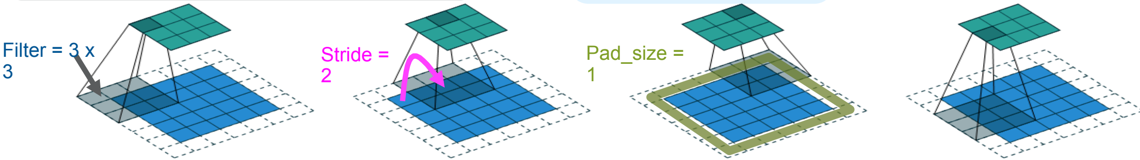


Image

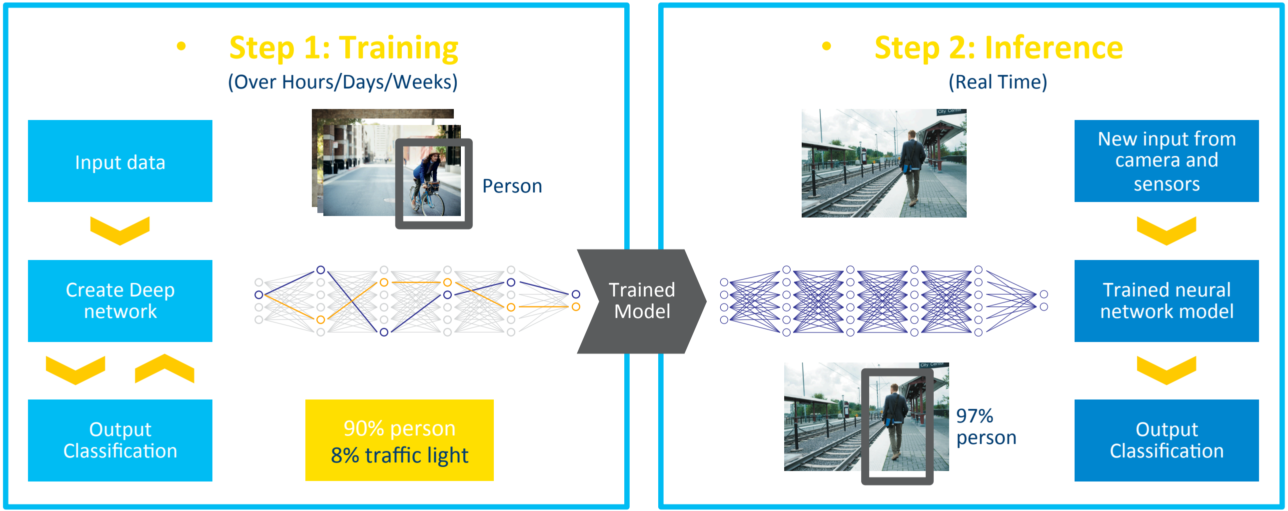
Convolved Feature

Convolution Parameters:
Number of outputs/feature-maps: < 4 >
Filter size: < 3 x 3 >
Stride: < 2 >
Pad_size (for corner case): <1>





Deep Learning: Train Once Use Many Times



Deep Learning: Why Now?

