

TECHNCIAL COMPUTING / HPC SOFTWARE TOOLS & AI SOLUTIONS

Accelerate AI, HPC, Enterprise & Cloud Applications

April 2019 @ CiTIUS: Centro Singular de Investigación en Tecnoloxías da Información

Intel Computing Performance and Software Products (CPDP) Edmund Preiss



Intel Software Development Tools

• Intel optimized AI Solutions



2

Intel® Software Developer Tools & SDKs

Intel[®] Parallel Studio XE

Comprehensive Enterprise , HPC Tools suite

Shared and distributed memory systems

Code creation and versatile SW Analysis Tools



Intel[®] Media Server Studio Media Encode/Decode Tools

Media SDK Graphics Perf Analyzer Computer Vision SDK Open CL SDK Context SDK



Intel[®] System Studio



Comprehensive, all-in-one, cross-platform system & IoT development tool suite Simplifies system bring-up, boosts performance and power efficiency, strengthens system reliability

OpenVINO[™] Machine Learning / Deep Learning Inference

Embedded Tools Suite



Computer Vision SDK Deep Learning (DL) Deployment Toolkit Deep Learning Algorithms Optimized DL Frameworks

Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others. INTEL CONFIDENTIAL

What's Inside Intel[®] Parallel Studio XE

Comprehensive Software Development Tool Suite

COMPOSER EDITION	PROFESSIONAL EDITION	CLUSTER EDITION						
BUILD Compilers & Libraries	ANALYZE Analysis Tools	SCALE Cluster Tools						
C / C++ Compiler Optimizing Compiler Fortran Compiler Optimizing Compiler	Intel® VTune™ Amplifier Performance Profiler Intel® Inspector Memory & Thread Debugger	Intel® MPI Library Message Passing Interface Library Intel® Trace Analyzer & Collector MPI Tuning & Analysis						
Building Blocks Acceleration Library C++ Threading Library	Vectorization Optimization & Thread Prototyping	Cluster Diagnostic Expert System						
Intel [®] Distribution for Python* High Performance Scripting								
Intel® Architecture Platforms		(intel) CORE inside inside						
Operating System: Windows*, Linux*, MacOS ¹ *								

More Power for Your Code - <u>software.intel.com/intel-parallel-studio-xe</u>

Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

Code Modernization

Stage 1: Use Optimized Libraries

Stage 2: Compile with Architecture-specific Optimizations

Stage 3: Analysis and Tuning

Stage 4: Check Correctness

Optimization Notice Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.



INTEL® PARALLEL STUDIO XE TOOLS DETAILS

BUILD

Intel[®] C++ Compiler Intel[®] Fortran Compiler Intel[®] Distribution for Python* Intel[®] Math Kernel Library Intel[®] Integrated Performance Primitives Intel[®] Threading Building Blocks Intel[®] Data Analytics Acceleration Library Included in Composer Edition

ANALYZE

Intel® VTune™ Amplifier Intel® Advisor Intel® Inspector

Part of the Professional Edition

SCALE

Intel® MPI Library Intel® Trace Analyzer & Collector Intel® Cluster Checker

Part of the Cluster Edition

Fast, Scalable, Parallel Code with Intel[®] Compilers

Deliver Industry-leading C/C++ Code Performance, Unleash the Power of the latest Intel[®] Processors

- Develop optimized and vectorized code for Intel[®] architectures, including Intel Atom[®], Intel[®] Core[™] and Intel[®] Xeon[®] processors
- Achieve Superior Parallel Performance—Vectorize & thread your code (using OpenMP*) to take full advantage of the latest SIMD-enabled hardware, including Intel[®] Advanced Vector Extensions 512 (Intel[®] AVX-512)
- Leverage latest language and OpenMP* standards, and compatibility with leading compilers and IDEs

Learn More: software.intel.com/intel-compilers





What's New for Intel[®] C++ Compilers 19.0

Additional C++17 Standard feature support

- Enjoy improvements to lambda and constant expression support
- Improved GNU C++ and Microsoft C++ compiler compatibility

Standards-driven parallelization for C++ developers

- Partial OpenMP Version 5.0 support
- Modernize your code by using the latest parallelization specifications



Systems Performance Advantage as Measured by SPEC* on Intel[®] Xeon[™] Processors—Intel[®] C++ Compiler





Performance results are based on testing as of Aug. 26, 2018 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. 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. For more complete information, see Performance Benchmark Test Disclosure.

Testing by Intel as of Aug. 26, 2018. Configuration: Linux hardware: Intel® Xeon® Platinum 8180 CPU @ 2.50GHz, 384 GB RAM, HyperThreading is on. Software: Intel compilers 19.0, GCC 8.1.0. PGI 18.5, Clang/LLVM 6.0. Linux OS: Red Hat Enterprise Linux Server release 7.4 (Maipo), 3.10.0-693.el7.x86 64. SPEC* Benchmark (www.spec.org). SmartHeap 10 was used for CXX tests when measuring SPECint[®] benchmarks.SPECint[®] rate base 2017 compiler switches: SmartHeap 10 were used for C++ tests. Intel C/C++ compiler 19.0: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-mem-layout-trans=3. GCC 8.1.0 -march=znver1 -mfpmath=sse -Ofast -funroll-loops -flto. Clang 6.0: -march=core-avx2 -mfpmath=sse -Ofast funroll-loops -flto.SPECfp[®] rate base 2017 compiler switches: Intel C/C++ compiler 19.0: -xCORE-AVX512 -ipo -O3 -noprec-div -qopt-prefetch -ffinite-math-only -qopt-mem-layout-trans=3. GCC 8.1.0: -march=skylake-avx512 -mfpmath=sse -Ofast -fno-associative-math -funroll-loops -flto. Clang 6.0: -march=znver1 -mfpmath=sse -Ofast -funroll-loops flto.SPECint®_speed_base_2017 compiler switches: SmartHeap 10 were used for C++ tests. Intel C/C++ compiler 19.0: xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-mem-layout-trans=3 -qopenmp. GCC 8.1.0 '-march=znver1 -mfpmath=sse -Ofast -funroll-loops -flto -fopenmp. Clang 6.0: -march=core-avx2 -mfpmath=sse -Ofast -funroll-loops -flto fopenmp=libomp. SPECfp[®] speed base 2017 compiler switches: Intel C/C++ compiler 19.0: -xCORE-AVX512 -ipo -O3 no-prec-div -qopt-prefetch -ffinite-math-only -qopenmp. GCC 8.1.0: -march=skylake-avx512 -mfpmath=sse -Ofast -fnoassociative-math-funroll-loops-flto-fopenmp. Clang 6.0: -march=skylake-avx512 -mfpmath=sse-Ofast-funroll-loopsflto -fopenmp=libomp.

Testing by Intel as of Aug. 26, 2018. Configuration: Linux hardware: Intel® Xeon® Platinum 8180 CPU @ 2.50GHz, 384 GB RAM, HyperThreading is on. Software: Intel compilers 19.0, GCC 8.1.0, PGI 18.5, Clang/LLVM 6.0, Linux OS: Red Hat Enterprise Linux Server release 7.4 (Maipo), 3.10.0-693.el7.x86 64. SPEC* Benchmark (www.spec.org). SmartHeap 10 was used for CXX tests when measuring SPECint® benchmarks.SPECint® rate base 2017 compiler switches: SmartHeap 10 were used for C++ tests. Intel C/C++ compiler 19.0: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-mem-layout-trans=3. GCC 8.1.0 -march=znver1 -mfpmath=sse -Ofast -funroll-loops -flto. Clang 6.0: -march=core-avx2 -mfpmath=sse -Ofast funroll-loops -flto.SPECfp® rate base 2017 compiler switches: Intel C/C++ compiler 19.0: -xCORE-AVX512 -ipo -O3 -noprec-div -qopt-prefetch -ffinite-math-only -qopt-mem-layout-trans=3. GCC 8.1.0: -march=skylake-avx512 -mfpmath=sse -Ofast -fno-associative-math -funroll-loops -flto. Clang 6.0: -march=znver1 -mfpmath=sse -Ofast -funroll-loops flto.SPECint® speed base 2017 compiler switches: SmartHeap 10 were used for C++ tests. Intel C/C++ compiler 19.0: xCORE-AVX512 -ipo -O3 -no-prec-div -gopt-mem-layout-trans=3 -gopenmp, GCC 8.1.0 '-march=znver1 -mfpmath=sse -Ofast -funroll-loops -flto -fopenmp. Clang 6.0: -march=core-avx2 -mfpmath=sse -Ofast -funroll-loops -flto fopenmp=libomp. SPECfp® speed base 2017 compiler switches: Intel C/C++ compiler 19.0: -xCORE-AVX512 -ipo -O3 no-prec-div -qopt-prefetch -ffinite-math-only -qopenmp. GCC 8.1.0: -march=skylake-avx512 -mfpmath=sse -Ofast -fnoassociative-math-funroll-loops-flto-fopenmp. Clang 6.0: -march=skylake-ayx512 -mfpmath=sse -Ofast -funroll-loops flto -fopenmp=libomp.

Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others. 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. Microprocessors-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microprocessors. Certain optimizations not specific to 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.



Fast, Scalable Code with Intel[®] Math Kernel Library (Intel[®] MKL)

- Speeds computations for machine learning applications through highly-optimized, threaded & vectorized math functions
- Provides key functionality for dense & sparse linear algebra (BLAS, LAPACK, PARDISO), FFTs, vector math, summary statistics, splines & more
- Dispatches optimized code for each processor automatically without the need to branch code
- Optimized for single core vectorization & cache utilization
- Great performance with minimal effort!

INTEL[®] MATH KERNEL LIBRARY OFFERS...



Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.



10

What's Inside Intel[®] Math Kernel Library



Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others. ¹Available only in Intel[®] Parallel Studio Composer Edition.

What's New for Intel® Math Kernel Library 2019?

Just-In-Time Fast Small Matrix Multiplication

 Improved speed of S/DGEMM for Intel® Advanced Vector Extensions (Intel® AVX-512, Intel® AVX2) and with JIT capabilities

Sparse QR Solvers

 Solve sparse linear systems, sparse linear least squares problems, eigenvalue problems, rank and null-space determination, and others

Generate Random Numbers for Multinomial Experiments

 Highly-optimized multinomial random number generator for finance, geological and biological applications





FFT Performance Boost on Intel[®] Core[™] Processor Intel[®] Math Kernel Library

2D FFT Performance Boost using Intel[®] Math Kernel Library 2019 vs FFTW Single Precision 2D FFT on Intel[®] Core[™] i5-7600 CPU @ 3.50GHz



Transforms Size (Power of Two)

Performance results are based on testing as of July 12, 2018 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. 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 products. For more complete information, see <u>Performance Benchmark Test Disclosure</u>. Testing by Intel as of July 12, 2018. **Configuration**: Intel[®] Core[™] 15-7600, 1x4 cores, 3.50GHz, 6MB CPU Cache, 64GB RAM, OS RHEL 7.2

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. Microprocessors-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to 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.

Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.



Speed-up Analytics at the Edge with Intel[®] Data Analytics Acceleration Library (Intel[®] DAAL)

Boost Machine Learning & Data Analytics Performance

- Helps applications deliver better predictions faster
- Optimizes data ingestion & algorithmic compute together for highest performance
- Supports offline, streaming & distributed usage models to meet a range of application needs
- Split analytics workloads between edge devices and cloud to optimize overall application throughput

Learn More: software.intel.com/daal

What's New in the 2019 Release

New Algorithms

- High performance Logistic Regression, most widely-used classification algorithm
- Extended Gradient Boosting Functionality provides inexact split calculations & algorithm-level computation canceling by user-defined callback for greater flexibility
- User-defined Data Modification Procedure in CSV & IDBC data sources to implement a wide range of feature extraction & transformation techniques



Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

Speed Imaging, Vision, Signal, Security & Storage Apps with Intel[®] Integrated Performance Primitives (Intel[®] IPP)

Accelerate Image, Signal, Data Processing & Cryptography Computation Tasks

- Multi-core, multi-OS and multi-platform ready, computationally intensive & highly optimized functions
- Use high performance, easy-to-use, production-ready APIs to quickly improve application performance
- Reduce cost & time-to-market on software development & maintenance

What's New in 2019 Release

- Improved LZ4 compression & decompression performance on high entropy data
- New color conversion functions for convert RBG images to CIE Lab color models, & vice versa
- Open source distribution of Intel[®] IPP Cryptography Library
- Extended optimization for <u>Intel® Advanced Vector Extensions (Intel®-512 & Intel® AVX2</u>)s
- Added Threading Layer with OpenMP* and Threading Building Blocks support for various image processing functions

Learn More: software.intel.com/intel-ipp



What's Inside Intel[®] Integrated Performance Primitives

High Performance, Easy-to-Use & Production Ready APIs



Optimization Notice Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

Performance Improvement for Data Compression

Intel[®] Integrated Performance Primitives

Data Compression Performance Ratio, Intel[®] Integrated Performance Primitives 2019 vs LZ4, Zlib, LZO Libraries



Performance results are based on testing as of Aug. 15, 2018 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. 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. For more complete information, see <u>Performance Benchmark Test Disclosure</u>. Testing by Intel as of August 15, 2018. Configuration: Intel® Core[™] 15-7600 CPU @3.50GHz, 4 cores, hyper-threading off; Cache: L1=32KB, L2=256KB, L3=6MB; Memory: 64GB; OS: RH EL Server 7.2 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. Please refer to the applicable product User and Reference Guides for none information regarding the specific instruction sets covered by this notice. Notice revision #20110804.

Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

Get the Benefits of Advanced Threading with Threading Building Blocks

Use Threading Techniques to Leverage Multicore Performance & Heterogeneous Computing for C++

- Parallelize computationally intensive work across CPUs & GPUs—deliver higher-level & simpler solutions using C++
- Most feature-rich & comprehensive solution for parallel programming
- Highly portable, composable, affordable, approachable, future-proof scalability

Learn More: software.intel.com/intel-tbb



Optimization Notice Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.



What's Inside Threading Building Blocks



Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

Advantages of Using Threading Building Blocks over other Threading Models

- Specify tasks instead of manipulating threads. Threading Building Blocks (TBB) maps your logical tasks onto threads with full support for nested parallelism
- TBB uses proven, efficient parallel patterns.
- TBB uses work stealing to support the load balance of unknown execution time for tasks. This has the advantage of **low-overhead** <u>polymorphism</u>.
- Flow graph feature in TBB allows developers to easily express dependency and data flow graphs.
- Has high level parallel algorithms, concurrent containers, and low level building blocks like scalable memory allocator, locks and atomic operations.

Excellent Performance Scalability with Threading Building Blocks on Intel[®] Xeon[®] Processor



Performance results are based on testing as of July 31, 2018 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks. 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. For more complete information visit www.intel.com/benchmarks.Configuration: Testing by Intel as of July 31, 2018. Software versions: Intel® 64 Compiler, Version 18.0, Threading Building Blocks (TBB) 2019; Hardware: 2x Intel® Xeon® Gold 6152 CPU @ 2.10GHz, 192GB Main Memory; Operating System: CentOS Linux* release 7.4 1708 (Core), kernel 3.10.0-693.e17.x86 64; Note: sudoku, primes and tachyon are included with TBB.

Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others. 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 microprotecture 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. <u>Notice revision #20110804</u>. For more complete information about compiler optimizations, see our



INTEL® PARALLEL STUDIO XE Component tools

BUILD

Intel® C++ Compiler Intel® Fortran Compiler Intel® Distribution for Python* Intel® Math Kernel Library Intel® Integrated Performance Primitives Intel® Threading Building Blocks Intel® Data Analytics Acceleration Library Included in Composer Edition

ANALYZE

Intel® VTune™ Amplifier Intel® Advisor Intel® Inspector

Part of the Professional Edition

SCALE

Intel® MPI Library Intel® Trace Analyzer & Collector Intel® Cluster Checker

Part of the Cluster Edition

Before diving Into a particular tool ...

- How to assess that I have potential in performance tuning?
- Which tool should I use first?
- What to use on **large scale** avoiding being overwhelmed with huge trace size, post processing time and collection overhead?
- How to **quickly** evaluate environment settings or incremental code changes?
- Answer:

Use VTune Amplifier's Application Performance Snapshot



Performance Optimization Workflow based on APS





Better, Faster Application Performance Snapshot Intel® VTune™ Amplifier

Better Answers

CPU utilization analysis of physical cores

Less Overhead

- Lower MPI trace overhead & faster result processing
- New data selection & pause/resume let you focus on useful data

Easier to Use

- Visualize rank-to-rank & node-to-node MPI communications
- Easily configure profiling for Intel[®] Trace Analyzer & Collector



Analyze & Tune Application Performance & Scalability with Intel[®] VTune[™] Amplifier—Performance Profiler

Advanced Hotspots Hotspots + ③ INTEL VTUNE AMPLIFIER 2019								
Analysis Configuration Collecti	ion Log Summary	Bottom-up Call	er/Callee	Top-down	Tree Plat	form		
Grouping: Function / Call Stack								
	CPU Time 🔻 🧉				Context S	Context Switt ^		
Function / Call Stack	Effective Time by Idle Poor Ok	y Utilization 💿 Ideal 🚦 Over	Spin Time	Overhead Time	Wait Time	Inactive Time	Preemption	
 updateBusinessAccount 	7.915s		0s	0s	0s	0.055s	934	
main\$omp\$parallel_for@269	7.915s		0s	0s	0s	0.055s	934	
kmp_invoke_microtas	7.915s		0s	0s	0s	0.042s	815	
updateBusinessAccount	l Os		0s	0s	0s	0.013s	119	
updateCustomerAccount	7.766s		0s	0s	0s	0.052s	1,111	
kmpc_atomic_fixed8_add	2.772s		0s	0s				
kmpc_critical	Os		2.021s	0s	0s	0.014s	262 🗸	
< >	<						>	
p: 🕇 🗕 🖝 🖝	5s 5.2s	5.4s 5.6s	5.8s	6s 6.	2s	✓ Thread	~ ^	
G OMP Worker Thread #2 (TI	and date and the state			ا بالحد ا	· · · ·	Runr	ning	
COMP Worker Thread #3 (TI	Context Switches							
dentest enseme (TID: 12722)								
nintest_openinp (nD. 12732)	Synchronization 🖂 🖂 🖂 🖂 Synchronization 🖂 Synchronization							
OMP Worker Thread #1 (Ti								
CPU Time Spin and Overh								
< > CPU_CLK_UNH V								
FILTER 🕥 100.0% 🦹 Any Proc Y Any Thread Y Any Moc Y Any L Y User functi Y Show inl Y Functior Y								

Fast, Scalable Code, Faster

- Accurately profile C, C++, Java*, Python*, Go*, or any mix
- Optimize CPU/GPU, threading, memory, cache, storage & more
- Save time: rich analysis leads to insight

What's New in 2019 Release (Highlights)

- Simplified workflow for easier tuning
- I/O Analysis—Tune SPDK storage & DPDK network performance
- New Platform Profiler—Get insights into platformlevel performance, identify memory & storage bottlenecks & imbalances

Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

Intel[®] VTune[™] Amplifier—What's New in 2019 Details Performance Profiler

Easier to Use

- New workflow provides easier-to-learn tuning workflow and a simplified setup
- New visualization simplifies general exploration complex counter information

I/O Analysis—Tune SPDK storage and DPDK network performance

Measure "empty" polling cycles where no real work is done

Platform Profiler

Longer data collection finds hardware configuration issues and poorly tuned applications

Tune CPU/GPU Rendering—GPU Rendering Analysis (Linux*, Android* only)

- Detects performance-critical API calls of OpenGL-ES applications
- Finds virtual Xen* hypervisor domains that bottleneck the system

Balance CPU Loading—CPU Analysis

- Balance CPU loading for better performance
- Assess memory transfer time, CPU balance, CPU context switches, FPU utilization, workload wait times, and more



Tune Workloads & System Configuration

Intel[®] VTune Amplifier

Finds

- Configuration issues
- Poorly tuned software

Target Users

- Infrastructure Architects
- Software Architects & QA



- Extended capture (minutes to hours)
- Low overhead coarse grain metrics
- Sampling OS & hardware performance counters
- RESTful API for easy analysis by scripts



Timelines & Histograms



Core to Core Comparisons



Memory Ops Per Instruction (average/core)



Loads Stores



Optimization Notice Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

Optimize Vectorization & Threading with Intel[®] Advisor

Performance Increases Scale with Each New Hardware Generation



Modern Performant Code

- Vectorized for Intel[®] Advanced Vector Extensions (Intel[®] AVX-512 & Intel[®] AVX)
- Efficient memory access
- Threaded

Capabilities

- Adds & optimizes vectorization
- Analyzes memory patterns
- Quickly prototypes threading

Benchmark: Binomial Options Pricing Model software.intel.com/en-us/articles/binomial-options-pricing-model-code-for-intel-xeon-phi-coprocessor

Performance results are based on testing as of August 2017 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks. See Vectorize & Thread or Performance Dies Configurations for 2010-2017 Benchmarks in Backup.