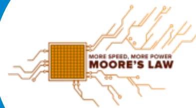
Bigger Data

Image: 1000 KB / picture

Audio: 5000 KB / song

Video: 5,000,000 KB / movie

Better Hardware



Transistor density doubles every 18 months

Cost / GB in 1995: $1000.00

Cost / GB in 2015: $0.03

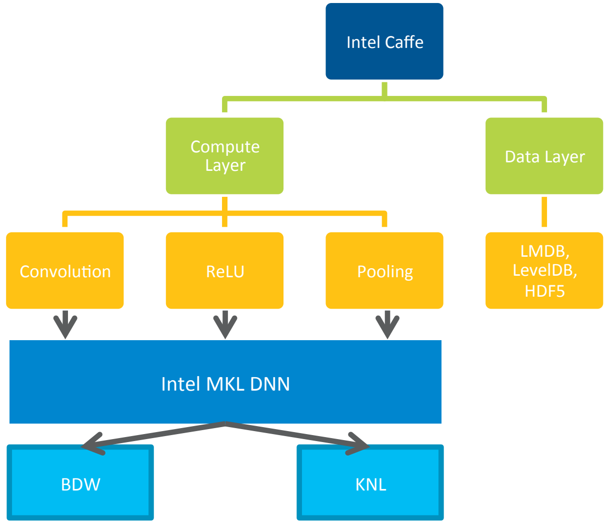
Smarter Algorithms

Advances in algorithm innovation, including neural networks, leading to better accuracy in training models

Intel Caffe – ML Framework

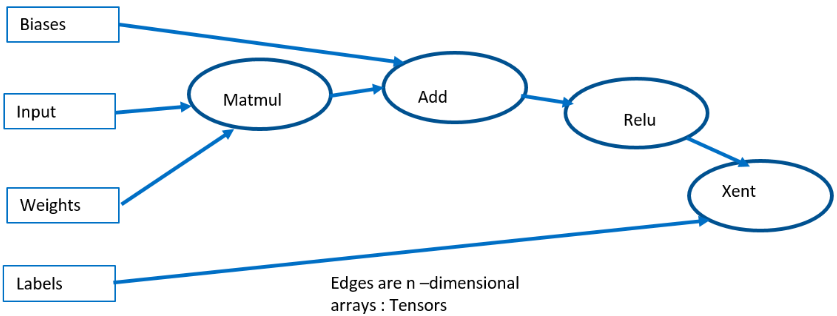
Optimized for Xeon and Xeon Phi Products

- ❑ Fork of BVLC Caffe by Intel to optimize for IA
- ❑ Leverages Intel MKL Deep Neural Network (DNN) API's
- ❑ Optimized for BDW (AVX2) and KNL (MIC_AVX512)
- ❑ https://github.com/intel/caffe



Tensorflow ™ : Open Source ML Framework (Google)

- **Computation is a Dataflow Graph with Tensors**
- General computing mathematical framework – widely used for
    - Deep Neural Networks
    - Other machine learning algorithms
    - HPC applications
- Key computational kernels, extendable user operations
- Core in C++, front end wrapper in python
- Multi node support using GRPC
    - Google Remote Procedural Calls



Example from Jeff Dean's presentation

Optimizing Deep Learning Frameworks

Performance Optimization on Modern Platforms

Hierarchical Parallelism

Coarse-Grained / multi-node

Domain decomposition

Fine-Grained Parallelism / within node

Sub-domain: 1) Multi-level domain decomposition (ex. across layers)
2) Data decomposition (layer parallelism)

Scaling

Improve load balancing

Reduce synchronization events, all-to-all comms

Utilize all the cores

OpenMP, MPI, TBB…

Reduce synchronization events, serial code

Improve load balancing

Vectorize/SIMD

Unit strided access per SIMD lane

High vector efficiency

Data alignment

Efficient memory/cache use

Blocking

Data reuse

Prefetching

Memory allocation

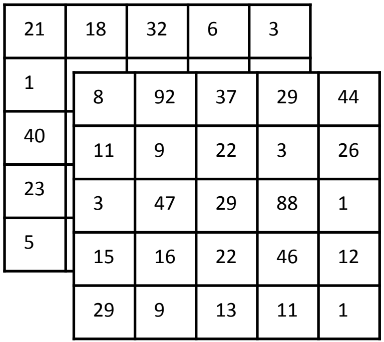
Intel Strategy: Optimized Deep Learning Environment

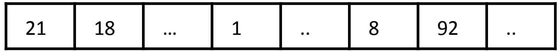
	**Fuel** the development of vertical solutions
Intel® Deep Learning SDK	**Accelerate** design, training, and deployment
Caffe, theano, TensorFlow, torch, Microsoft CNTK, dmlc mxnet	**Drive** optimizations across open source deep learning frameworks
Intel® Math Kernel Library (Intel® MKL); Intel® MKL-DNN	**Maximum** performance on Intel architecture
Training: intel XEON PHI inside + Intel® Omni-Path Architecture (Intel® OPA); Inference: intel XEON inside + ALTERA Arria 10 FPGA • SoC	**Deliver** best single node and multi-node performance

Training **Inference**

Example Challenge 1: Data Layout Has Big Impact on Performance

- Data Layouts impacts performance
  - Sequential access to avoid gather/scatter
  - Have iterations in inner most loop to ensure high vector utilization
  - Maximize data reuse; e.g. weights in a convolution layer
- Converting to/from optimized Layout is some times less expensive than operating on unoptimized Layout



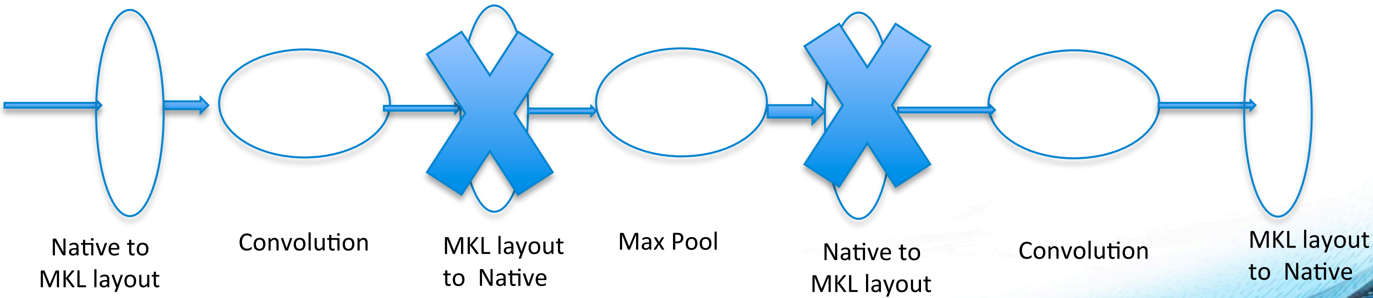


Better optimized for some operations

vs

Example Challenge 2: Minimize Conversions Overhead

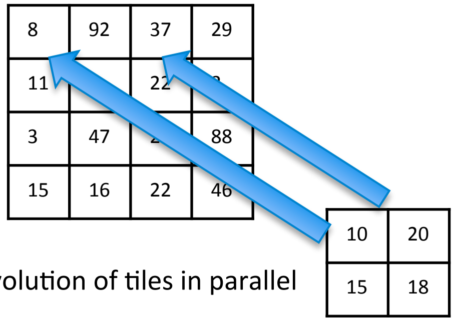
- End to end optimization can reduce conversions
- Staying in optimized layout as long as possible becomes one of the tuning goals
- Minimize the number of back and forth conversions
  - Use of graph optimization techniques



Example Challenge 3: Ensuring Enough Parallelism to Leverage all Cores

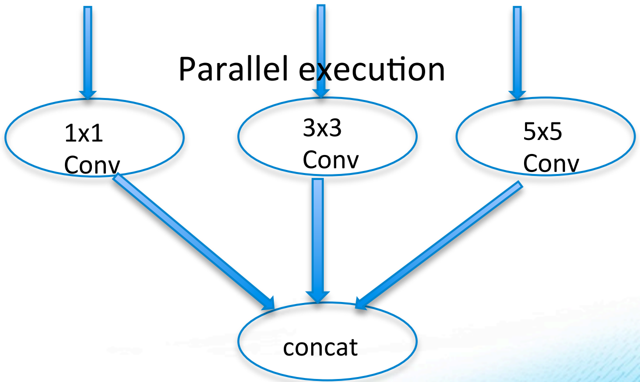
- Maximize parallelism to use all cores efficiently

- Intra operation/layer parallelism within operators (OpenMP)