Learn More: http: intel.ly/advisor

Testing by Intel as of August 2017.

Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others. 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, & SSSE3 instruction sets & 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



Get Breakthrough Vectorization Performance

Intel[®] Advisor—Vectorization Advisor

Faster Vectorization Optimization

- Vectorize where it will pay off most
- Quickly ID what is blocking vectorization
- Tips for effective vectorization
- Safely force compiler vectorization
- Optimize memory stride

Data & Guidance You Need

- Compiler diagnostics + Performance Data + SIMD efficiency
- Detect problems & recommend fixes
- Loop-Carried Dependency Analysis
- Memory Access Patterns Analysis

Ó	Elapsed time: 70.29s 🧩 🗿 Vectorized 🖉 Not Vectorized							C	FF	Sma	rt Mode	•		٩,	
FILT	FILTER: All Modules 🔻 All Sources 👻 Loops And Functions 👻 All Threads 👻									019					
P	P Summary Survey & Roofline 🎓 Refinement Reports														
8	E E Europies Cell Characteries		Vector	Call Time	Total	T	FLOPS	\gg	Why No	Vectorize	ed Loops		\gg	Trip	> ^
OFI	+ - Function Call Sites and Loops		Issues	Self Time*	Time	туре	GFLOPS	AI	Vectorization?	Vector	Efficiency	Gain	VL	Counts	
INE	+ ⁽⁵ [loop in S252 at loops90.f:1172]	-	💡 1 Possible	. 3.129s 7.0%	3.129	Vectorized	0.191	0.115	🖬 1 vectorizat	AVX2	17%	1.36x	4; 8	99; 6; 1; 1	
	[loop in S2101 at loops90.f:1749]	-	2 Possible	. 2.765s 6.2%	2.765s	Scalar	0.1421	0.067	vectorizatio					12	
	🛨 🖱 [loop in s442_\$omp\$parallel_for		💡 1 Ineffecti	. 1.492s 3.4%	1.492s	Vectorized+	0.5861	0.165		AVX2	14%	1.09x	8	30; 1; 3	
				1.108s 2.5%	1.108s	Vector Funct	3.9111	0.156		AVX2					
	⊕ [loop in \$353 at loops90.f:2381]		9 1 Possible	. 0.989s 2.2%	0.989s	Vectorized (2.0231	0.134		AVX2	27%	2.16x	8	6; 4; 1	~
	< >	<													>

Optimize for Intel® Advanced Vector Extensions 512 (Intel® AVX-512) with or without access to Intel AVX-512 hardware

Optimization Notice Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

Find Effective Optimization Strategies

Intel® Advisor—Cache-aware Roofline Analysis

Roofline Performance Insights

- Highlights poor performing loops
- Shows performance 'headroom' for each loop
 - Which can be improved
 - Which are worth improving
- Shows likely causes of bottlenecks
- Suggests next optimization steps

Nicolas Alferez, Software Architect Onera – The French Aerospace Lab



"I am enthusiastic about the new "integrated roofline" in Intel® Advisor. It is now possible to proceed with a step-bystep approach with the difficult question of memory transfers optimization & vectorization which is of major importance."

Optimization Notice Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

Visualize Parallelism

Intel® Advisor—Flow Graph Analyzer (FGA)

- Interactively build, validate, and analyze algorithms using the Flow Graph Analyzer module
 - Visually generate code stubs, use Threading Building Blocks (TBB) Flow Graph to get started with generating parallel C++ programs
 - Click and zoom through your algorithm's nodes and edges to understand parallel data and program flow
 - Use FGA dashboard to analyze your algorithm's load balancing, concurrency, and other parallel attributes to fine tune your program



Enjoy full support for TBB Flow Graph; initial support for OpenMP* 5 (draft) OMPT API



Debug Memory & Threading with Intel[®] Inspector Find & Debug Memory Leaks, Corruption, Data Races, Deadlocks

D								
2								
tate								
New								
Confirmed								
8								
ing_errors.ex								
ing_errors.ex								
6								
Write winvideo.h:270 next_frame find_and_fix_threading_errors.exe g_updates								
ing_errors.ex								
ing_errors.ex								
ing_errors.ex								
find_and_fix_threading_errors.ex								
ing_errors.ex								

New Resource: How to Use Intel® Inspector—Persistence Inspector

Learn More: intel.ly/inspector-xe

Correctness Tools Increase ROI by 12%-21%¹

- Errors found earlier are less expensive to fix
- Races & deadlocks not easily reproduced
- Memory errors are hard to find without a tool

Debugger Integration Speeds Diagnosis

- Breakpoint set just before the problem
- Examine variables and threads with the debugger

What's New in 2019 Release Find Persistent Memory Errors

- Missing / redundant cache flushes
- Missing store fences
- Out-of-order persistent memory stores
- PMDK* transaction redo logging errors

¹Cost Factors – Square Project Analysis – CERT: U.S. Computer Emergency Readiness Team, and Carnegie Mellon CyLab NIST: National Institute of Standards & Technology: Sauare Project Results

Race Conditions are Difficult to Diagnose

They only Occur Occasionally & are Difficult to Reproduce—Intel® Inspector

Correct Answer

Thread 1	Thread 2		Shared Counter
			0
Read count		÷	0
Increment			0
Write count		→	1
	Read count	÷	1
	Increment		1
	Write count	→	2

Incorrect Answer

Thread 1	Thread 2	Shared Counter		
			0	
Read count		÷	0	
	Read count	÷	0	
Increment			0	
	Increment		0	
Write count		>	1	
	Write count	→	1	



Boost Distributed Application Performance with Intel[®] MPI Library Performance, Scalability & Fabric Flexibility

Standards Based Optimized MPI Library for Distributed Computing

- Built on open source MPICH Implementation
- Tuned for low latency, high bandwidth & scalability
- Multi-fabric support for flexibility in deployment

What's New in 2019 Release

- New MPI code base- MPI-CH4 (on the path to Exascale & beyond)
- Greater scalability & shortened CPU paths
- Superior MPI Multi-threaded performance
- Supports the latest Intel[®] Xeon[®] Scalable processor





INTEL® PARALLEL STUDIO XE Component tools

BUILD

Intel® C++ Compiler Intel® Fortran Compiler Intel® Distribution for Python* Intel® Math Kernel Library Intel® Integrated Performance Primitives Intel® Threading Building Blocks Intel® Data Analytics Acceleration Library Included in Composer Edition

ANALYZE

Intel® VTune™ Amplifier Intel® Advisor Intel® Inspector

Part of the Professional Edition

SCALE

Intel® MPI Library Intel® Trace Analyzer & Collector Intel® Cluster Checker

Part of the Cluster Edition
Intel[®] MPI Library Features

Optimized MPI Application Performance

- Application-specific tuning
- Automatic tuning
- Support for Intel[®] Omni-Path Architecture Fabric

Multi-vendor Interoperability & Lower Latency

- Industry leading latency
- Performance optimized support for the fabric capabilities through OpenFabrics* (OFI)

Faster MPI Communication

Optimized collectives

Sustainable Scalability

 Native InfiniBand* interface support allows for lower latencies, higher bandwidth, and reduced memory requirements

More Robust MPI Applications

Seamless interoperability with Intel[®] Trace Analyzer & Collector



Intel[®] MPI Library = 1 library to develop, maintain & test for multiple fabrics



Superior MPI Performance with Intel® MPI Library on Linux* 64

1,280 Processes, 32 Xeon nodes (Intel® Omni-Path) Linux* 64

Relative (Geomean) MPI Latency Benchmarks (Higher is Better)



Performance results are based on testing as of Sept. 5, 2018 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. 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. For more complete information, see <u>Performance Benchmark Test Disclosure</u>.

Configuration: Testing by Intel as of Sept. 5, 2018.Hardware: Intel[®] Xeon[®]Gold 6148 CPU @ 2.40GHz; 192 GB RAM. Interconnect: Intel[®] Omni-Path Host Fabric InterfaceSoftware: RHEL* 7.4; IFS 10.7.0.0.145; Libfabric internal; Intel[®] MPI Library 2019; Intel[®] MPI Benchmarks 2019 (built with Intel[®] C++ Compiler XE 18.0.2.199 for Linux*);

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. Microprocessors. Certain optimizations not specific to Intel microprocessors. Certain optimizations not specific to Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. <u>Notice revision #20110804</u>. For more complete information about compiler optimizations, see our <u>Optimization Notice</u>.

Optimization Notice

Profile & Analyze High Performance MPI Applications Intel® Trace Analyzer & Collector

Powerful Profiler, Analysis & Visualization Tool for MPI Applications

- Low overhead for accurate profiling, analysis & correctness checking
- Easily visualize process interactions, hotspots & load balancing for tuning & optimization
- Workflow flexibility: Compile, Link or Run

What's New in 2019 Release

- Minor updates & enhancements
- Supports the latest Intel[®] Xeon[®] Scalable processors

Learn More: software.intel.com/intel-trace-analyzer





Efficiently Profile MPI Applications Intel® Trace Analyzer & Collector

Helps Developers

- Visualize & understand parallel application behavior
- Evaluate profiling statistics & load balancing
- Identify communication hotspots

Features

- Event-based approach
- Low overhead
- Excellent scalability
- Powerful aggregation & filtering functions
- Idealizer
- Scalable

Optimization Notice	
Copyright © 2018, Intel Corp	ooration. All rights reserved.
*Other names and brands ma	y be claimed as the property of others.