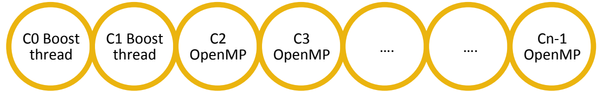


Convolution of tiles in parallel

Inter operation parallelism across operators



Example Challenge 4: Optimizing the Data Layer

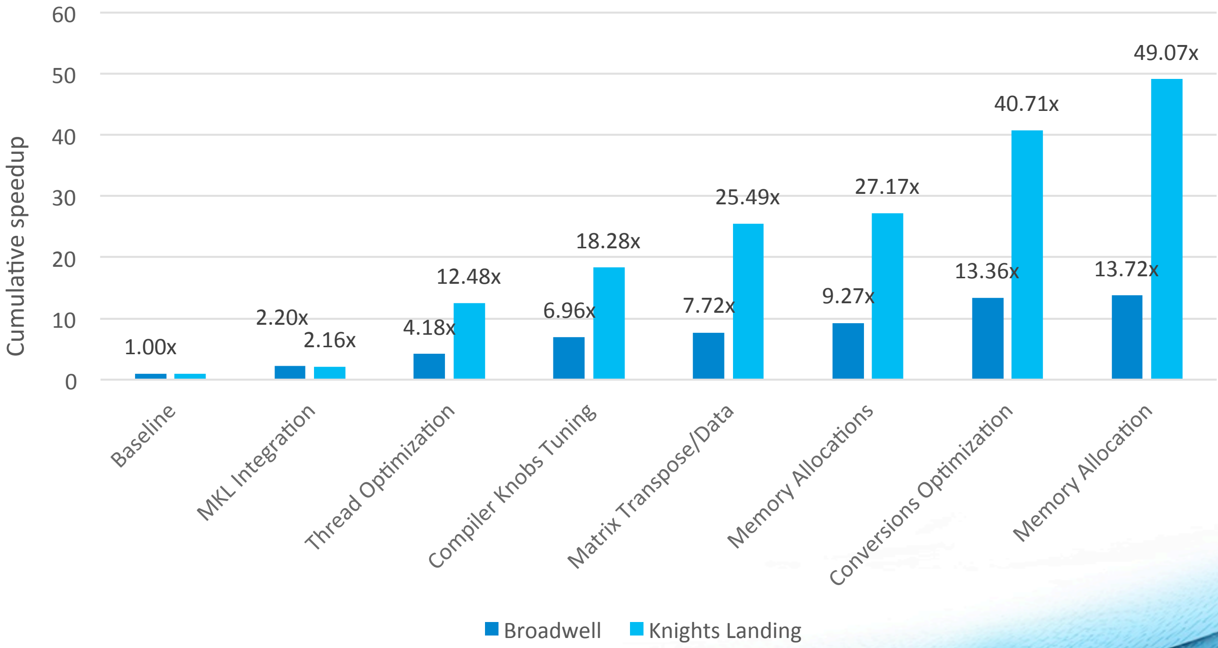


- Data Layer comprises 3 major ops
  - Read data
  - Decode data: e.g. JPEG decode, decompression
  - Transform data
- Result of read, decode & transform is input to DNN layers
- Reduce number of cores dedicated to feed DNN
  - IO optimization: consider compression
  - Decode: consider LMDB instead of JPEG
  - Resizing/data processing: consider pre-processing
  - Then vectorize, parallelize

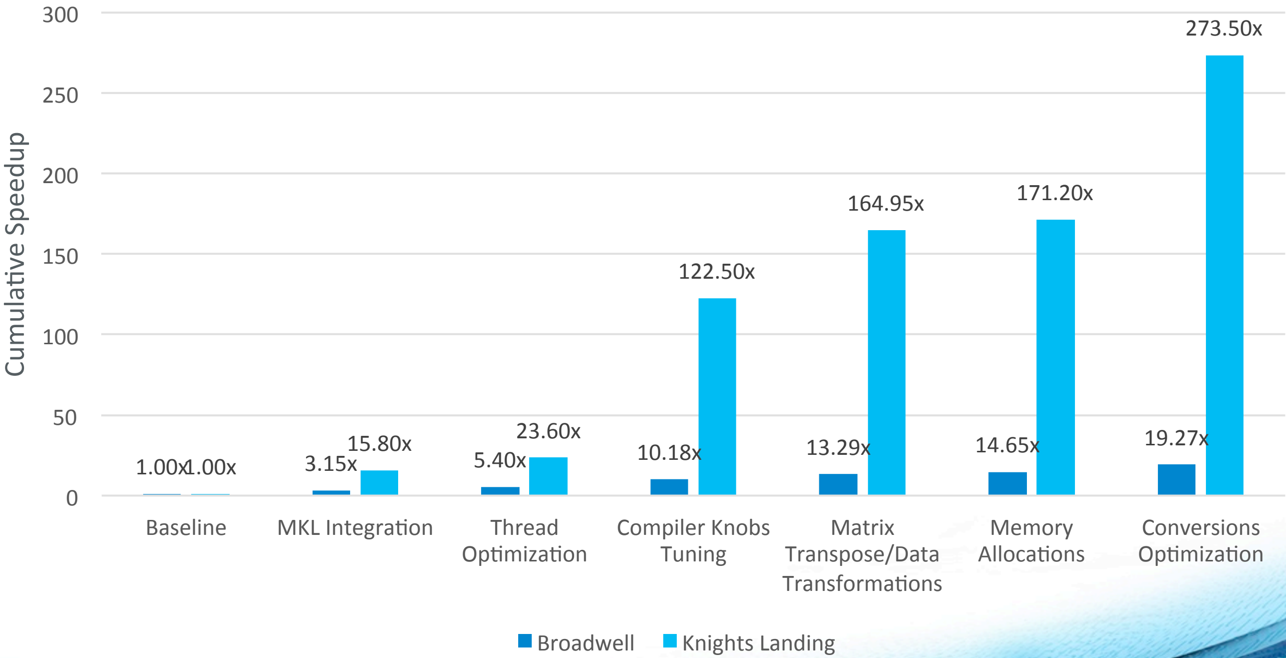
Optimizing Deep Learning Frameworks for Intel® Architecture

- Leverage high performant compute libraries and tools
  - e.g. Intel® Math Kernel Library, Intel® Python, Intel® Compiler etc.
- Data Format/Shape:
  - Right format/shape for max performance: blocking, gather/scatter
- Data Layout:
  - Minimize cost of data layout conversions
- Parallelism:
  - Use all cores, eliminate serial sections, load imbalance
- Other Functions/Primitives (un-optimized in libraries):
  - Optimize via compiler knobs, improve existing implementations
- Memory allocation
  - unique characteristics and ability to reuse buffers
- Data layer optimizations:
  - parallelization, vectorization, IO
- Optimize hyper parameters:
  - e.g. batch size for more parallelism
  - learning rate and optimizer to ensure accuracy/convergence

AlexNet Optimization Progression



VGG Optimization Progression



Configuration details

Intel® Xeon™ processor E5-2699v4 (22 Cores, 2.2 GHz), 128GB DDR memory, Centos 7.2 based on Red Hat* Enterprise Linux 7.2

Intel® Xeon Phi™ processor 7250 (68 Cores, 1.4 GHz, 16GB MCDRAM: Flat mode), 96GB DDR memory, Centos 7.2 based on Red Hat* Enterprise Linux 7.2