Eile	Options Project	Windows	Help								- 8
view C	narts <u>N</u> avigate	Advanced	Layour	Currents and "		dan 🔛 14			V 382		a.
i L	· · · · · · · · · · · · · · · · · · ·	34 - 69.372 29	2:0.190 550	Seconds +		Des 🗾 m	a)or Function Groups	² 🐖	1 2		: 🗙
	69.18 8	69.20 s	69.22 5	69.24 5	6a.4	6 .	69.28 s	.20 .	69.32 3	69.34 5	69.36 s
i node 1	Application	AppliderApp	dication		ApplidArA	oplication		Applid	ArApplication		ApplidAtApplication
						opined out t		17	e deference en		
i node 1-mir	c0 MPE	AMPI Appli	cation A	VP:		plication	MPI	AMPI	Application	MPI	AMPI Application
node 1-mic	c1/MPI	IA MPI JAppli	cation 4	MPI	AMPI 4API	plication	MPI		Application	MPI	AMPI (Application
								\mathbf{N}			
node2	Application	ApplicAcApp	plication		Applicação	pplication		Applid	AcApplication		ApplidArApplication
					17			17			
node2-mic	c0 <mark>wMPI</mark>	AMPI Appli	cation a	MPT		plication	MPI	AMPIA	Application	MPi	April Pl Application
i node2-mic	c1 MPI	ArMP14Appli	cation	MPI	AMPI (Ap)	plication	4MPI	AMPI	Application	MPI	AMPI Application
			_			_					
i node3	Application	ApplidAcApp	dication		NApplidAcA	pplication		Applid	AcApplication		ApplidAjApplicatio
								- /			
node3-mic	20 MIP1	AMPLApp	cation	MP1	AMP14AD	plication	MP1	AMPL	Application	MPI	AMPI Application
-ada2 mi						- Conting			tenlection		A HDT (Application
nodeamin	:10001	AL ME LOOKAN	cauon	MP1		plication	MP1		Application	mP1	AMP1 Application
node4	Application	(Applicar.Apr	dication		ApplicarA	onlication		Month	ArApplication		ApplicArApplicatio
These -			arcu dan	_				17	di di finano a con		
node4-mir	CO AMPL	AMPLAppl	cation 1	MPI		plication	MP	AMPI	Application	MPI	AMPE Application
		A MOTION OF		MDT	AMPT (Apr	olication	MPT	AMPT	Application	MPT	AVIPI (Application



40

Use an Extensive Diagnostic Toolset for High Performance Compute Clusters—Intel[®] Cluster Checker (for Linux*)

Ensure Cluster Systems Health

- Expert system approach providing cluster systems expertise verifies system health: find issues, offers suggested actions
- Provides extensible framework, API for integrated support
- Check 100+ characteristics that may affect operation & performance improve uptime & productivity

New in 2019 Release: Output & Features Improve Usability & Capabilities

- Simplified execution with a single command
- New output format with overall summary
 - Simplified issue assessment for 'CRITICAL', 'WARNING', or 'INFORMATION'
 - Extended output to logfile with details on issue, diagnoses, observations
- Added auto-node discovery when using Slurm*
- Cluster State 2 snapshot comparison identifies changes
- And more...



For application developers, cluster architects & users, & system administrators

Functionality, Uniformity, & Performance Tests

Intel[®] Cluster Checker

Comprehensive pre-packed cluster systems expertise out-of-the-box

- ✓ Suitable for HPC experts & those new to HPC
- Tests can be executed in selected groups on any subset of nodes



System Qualification with Intel® Cluster Checker

3 Phases of Use from Standing Up to Ongoing Cluster Operations

Installation	Operational	Performance
Qualification	Qualification	Qualification
Evaluate	Evaluate	Evaluate
Verify correct installation	Verify correct operation	Verify performance
Ensure	Ensure	Ensure
Correct installation to specs	Correct operation to specs	Correct performance to specs
Establish Baseline for cluster	Verify Cluster meets specs & system compliance	Verify Cluster meets customers performance targets

Intel Cluster Checker



Developer: Call to Action/More Resources

Download Intel® System Studio

- Free Community License
- Free/Discounted Versions for <u>Students</u> /<u>Academia</u>

Access Developer Resources

- Product site
- <u>Get Started/Documentation</u>
- Code Samples/Training
- Tech.Decoded webinars, how-to videos/articles
- Expert Community Support Forum



Optimization Notice

Paid Licenses include Priority Support





• Intel Software Development Tools

Intel optimized AI Solutions



ARTIFICIAL Intelligence

is the ability of machines to learn from experience, without explicit programming, in order to perform cognitive functions associated with the human mind

ARTIFICIAL INTELLIGENCE

MACHINE LEARNING

Algorithms whose performance improve as they are exposed to more data over time

DEEP LEARNING

Subset of machine learning in which multi-layered neural networks learn from vast amounts of data



MACHINE VS. DEEP LEARNING







DEEP LEARNING BREAKTHROUGHS

Machines able to meet or exceed human image & speech recognition



Source: ILSVRC ImageNet winning entry classification error rate each year 2010-2016 (Left), https://www.microsoft.com/en-us/research/blog/microsoft-researchers-achieve-new-conversational-speech-recognition-milestone/ (Right)



		ARTIFICIAL INTELLIGENCE						
	Solution Architects	Al Solutions Catalog (Public & Internal) Platforms Finance Healthcare Energy Industrial Transport Retail Home More						
	τοοι μπο	DEEP LEARNING DEPLOYMENT DEEP LEARNIN						
KN	I U U U U U U U U U U U U U U U U U U U	<u>OpenVINO[™] †</u> <u>Intel[®] Movidius[™] SDK</u> Intel [®] Deep						
II	App 👌 Developers 🍑	Open Visual Inference & Neural Network Optimization toolkit for inference deployment on CPU, processor graphics, FPGA & VPU using TF, Caffe* & MXNet* Optimized inference deployment for all Intel® Movidius™ VPUs using TensorFlow* & Caffe* Learning Studio*						
112								
FI	I IRRARIES	Python R Distributed Now optimized for CPU Optimizations in progress						
11		• Scikit-learn • Cart • MILib (on Spark)						
11		• <u>Pandas</u> • <u>Random</u> • <u>Mahout</u> • NumPy Forest						
C I		• <u>e1071</u> <u>TensorFlow</u> * <u>MXNet</u> * <u>Caffe</u> * <u>BigDL/Spark</u> * Caffe2* PyTorch* PaddlePaddle*						
U I		ANALYTICS, MACHINE & DEEP LEARNING PRIMITIVES DEEP LEARNING GRAPH COMPILER						
IG	TUUNDATIUN	<u>Python DAAL MKL-DNN clDNN Intel®nGraph™ Compiler</u> (Alpha)						
ΛC	Library 👩	Intel distribution Intel® Data Analytics Open-source deep neural Open-sourced compiler for deep learning model						
AL	Developers 🌄	optimized for Acceleration Library network functions for computations optimized for multiple devices (CPU, GPU, machine learning (for machine learning) CPU, processor graphics NNP) using multiple frameworks (TF, MXNet, ONNX)						
LN								
Ĉ	HARNWARF	AI FOUNDATION DEEP LEARNING ACCELERATORS						
Ŀ								
E E	IT System	Luge Luge Strattx to inside						
[†] Formerly the Intel [®] Computer	Vision SDK	NNP L-1000 Inference						
All products, computer systems	be claimed as the property of others. , dates, and figures are preliminary bas	ed on current expectations, and are subject to change without notice.						
Optimization Copyright © 2018,	Notice Intel Corporation. All rights reserved.	AI.INTEL.COM						



INTEL® MATH KERNEL LIBRARY

Optimization Notice Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.



Components Comparison : Intel MKL-DNN vs Intel MKL

MKL-DNN (Open Source)

- Convolution
- Pooling

- ReLU
- Inner Product

Normalization

MKL (Math Kernel Library)								
Linear Algebra	Fast Fourier Transforms	Vector Math	Summary Statistics	And More				
 BLAS LAPACK ScaLAPACK Sparse BLAS Sparse Solvers 	MultidimensionalFFTW interfacesCluster FFT	 Trigonometric Hyperbolic Exponential Log Power 	 Kurtosis Variation coefficient Order statistics 	 Splines Interpolation Trust Region Fast Poisson Solver 				
 Iterative PARDISO* Cluster Sparse Solver 		 Root Vector RNGs	 Min/max Variance- covariance 					

Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.



INTEL[®] DATA ANALYTICS ACCELERATION LIBRARY

Optimization Notice Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.



Speed-up Machine Learning and Analytics with Intel[®] Data Analytics Acceleration Library (Intel[®] DAAL)

Boost Machine Learning & Data Analytics Performance

- Helps applications deliver better predictions faster
- Optimizes data ingestion & algorithmic compute together for highest performance
- Supports offline, streaming & distributed usage models to meet a range of application needs
- Split analytics workloads between edge devices and cloud to optimize overall application throughput

Learn More: software.intel.com/daal

What's New in the 2019 Release

New Algorithms

- High performance Logistic Regression, most widely-used classification algorithm
- Extended Gradient Boosting Functionality provides inexact split calculations & algorithm-level computation canceling by user-defined callback for greater flexibility
- User-defined Data Modification Procedure in CSV & IDBC data sources to implement a wide range of feature extraction & transformation techniques



Optimization Notice

Processing Modes

Batch Processing



 $\mathsf{R} = \mathsf{F}(\mathsf{D}_1, \dots, \mathsf{D}_k)$



Online

Processing

 $S_{i+1} = T(S_i, D_i)$ $R_{i+1} = F(S_{i+1})$

Distributed Processing





Data Transformation & Analysis Algorithms

Intel[®] Data Analytics Acceleration Library



Algorithms supporting batch processing

Algorithms supporting batch, online and/or distributed processing



Machine Learning Algorithms

Intel® Data Analytics Acceleration Library



Algorithms supporting batch, online and/or distributed processing

Optimization Notice

INTEL DISTRIBUTION OF PYTHON

Optimization Notice Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.





Chapter 19: Performance Optimization of **Black—Scholes** Pricing

$$\begin{split} & V_{\text{call}} = S_0 \cdot \text{CDF}\left(d_1\right) - e^{-rT} \cdot X \cdot \text{CDF}\left(d_2\right) \\ & V_{\text{pat}} = e^{-rT} \cdot X \cdot \text{CDF}\left(-d_2\right) - S_0 \cdot \text{CDF}\left(-d_1\right) \end{split}$$



Performance gap between C and Python





The most popular ML package for Python – Intel Optimized





scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- · Accessible to everybody, and reusable in various contexts
- · Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.

Algorithms: SVM, nearest neighbors,

random forest, ...

- Examples

Regression

...

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices. Algorithms: SVR, ridge regression, Lasso,

Examples

Clustering

Automatic grouping of similar objects into sets.

 Applications: Customer segmentation,

 Grouping experiment outcomes

 Algorithms: k-Means, spectral clustering,

 mean-shift, ...

 — Examples

Optimization Notice



Optimizing scikit-learn with Intel® DAAL

scikit-learn

Intel[®] DAAL

Optimized kernels from Intel® MKL

- The most popular package for machine learning
- Hundreds of algorithms with different parameters
- Has a very flexible and easy-to-use interface

High performance of analytical and machine learning algorithms on Intel architecture

High performance basic mathematical routines (BLAS, vector math, RNG, ...)



Faster Python* with Intel® Distribution for Python

Intel® Distribution for Python* Performance Speedups for Select Math Functions on Intel® Xeon™ Processors



Configuration: Hardware: Intel[®] Xeon[®] CPU E5-5699 v4@ 22:064:12 20 cores per socket, 1 thread per core – HT is 6tfl, 2566 2001 A@ 2400 HHz. Software Stock CentOS Linux release 73.1611 (Core), prihon 3.62, pip 90.1, nump 1.13.1, scipiv 1.91.1, scikit-learn 0.190. Intel[®] Distribution for Python* 2018 Gold: mkl 2018.0.0 intel₄, daal 2018.0.0.2017 084 Junny 1.13.1 py36, Intel₁ 5, operm p201800. Intel₁, scikit-learn 0.190, finte[®] 1, scikit-learn 0.190, Intel[®] Distribution for Python* 2018 Gold: mkl 2018.0.0 intel₄, daal

Software and workloads used in performance tests may have been optimized for performance only on intel microgrocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and inclusion. Any change to any of those factors may cause the results to vary. You should coust done information and performance tests to assist your in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to <u>https://www.intel.comberformance</u>. Benchmark Source: Intel Corporation.

Combinitions before intel's complies may or may no optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SE2, 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 line microprocessors. The optimizations are specific bined microprocessors and manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to line microprocessors. The optimizations are specific bined in the microprocessors. Certain optimizations not specific bine microprocessors are an availability of the optimizations on specific bine microprocessors and manufactured by Intel. Microprocessor-dependent optimizations are therefore the optimization of the microprocessors. The optimization optimization on the microprocessors are applicable product. User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804. "Other brands and names are the property of their respective owners.

Learn More: software.intel.com/distribution-for-python

High Performance Python Distribution

- Accelerated NumPy, SciPy, scikit-learn well suited for scientific computing, machine learning & data analytics
- Drop-in replacement for existing Python. No code changes required
- Highly optimized for latest Intel processors
- Take advantage of <u>Priority Support</u> connect direct to Intel engineers for technical questions²

What's New in 2019 version

- Faster Machine learning with Scikit-learn functions
 - Support Vector Machine (SVM) and K-means prediction, accelerated with Intel[®] DAAL
- Built-in access to XGBoost library for Machine Learning
- Access to Distributed Gradient Boosting algorithms
- Ease of access installation
- Now integrated into Intel[®] Parallel Studio XE installer.

Software & workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark & MobileMark, are measured using specific computer systems, components, software, operations & functions. Any change to any of those factors may cause the results to vary. You should consult other information & performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to https://www.intel.com/performance.

Optimization Notice ²Paid versions only. Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others. 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 SSE3 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 informatio regarding the specific institution sets covered by this notice.



Installing Intel[®] Distribution for Python* 2018

Standalone Installer	Download full installer from https://software.intel.com/en-us/intel-distribution-for-python	
		2.7 & 3.6
Anaconda.org Anaconda.org/intel channel	<pre>> conda configadd channels intel > conda install intelpython3_full > conda install intelpython3_core</pre>	(3.7 coming soon)
РуРІ	<pre>> pip install intel-numpy > pip install intel-scipy + Intel library Runtime packages > pip install mkl_fft + Intel development packages > pip install mkl_random</pre>	Linux* Windows*
Docker Hub	docker pull intelpython/intelpython3_full	
		OS X*
YUM/APT	Access for yum/apt: https://software.intel.com/en-us/articles/installing-intel-free- libs-and-python	

Optimization Notice Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

intel

Tune Python* + Native Code for Better Performance

Analyze Performance with Intel[®] VTune[™] Amplifier (available in Intel[®] Parallel Studio XE)

Challenge

- Single tool that profiles Python + native mixed code applications
- Detection of inefficient runtime execution

Solution

- Auto-detect mixed Python/C/C++ code & extensions
- Accurately identify performance hotspots at line-level
- Low overhead, attach/detach to running application
- Focus your tuning efforts for most impact on performance

Available in Intel® VTune™ Amplifier & Intel® Parallel Studio XE

Optimization Notice Copyright © 2018, Intel Corporation. All rights reserved

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

Basic Hotspots Hotspots by C	PU Usage viewpoint (<u>change</u>) ®	Col	ler/Ca	llee 🔹 Ton-down Tree	INTEL VTUNE AMPLIFIER XE 2017	
ouping: Function / Call Stack		Curr		- 5. Q	CPU Time -	
	CPU Time *		«		Viewing ↓ 1 of 1 ▷ selected stack(s)	
Function / Call Stack	Effective Time by Utilization	Spin Time	Ove. Tim.	Module	100.0% (10.809s of 10.809s) demo.pyl <u>process_slow</u> - demo.py run.pyl <u>slow_encode</u> +0x11 - run.py.4	
process_slow	10.403s	0.406s	0s	demo.py	run.pyl <u><module></module></u> +0x57 - run.py:13 python.exe! <u>tmainCRTStartup</u> +0x119 - crte.	
⊕process_fast	4.097s	0.190s	0s	demo.py	KERNEL32.DLLI <u>BaseThreadInitThunk</u> +0x2.	
■fast_encode	0.031s	Os	0s	run.py	ntali.ali <u>!Atioseri nreadstart</u> +0x33 - junkno	
®get_data	0.016s	0.010s	0s	<frozen importlibboo<="" td=""><td></td></frozen>		
■_tmainCRTStartup	0.016s	Os	0s	python.exe		
■ <module></module>	0.009s	Os	0s	run.py		
E_call_with_frames_removed	0.006s	0s	0s	<frozen importlibboo<="" th=""><th></th></frozen>		
Selected 1 row(s):	10.403s	0.406s	0s			
Q®Q• 1s 2s 3s wmainCRTStartu 1f 1f	4s 5s 6s 7s 8s 9	s 10)s	11 <u>5 125 135 1</u>	Ids 15s Ø Thread Ø Running Ø àt CPU Time Ø àt CPU Time Ø CPU Usage Ø àt CPU Time Ø àt Spin and .	

Auto detection & performance analysis of Python & native functions

63

INTEL OPTIMIZED FRAMEWORKS

Optimization Notice Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

Intel Confidential under CNDA



64









A



Optimization Notice



INTEL AI (DISTRIBUTED) FRAMEWORKS

FRAMEWORKS OPTIMIZED BY INTEL



ai.intel.com/framework-optimizations/







SEE ALSO: Machine Learning Libraries for Python (Scikit-learn, Pandas, NumPy), R (Cart, randomForest, e1071), Distributed (MlLib on Spark, Mahout) *Limited availabiling todayon NOLICE Other names, and brands may be claimed as the property of others.



Deep Learning Software Stack for Intel processors



Deep learning and AI ecosystem includes edge and datacenter applications.

- Open source frameworks (Tensorflow*, MXNet*, CNTK*, PaddlePaddle*)
- Intel deep learning products (Neon[™] framework, BigDL, OpenVINO[™] toolkit)
- In-house user applications

Intel MKL and Intel MKL-DNN optimize deep learning applications for Intel processors :

- through the collaboration with framework maintainers to upstream changes (Tensorflow*, MXNet*, PaddlePaddle*, CNTK*)
- through Intel optimized forks (Caffe*, Torch*, Theano*)
- by partnering to enable proprietary solutions

Intel MKL-DNN is an open source performance library for deep learning applications (available at https://github.com/intel/mkl-dnn)

- Fast open source implementations for wide range of DNN functions
- · Early access to new and experimental functionality
- Open for community contributions

Intel MKL is a proprietary performance library for wide range of math and science applications

Distribution: Intel Registration Center, package repositories (apt, yum, conda, pip)



Optimization Notice

INTEL® XEON® PROCESSOR PLATFORM PERFORMANCE

Hardware plus optimized software

INFERENCE THROUGHPUT

TRAINING THROUGHPUT



Intel® Xeon® Platinum 8180 Processor higher Intel optimized Caffe GoogleNet v1 with Intel® MKL inference throughput compared to Intel® Xeon® Processor E5-2699 v3 with BVLC-Caffe

Inference and training throughput uses FP32 instructions

Intel® Xeon® Platinum 8180 Processor higher Intel Optimized Caffe AlexNet with Intel® MKL training throughput compared to Intel® Xeon® Processor E5-2699 v3 with BVLC-Caffe



Deliver significant AI performance with hardware and software optimizations on Intel® Xeon® Scalable Family

Up to 191X Intel[®] Xeon[®] Platinum 8180 Processor higher Intel optimized Caffe Resnet50 with Intel[®] MKL inference throughput compared to Intel[®] Xeon[®] Processor E5-2699 v3 with BVLC-Caffe Up to 93X Intel[®] Xeon[®] Platinum 8180 Processor Higher Intel optimized Caffe Resnet50 with Intel[®] MKL training throughput compared to Intel[®] Xeon[®] Processor E5-2699 v3 with BVLC-Caffe

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 charge 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 information and performance tests to assist you in fully evaluating source: Intel measured as of June 2017. Configurationas See the last slide in this presentation, and brands may be claimed as the property of others.