AlexNet and VGG benchmarks:

https://github.com/soumith/convnet-benchmarks

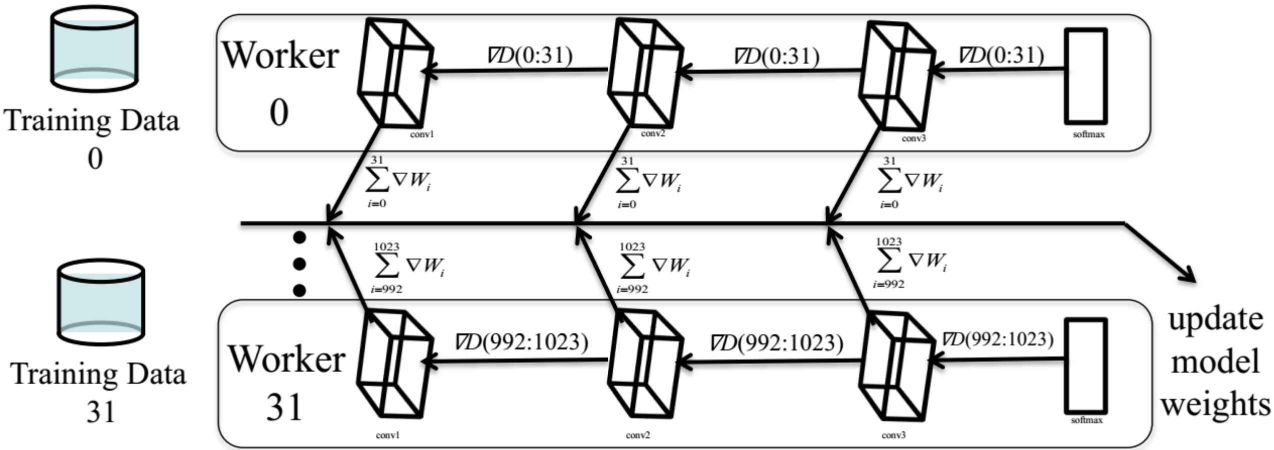
Multi-Node Distributed Training

- Model Parallelism
  - Break the model into *N* nodes
  - The same data is in all the nodes

- Data Parallelism
  - Break the dataset into *N* nodes
  - The same model is in all the nodes
  - Good for networks with few weights, e.g. GoogLeNet

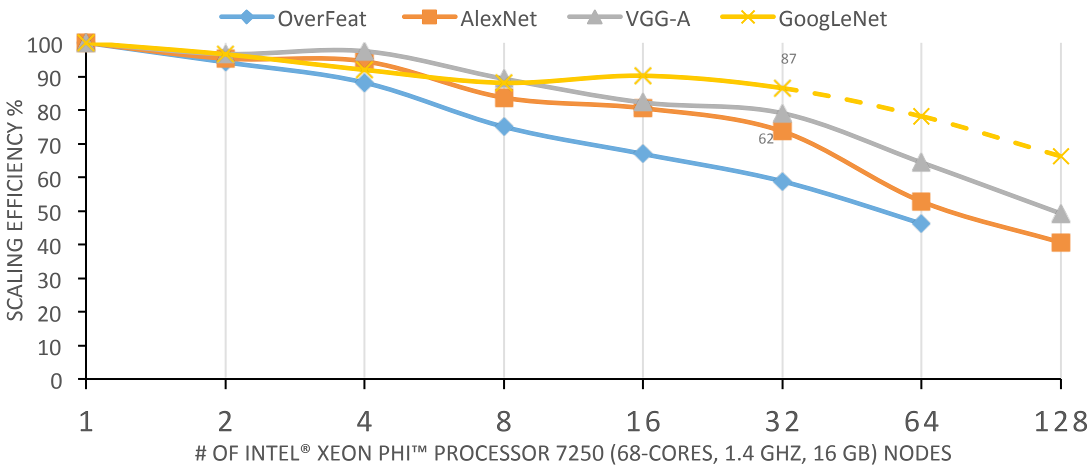
- You can use either model or data parallelism or a hybrid of both