Components of Apache Spark

An Integrated Framework for Advanced Analytics

Spark SQL

- Seamlessly mix SQL queries
- Unified Data Access (load/query)
- Schema RDD to support structured/ semi structured data

Spark Streaming

- Data continuously streamed in & processed in near real-time
- Support in-memory computation in a fault tolerant manner

GraphX (Graph)

 Distributed graph processing framework for advanced machine learning & data mining

MLlib

(Machine Learning)

- Distributed machine learning framework
- Library of distributed machine learning & statistical algorithms

Apache Spark*/Spark Core API

Provides distributed task scheduling, dispatching, memory management, fault recovery, and basic I/O functionalities with a fundamental programming abstraction, RDD for in-memory computation in a fault tolerant manner

R	SQL	Python	Scala	Java
---	-----	--------	-------	------

Runs on and access diverse data sources







Optimization Not

Copyright © 2018, Intel

*Other names and brands may be claimed as the property of others.

*Other names and brands may be claimed as the property of others

Source: http://spark.apache.org/



INTEL BIGDL - DISTRIBUTED DEEP LEARNING LIBRARY Performance Learning for Apache



BigDL is an **open-source** distributed deep learning library for Apache Spark* that can run directly on top of existing Spark or Apache Hadoop* clusters

Ideal for DL Models TRAINING and INFERENCE

Designed and Optimized for Intel® Xeon®

No need to deploy costly accelerators, duplicate data, or suffer through scaling headaches!





Feature Parity with TensorFlow*, Caffe* and Torch*

Lower TCO and improved ease of use with existing infrastructure



Deep Learning on Big Data Platform, Enabling Efficient Scale-Out

Powered by Intel[®] MKL and multi-threaded programming

Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

software.intel.com/bigdl



TensorFlow with Intel MKL/MKL-DNN

Use Intel Distribution for Python*

- Uses Intel MKL for many NumPy operations thus supports MKL_VERBOSE=1
- Available via <u>Conda</u>, or <u>YUM</u> and <u>APT</u> package managers

<u>Use pre-built Tensorflow* wheels</u> or build TensorFlow* with `bazel build -- config=mkl`

- Building from source required for integration with Intel Vtune[™] Amplifier
- Follow the <u>CPU optimization</u> advices including setting affinity and # of intra- and inter- ops threads
- More Intel MKL-DNN-related optimizations are slated for the next version: Use the latest TensorFlow* master if possible



Distributed TensorFlow[™] Compare



The parameter server model for distributed training jobs can be configured with different ratios of parameter servers to workers, each with different performance profiles.



The ring all-reduce algorithm allows worker nodes to average gradients and disperse them to all nodes without the need for a parameter server.

Source: https://eng.uber.com/horovod/

íntel

Intel and the Intel logo are trademarks of Intel Corporation in the U. S. and/or other countries. *Other names and brands may be claimed as the property of others. Copyright © 2018, Intel Corporation.

Optimization Notice Copyright © 2018, Intel Corporation. All rights reserved.

*Other names and brands may be claimed as the property of others.

72
Scaling TensorFlow

There is way more to consider when striking for peak performance on distributed deep learning training.:

https://ai.intel.com/white-papers/best-known-methods-forscaling-deep-learning-with-tensorflow-on-intel-xeonprocessor-based-clusters/

WHILE PAPER IP roducts Group, Intel Corporation ustomer Solutions Technical Enabling Best Practices for Scaling Deep Learning Training and Inference with TensorFlow* On Intel® Xeon® Processor-Based HPC Infrastructures Bub Contents Best Practices for TensorFlow* On Intel® Xeon® Processor-Based HPC Infrastructures Status Internet Inter	WHILE PAPER At Products Group, Intel Corporation Customer Solutions Technical Enabling Besst Practices for Scaling Deep Learning Traini and Inference with TensorFlow* On Intel® Xeo Processor-Based HPC Infrastructures Table of Contents Table of Co	
Products Group, Intel Corporation ustomer Solutions Technical Enabling Best Practices for Scaling Deep Learning Training and Inference with TensorFlow* On Intel® Xeon® Processor-Based HPC Infrastructures able of Contents Best Practices for Inserference able of Contents Best Provide Store of Contents Best Provi	Al Products Group, Intel Corporation Customer Solutions Technical Enabling Best Practices for Scaling Deep Learning Traini and Inference with TensorFlow* On Intel® Xeo Processor-Based HPC Infrastructures Table of Contents 1. Isots for Setup and Institution 1. Isots for Setup	tol
ustomer Solutions Technical Enabling	Customer Solutions Technical Enabling Best Practices for Scaling Deep Learning Traini and Inference with TensorFlow* On Intel® Xeo Processor-Based HPC Infrastructures Table of Contents 1. Trosofflow "Setup and Installation 1.3 Intail Uter's Horovord "Lbray 1.3 Intail Uter's Hor	ner
Best Practices for Scaling Deep Learning Training and Inference with TensorFlow* On Intel® Xeon® Processor-Based HPC Infrastructures Able of Contents Aster Pactors For Forerflow 'On long' Koon® Processors 11 Tensorflow Setup and Instillation 12 Instill VB, Instalaredy instillation 13 Instill VB, Constanting Reset-Sol 14 Instilling training Reset-Sol With Constants 15 Preparing the InageNet* 2012-18 Dataset 15 Asterior Bingle Net# 2012-	Best Practices for Scaling Deep Learning Traini and Inference with TensorFlow* On Intel® Xeo Processor-Based HPC Infrastructures Table of Contents 1. Best Practices for Tenserflow On Intel Xeon*Processes 1. Transflow Testing and Installation 1. Stransflow Testing Testing and Testing and Testing Andream 1. Stransflow Testing Testing and Testing Andream 1. Stransflow Testing Testing Andream 1. Transflow United Testing An	
Self Practices for Scaling Deep Learning training and Inference with TensorFlow* On Intel® Xeon® Processor-Based HPC Infrastructures biological statement of the self of the	Sest Practices for Scaling Deep Learning framing and Inference with TensorFlow* On Intel® Xeo Processor-Based HPC Infrastructures Table of Contents 1. Bist Practices for Tensorflow* On Intel* Xeon* Processors 1. Transfer Steps and Installation 1. Transfer Steps 1. T	ing
And Inference with TensorFlow* On Intel® Xeon® Processor-Based HPC Infrastructures able of Contents able of Contents	and Inference with TensorFlow* On Intel® Xeo Processor-Based HPC Infrastructures	iing
Processor-Based HPC Infrastructures bullet of contents Assessment States for Transformed Tools and Tools	Processor-Based HPC Infrastructures Table of Contents 1. Best Practices For TensorFlow "On Intel" Xeon ^a Processors. 1. TensorFlow "Setup and Installation 1. So Proparing the ImageNet 2012-11 Dataset 1. So Proparing the ImageNet 2012-11 Dataset 1. So Provide the ImageNet 2012-11 Dataset 1. So Dataset Striped on Listre" 1. So Dataset Striped on Listre 1. So Manifer Methods and Proper the ImageNetDataset 1. So Dataset Striped on Listre 1. So Dataset Striped Striped 1. So Dataset Striped Striped 1. So Dataset Striped Striped Networkers 1. So Dataset Striped Striped 1. So Dataset S	on®
Able of Contents Best Practices For Tensorflow" On Inst ^{er} Xeon ^a Processors 1.1 Tensorflow" Setup and Installation 1.2 Install RPI, Tensorflow" Con Inst ^{er} Xeon ^a Processors 1.3 Tensorflow" Setup and Installation 1.3 Install RPI, Tensorflow Tensorflow Tensorflow Tensorflow Tensorflow Tensorflow Tensorflow 1.3 Install RPI, Tensorflow Tensorflow Tensorflow Tensorflow Tensorflow 1.4 Install RPI, Tensorflow Tensorflow 1.5 Setup and Tensorflow Tensorflow 1.5 Setup and Tensorflow 1.5 Setup an	Table of Contents 1. Best Practices For Tensorflow 'On Intel' Xeon ⁴ Processors 1.1 Tensorflow'' Setup and Installation 1.2 Install MPL (In ot lared) installed 1.3 Install User's Horoword' Library. 1.4 Installing (Iron Lendy installed) 1.5 Install MPL were the mageNeet'R Dataset. 1.5.2 Aready have the mageNeet'R Dataset. 1.5.2 Aready Naming ReNAt-50 with Tr. cm. benchmarks. 1.6.3 Using Construction Singuistry Have to Multiple Workers. 1.6.4 Training on Singuistry Have the Trained Model. 1.6.5 Using Konstro Hite Trained Model. 1.6.6 Worker Hite System. 2.0 Using Singuistry Mage. 1.1 Using KN Monted File System. 2.0 Using Singuistry Mage. 1.1 Using KN Monted File System. 2.3 Using Singuistry Mage. 3.1 Installing Singularity mage. 3.2 Using Singuistry Mage. 3.1 Using KN Monted File System. 3.2 Using Singuistry Mage. 3.1 Using KN So	
Bask of Contents Best Practices For Tensorflow ² On Intel [®] Xeon ⁴ Processors 11 Tensorflow ² Setup and Installation 12 Tensorflow ² Setup and Installation 13 Tensorflow ² Setup and Installation 14 Installing ¹ (con, benchmarks, 14 Installing ¹ (con, benchmarks, 15.2 Already have the ImageNet/Dataset 15.2 Already have the ImageNet/Dataset 15.2 Already have the ImageNet/Dataset 15.3 Already have the ImageNet/Dataset 15.4 Olding One MPI 16.4 Using One MPI 16.4 Using One MPI 16.5 Using MVARICL ² 16.7 Valuation the Accuracy of the Trained Model 17.8 Using MVARICL ² 2.1 Installing Singularity 2.2 Building Singularity 2.3 Building Singularity 2.4 Building Singularity 2.3 Building Singularity 2.4 Building Singularity 2.5 Building Singularity 2.6 Building Singularity 2.7 Building Singularity	Table of Contents 1. Best Practices For TensorFlow* On Instel* Xeon* Processors. 1.1 TensorFlow* In the addition of the state of th	
Best Percline For Transcriptor On Insel Xuore Processors. 12 Install UPE, If not already installed 13 Install UPE, If not already installed 14 Installing IT, con, benchmarks. 15. Stream of Installation 14 Installing IT, con, benchmarks. 15. Stream of Installation 15. Stream of Installing IT, con, benchmarks. 16. Stream of Installing IT, con, benchmarks. 17. Stream of Installing IT, constant. 18. Stream of Installing IT, constant. 18. Stream of Installing Installing IT, constant. 18. Stream of Installing Installing IT, constant. 18. Stream of Installing Ins	1. Bear Practices For Transortions' On Intel ² Xouro Processors. 1. Bransoftwork Storay and Installation 1.2 Install WR, If not a transdy installed 1.3 Install WR, If not a transdy installed 1.3 Install WR, If not a transdy installed 1.4 Installing Eff. con. Denochmarks. 1.5. Proparing the Integrebert 2072-11: Dataset 1.5. Stransoft Row The Integrebert 2072-11: Dataset 1.5. St	
1.1 Tenserflow "Setup and Installation 1.2 Install VPI, Inter already installation 1.3 Install Uber's Horowork' Uberay. 1.4 Installing of the on kenchmarks. 1.5 Install Uber's Horowork' Uberay. 1.5 Lake Setup Setup and grouper the ImageNet Dataset. 1.5 Lake Setup	1.1 TensorFlow" Setup and Installation 1.2 Install MP, If not already installed 1.3 Install Up (in not already installed 1.3 Install Up (in on jenechmark) 1.5 Preparing the imageNet" 2012/1X Dataset 1.5 Preparing the reference with Mutiple Workers 1.6 Using imageNet Net 1.6 Preparing the Accuracy of the Trained Model 1.6 Using Singularity" 1.7 Dataset Setup of the Trained Model 1.6 Using Singularity" 1.8 Dataset Setup of the Trained Model 1.8 Dataset	2
1.2 Install UPP, if not already installed 1.3 Install UPP, if not already installed 1.4 Installing trigon, benchmarks, 1.4 Installing trigon, benchmarks, 1.5.1 Steps to developed and prepare the ImageNet Dataset. 1.5.2 Dataset Striped on Latter 1.5.2 Dataset Striped on Latter 1.6.3 Dataset Striped on Latter 1.6.4 Dataset Developed and the Striped on Latter 1.6.7 Dataset Developed and the Striped on Latter 1.6.7 Dataset Developed and the Striped on Latter 1.6.8 Public Developed and the Striped on Latter 1.7 Dataset Developed and the Striped on Latter 1.8.1 Dataset Developed and the Striped on Latter 1.8.2 Dataset Developed and the Striped on Latter 1.8.2 Dataset Developed and the Striped on Latter 1.8.1 Dataset Developed and the Striped on Latter 1.8.1 Dataset Developed and the Striped on Latter 1.8.1 Dataset Developed and the Striped on Latter 1.9.1 Dataset Developed and the Striped on Latter 1.9.1 Dataset Developed and the Striped on Latter 1.9.2 Dataset Developed and the Striped on Latter 1.9.1 Dataset Developed and the Striped on Latter 1.9.1 Dataset Developed and the Striped on Latter 1.9.2 Dataset Developed and the Striped on Latter 1.9.1 Dataset Developed and the Striped on Latter Devel	1.2 Install UMP, If not already installed 1.3 Install UMP, If not already installed 1.5 Preparing the ImageNet 2012-19 Dataset 1.5 Preparing the ImageNet 2012-19 Dataset 1.5 Paparing the ImageNet 2012-19 Dataset 1.5 A Urgen Open Marking Parameter ImageNet Dataset 1.5 A Urgen (Denothmark RegNet-50 orbit U, complementmarks 1.6 U, complementmark RegNet-50 1.6 J Using MVAPIC122 1.6 J Using MVAPIC122 1.6 J Using MVAPIC122 1.7 Dataset 1.8 J Using MVAPIC122 1.9 Dataset 1.9	2
 1.4 Initialization of the second state of the second stat	1 d Initialization of the state	
1.5 Preparing The ImageNet 2012-IX Dataset 1.5 S1 Steps to devine and and prepare the ImageNet Dataset 1.5 A Ready have the ImageNet-IX Dataset 1.5 A Ready have the ImageNet-IX Dataset 1.6 Dataset 1.6 Training on Single-Holde with Hollible Workers 1.6 Training on Single-Holde with Multiple Workers 1.6 A Training on Single-Holde with Multiple Workers 1.6 Training on Nutlpic Nodes with Multiple Workers 1.6 Training SingleJarity 3.1 Unit Simular SingleJarity 3.2 Building SingleJarity 3.3 Units Siume 3.4 Single SingleJarity 3.4 Single SingleJarity 3.5 Training on Single 3.5 SingleSingle 3.5 SingleSingleSingle 3.5 SingleSingleSingle 3.5 SingleSingleSingle 3.5 SingleSingleSingle 3.5 SingleSingl	1:5 Preparing the ImageNet: 2012-II: Dataset. 1:5.1 Steps to download and prepare the ImageNet Dataset. 1:5.2 Dataset Stripd on List? 1:6.2 Tasket Stripd on List? 1:7 Tasket Stripd Stripd 1:6 Tasket Stripd Stripd Stripd 1:6 Tasket Stripd Stripd Stripd 1:6 Tasket Stripd Stripd 1:6 Tasket Stripd Stripd Stripd 1:6 Tasket Stripd Stripd Stripd 1:6 Tasket Stripd S	
1.5.1 Steps to download and prepare the ImageNet Dataset 1.5.2 Already have the ImageNet Dataset 1.5.3 Dataset Striped on Istrief 1.6.1 Stagente Home Home Home Steps 1.6.1 ("Integration of the Istrief" 1.6.1 Using Open PMP. 1.6.1 Using Open PMP. 1.6.1 Using Open PMP. 1.6.3 Using Open PMP. 1.6.3 Using Open PMP. 1.6.3 Using Open PMP. 1.6.4 Using Istrief * MPI. 1.6.4 Using Istrief * MPI. 1.6.4 Using Open PMP. 1.6.3 Using Open PMP. 1.6.3 Using Open PMP. 1.6.4 Using Istrief * MPI. 1.6.4 Using Open PMP. 1.6.4 Using Istrief * MPI. 1.6.4 Using Open PMP. 1.6.7 Valuation the Accuracy of the Trained Model 1.5.7 Valuation the Accuracy of the Trained Model 1.5.8 Using WARKICK2* 2.1 Busing Texaser of the Trained Model 2.1 Busing Texaser of the Using Using Value 1.5.4 Using Value Value 1.5.5 Using WARKICK2* 1.	1.5.1 Steps to download and prepare the ImageNet Dataset. 1.5.2 Attack have the ImageNet-IX Dataset. 1.5.2 Attack Have the ImageNet-IX Dataset. 1.6.1 Example Round and prepare the ImageNet Dataset. 1.6.2 Example Round	