Data Parallelism



Scaling Efficiency: Intel® Xeon Phi™ Processor

Deep Learning Image Classification Training Performance : MULTI-NODE Scaling

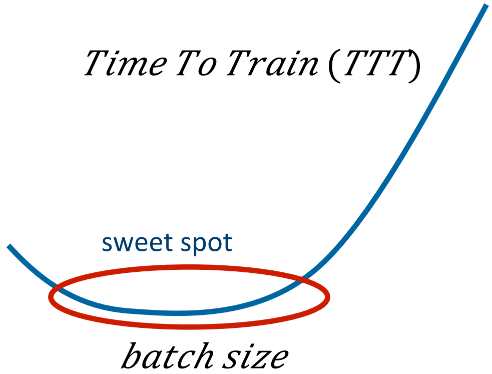


Configurations:

- Intel® Xeon Phi™ Processor 7250 (68 Cores, 1.4 GHz, 16GB MCDRAM), 128 GB memory, Red Hat* Enterprise Linux 6.7, Intel® Optimized Framework

Multi-node Challenges

- Need to optimize both compute (iteration) and communication (weight updates)
- More nodes mean higher batch per iteration
  - Enough work for each node
- Optimized hyper parameters (e.g. Batch Size)
  - Time to Train: increases with batch size
  - Accuracy: batch size impacts convergence and accuracy
- Communication overheads if small per node batch
  - e.g. Total batch size = 1024
    - 1024 nodes : Batch size = 1 per node – **communication dominates**
    - 64 nodes each : Batch size = 16 per node – **computation dominates**



Summary

- Don’t be fooled by performance of DL workloads when using unoptimized frameworks
- Significant performance headroom from optimization on Xeon and Xeon Phi
    - Close to 300x speedup in certain topologies
- Traditional vectorization and parallelization strategies apply
- Other unique performance challenges: hyper parameters, data layer, inter/intra layer parallelization, etc.
- Call to action:
    - Try Intel optimized frameworks available today, more to come soon

Legal Disclaimers

- Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families: Go to: Learn About Intel® Processor Numbers http://www.intel.com/products/processor_number
- Some results have been estimated based on internal Intel analysis and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance.
- Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.
- Intel does not control or audit the design or implementation of third party benchmarks or Web sites referenced in this document. Intel encourages all of its customers to visit the referenced Web sites or others where similar performance benchmarks are reported and confirm whether the referenced benchmarks are accurate and reflect performance of systems available for purchase.
- Relative performance is calculated by assigning a baseline value of 1.0 to one benchmark result, and then dividing the actual benchmark result for the baseline platform into each of the specific benchmark results of each of the other platforms, and assigning them a relative performance number that correlates with the performance improvements reported.
- SPEC, SPECint, SPECfp, SPECrate, SPECpower, SPECjbb, SPECompG, SPEC MPI, and SPECjEnterprise* are trademarks of the Standard Performance Evaluation Corporation. See http://www.spec.org for more information.
- TPC Benchmark, TPC-C, TPC-H, and TPC-E are trademarks of the Transaction Processing Council. See http://www.tpc.org for more information.
- No computer system can provide absolute reliability, availability or serviceability. Requires an Intel® Xeon® processor E7-8800/4800/2800 v2 product families or Intel® Itanium® 9500 series-based system (or follow-on generations of either.) Built-in reliability features available on select Intel® processors may require additional software, hardware, services and/or an internet connection. Results may vary depending upon configuration. Consult your system manufacturer for more details.

  For systems also featuring Resilient System Technologies: No computer system can provide absolute reliability, availability or serviceability. Requires an Intel® Run Sure Technology-enabled system, including an enabled Intel processor and enabled technology(ies). Built-in reliability features available on select Intel® processors may require additional software, hardware, services and/or an Internet connection. Results may vary depending upon configuration. Consult your system manufacturer for more details.

  For systems also featuring Resilient Memory Technologies: No computer system can provide absolute reliability, availability or serviceability. Requires an Intel® Run Sure Technology-enabled system, including an enabled Intel® processor and enabled technology(ies). built-in reliability features available on select Intel® processors may require additional software, hardware, services and/or an Internet connection. Results may vary depending upon configuration. Consult your system manufacturer for more details.

Optimization Notice

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel.

Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

intel®