1.5.2 Markad's have the ImageNet-IX Dataset 1.5.2 Dataset Striped on Listre' 1.6 Example: Bunning ResNet-S0 with If _com_benchmarks 1.6.1 If _com_benchmark. Bastlet-S0. 1.6.1 Using Open MPI 1.6.2 Using interventional intervention of the Workers 1.6.4 Using interventional intervention 1.6.2 Using Development 1.6.2 Using intervention 1.6.3 Multi-State Intervence on the Trained Model 1.6.4 Using intervention 1.6.4 Using intervention 1.6.4 Using intervention 1.6.5 Training on Multiple Norkers 1.6.7 Polyanding Comparison 1.6.7 Volume 1.6.8 Multi-State Intervence on the Trained Model 1.6.8 Multi-State Intervence on the Trained Model 1.2 Bunning Transprise 1.5.7 Dataset 1.6.7 Dataset 1.6.7 Using Development 1.6.8 Using Multiple Norkers 1.6.8 Multi-State Intervence on the Trained Model 1.2 Bunning Transprise 1.6.7 Dataset 1.6.7 Datase	1.5.2 Already have the ImageNet-IK Dataset 1.5.2 Dataset Striped on Latter 1.6 Exampler Running ResNet-S0 with Y_con_benchmarks 1.6 Exampler Running ResNet-S0 with Y_con_benchmarks 1.6.2 Training on Single-Hold with Multiple Workers 1.6.3 Using WAMP(A22 1.6.4 Using Intel [*] MPI 1.6.4 Using Intel [*] MPI 1.6.5 Training on Multiple Roder with Multiple Workers 1.6.7 Evaluating the Accuracy of the Trained Model 1.2 Using Single-Hold Roder with Multiple Workers 1.2.7 Evaluating the Accuracy of the Trained Model 1.2 Using Single-Roder With Singularity 2.1 Installing Singularity 2.1 Singularity 2.3 Running TensorFlow With Singularity 2.3 Running TensorFlow With Singularity 3.2 Using Singularity 4.3 Evaluating Singularity 5.3 Rode Rode Rode Rode 5.3 Rode Rode Rode Rode Rode Rode Rode Rode	
1.3.2 Dataset Striped on Listre" 1.4.2 Example: Rounding ResNet-30 with the complementation 1.4.2 Treatment ResNet-30 with the complementation 1.4.3 Using gran Standard with Summer 1.4.3 Using gran Standard with Summer 1.4.4 Treatment 1.4.4 Using Instit Mark 1.4.4 Treatment 1.5.4 Using Gran Mark 1.5.4 Using Mark 1.5.5 Using mark 1.5.5 Usi	1.3. Dataset Striped on Listre* 1.6. Example Namoning ReNA-150 with If_com_benchmarks. 1.6. 11f_com_benchmarks. ReNA+150. 1.6. 11f_com_benchmarks. ReNA+150. 1.6. 3. Uty com_benchmarks. 1.6. 3. Uty com_benchmarks. 1.6. 3. Uty com_benchmarks. 1.6. 3. Uty com_benchmarks. 1.6. 4. Uty com_benchmarks. 1.6. 3. Uty com_benchmarks. 3. Uty com_bencon_benchmarks.	
An Example: Humang Reserved Source Sour	As Example's subming sectors built by Comp. Sendember 20 As Training on Single-Holds with Multiple Workers. A.4. Uraining on Single-Holds with Multiple Workers. A.5. Training on Multiple Nodes with Multiple Workers. A.5. Training the Accuracy of the Trained Model. Using Singularity*. All Notes of Single-Holds. As a second of the Accuracy of the Trained Model. Subming TensorFlow With Singularity. All Notes of Single-Holds. As a second of the Accuracy of the Trained Model. Subming TensorFlow With Singularity. All Notes of Single-Holds. As a second of the Accuracy of the Trained Model. All Notes of Single-Holds. As a second of the Accuracy of the Trained Model. All Notes of Single-Holds. As a second of the Accuracy of the Trained Model. As a second of the Accuracy of the Trained Model. As a second of the Accuracy of the Trained Model. As a second of the Accuracy of the Trained Model. As a second of the Accuracy of the Trained Model. As a second of the Accuracy of the Trained Model. As a second of the Accuracy of the Trained Model. As a second of the Accuracy of the Accura	
	1.6.2 Training on Single-Hode with Multiple Workers. 1.6.3 Using (Den HP) 1.6.4 Using (Den HP) 1.6.4 Using (Den HP) 1.6.5 Using (NM2PEL2) 1.6.5 Using (NM2PEL2) 1.6.5 Using (NM2PEL2) 1.6.7 Using Using on Multiple Nodes with Multiple Workers. 1.6.7 Using Using On Multiple Nodes with Multiple Workers. 1.6.7 Using Using On Multiple Nodes with Multiple Workers. 1.6.7 Using Using On Multiple Nodes with Multiple Workers. 1.6.7 Using Using Using On Multiple Nodes. 1.6.8 Using Nulti-Stream Inference on the Trained Model. 1.8.8 Using Storm Singularity Image. 2.8 Using Storm Singularity Image. 2.8 Using Storm Scheduler. 1.9 Using Storm Scheduler.	
1.6.3 Using Open MPI 1.6.4 Using Intel [®] MPI 1.6.5 Using INVAPICIA [®] 1.6.7 Training on Multiple Nodes with Multiple Workers 1.6.7 Toxinasing Nultiple Nodes with Multiple Workers 1.6.7 Evaluating the Accuracy of the Trained Model Using Singularity 1.2.2 Dualing Singularit	1.6.1 Using Open MPi 1.6.4 Using Interf MPI 1.6.5 Training on Multiple Roder with Multiple Workers 1.6.6 Training on Multiple Roder with Multiple Workers 1.6.7 Training on Multiple Roder with Multiple Workers 1.6.8 Training on Multiple Roder with Multiple Workers 1.6.8 Training on Multiple Roder with Multiple Workers 2.6 Training Stream Inference on the Trained Model 2.1 Using Singularity 2.2 Installing Singularity 2.3 Using Stream Schwarts 2.3 Using Stream Schwarts 2.4 Using Singularity 2.3 Using Stream Schwarts 3.4 Using Stream Schwarts 3.5 Using Stream Schwarts 3.6 Training TensorFlow 3.7 Using Stream Schwarts 3.8 Using Stream Schwarts 3.9 Using Stream Schwarts 3.1 Training Schwarts 3.5 TreamsofFlow Using Script 5.3 Tream Flow Schwarts 3.3 Singularity script 5.3.3 Rocipe His for TensorFlow wheed downloaded from a URL 5.3.3 Rocipe His for TensorFlow wheed downloaded from a URL 5.3.3 Rocipe Life for TensorFlow wheed downloaded from a URL 5.3.3 Rocipe Life for TensorFlow wheed downloaded from a URL 5.3.3 Rocipe Life for TensorFlow wheed downloaded	
	1.4. Using Intel [®] MPI 1.6.3. Using WAXPE122 1.6.3. Using WAXPE122 1.6.4. Using WAXPE122 1.6.4. Using Intel [®] MAXPE122 1.6.2. Using Interface on the Trained Model 1.0.1.2. Using Interface 1.0.1.2. Using Interface 1.1.1.2.1.2.1.2.1.2.1.2.1.2.1.2.1.2.	
16.5 Using WVAPLNC 16.6 Tailing with Nultiple Workers 1.6.7 Solution in the Nultiple Nodes with Nultiple Workers 1.6.7 Solution in the Nultiple Nodes with Nultiple Workers 1.6.7 Solution in the Nultiple Nodes with Nultiple Workers 1.6.7 Solution in the Nultiple Nodes with Nultiple Workers 1.6.7 Solution in the Nultiple Nodes with Nultiple 1.6.7 Solution in the Nultiple Nodes with Nultiple 1.6.7 Solution in the Nultiple 1.7 Solution in the Nultiple 1.7 Solution in the Nultiple 1.6.7 Solut	1.8. Using MVAPCLAY. 1.6. To Limit provide the transmitter of the second seco	
1.4.7 brainstage the Accoracy of the Trained Model. 1.4.8 Huild-Texan Inference on the Trained Model. Using Singularity: 2. Building Singularity in age. 3. Singularity in a set of the	14.5 Trainsing the Accuracy at the Trained Model. 14.8 Multi-Stream Inference on the Trained Model. 14.8 Multi-Stream Inference on the Trained Model. 14.8 Multi-Stream Inference on the Trained Model. 12.1 Installing Singularity*. 12.1 Installing Singularity*. 12.3 Running TensorFlow With Singularity. 13.1 Using WS Mounted File System. 13.1 Using WS Mounted File System. 13.2 Using Singularity insign excipt 15.3 Singularity insign excipt. 15.3 Singularity insign excipt. 15.3 Singularity conjust. 15.3 Running TensorFlow With System. 15.3 Running T	
1.6.8 Multi-Stream Inference on the Trained Model Uuling Singularity 2.1 Installing Singularity 2.2 Installing Singularity 2.2 Installing Singularity 2.1 Stalling Singularity 2.1 Installing Singularity 2.2 Installing 2.2 In	1.6.8 Multi-Stream inference on the Trained Model 2.1 Installing Singularity*. 2.1 studied Singularity*. 2.3 Building Singularity*. 2.3 Building Singularity*. 2.3 Numing TensorFlow With Singularity. 3.1 Using Silvers* Scheduler. 3.2 Using Silvers* Scheduler. 4.1 Building TensorFlow 4.2 Install TensorFlow with script. 5. Sample Scripts. 5. Sample Scripts. 5.3 Singularity script. 5.3 Singularity script. 5.3 Singularity script. 5.3 Negniarity script. 5.3 Singularity script. 5.4 Inference scripts. <	
Using Singularity* 2.2 Building Singularity* 2.3 Building Singularity image 3.3 Using West Meanuel File Singularity. 3.1 Using West Meanuel File System. 3.2 Using Sturms* Scheduler. 3.1 Starf Scheduler. 4.2 Install TencorFlow using script Sample Script. 5.3 TencorFlow Build script. 5.3 TencorFlow Build Script. 5.3 TencorFlow Build Script. 5.3 TencorFlow Build Script. 5.3 Recipe file for TencorFlow wheel downloaded from a URL 5.3 Recipe file for TencorFlow wheel downloaded from a URL 5.3 Recipe file for TencorFlow wheel con local file system. 5.4 Recipe file for TencorFlow wheel downloaded from a URL 5.3 Recipe file for TencorFlow wheel downloaded from a URL 5.3 Recipe file for TencorFlow wheel downloaded from a URL 5.3 Recipe file for TencorFlow wheel downloaded from a URL 5.3 Recipe file for TencorFlow wheel downloaded from a URL 5.3 Recipe file for TencorFlow wheel downloaded from a URL 5.3 Recipe file for TencorFlow wheel downloaded from a URL 5.4 Recipe file for TencorFlow wheel downloaded from a URL 5.3 Recipe file for TencorFlow wheel downloaded from a URL 5.4 Singer Script for tencorFlow the URL 5.4 Recipe file for TencorFlow wheel downloaded from a URL 5.3 Recipe file for TencorFlow wheel downloaded from a URL 5.4 Singer Script for tencorFlow the URL 5.4 Recipe for CencorFlow wheel downloaded from a URL 5.3 Recipe file for TencorFlow wheel downloaded from a URL 5.4 Recipe for CencorFlow wheel downloaded from a URL 5.3 Recipe file for TencorFlow wheel downloaded from a URL 5.4 Recipe for CencorFlow wheel downloaded from a URL 5.3 Recipe file for TencorFlow wheel downloaded from a URL 5.4 Recipe for CencorFlow wheel downloaded from a URL 5.3 Recipe file for TencorFlow wheel downloaded from a URL 5.4 Recipe for CencorFlow wheel downloaded from a URL 5.4	2. Using Singularity*. 2. Installing Singularity*. 2. 2 Building Singularity*. 2. 2 Building Singularity*. 3. 3 Running TencorFlow Wild Singularity. 3. Using WS A dotated This System. 4. 2 Installing Singularity and the System. 4. 2 Install ConsorFlow Long Script. 5. 3 ForsoFlow Long Script. 5. 3 Running TencorFlow Long Script. 5. 3 Running	10
2.1 Installing Singularity 2.3 Installing Singularity Using NTs And Shurret 3.4 Using NTS And Shurret 3.4 Using NTS Another of His System. 3.2 Using Shurns' Scheduler. Tensorflow Ibuild Instructions 4.3 Building Tensorflow Scheduler. 5.3 Singularity script. 5.3 Singularity script. 5.4 Singer Mir for Tensorflow wheel downloaded from a URL 5.3 Singularity script. 5.4 Singer Mir for Tensorflow Insert Scheduler. 5.4 Singer Mir Forsorflow Insert Scheduler. 5.4 Johnstein Tensorflow Insert Scheduler. 5.4 Singer Mir Forsorflow Insert Scheduler. 5.4 Singer Mir Common Scheduler. 5.5 Singer Mir Common Scheduler. 5.5 Singer Mir Common Scheduler. 5.5 Singer Mir Common Scheduler. 5.6 Mir Common Scheduler. 5.7 Singer Mir Common Scheduler. 5.8 Singer Mir Common Scheduler. 5.8 Singer Mir Common Scheduler. 5.9 Singer Mir Common Scheduler. 5.9 Singer Mir Common Scheduler. 5.1 Singer Mir Common Scheduler. 5.1 Singer Mir Common Scheduler. 5.1 Singer Mir Common Scheduler. 5.3 Singer Mir Common Scheduler. 5.4 Singer Mir Common Scheduler. 5.5 Scheduler. 5.6 Singer Mir Common Scheduler. 5.7 Scheduler. 5.8 Singer Mir Common Scheduler. 5.8 Scheduler. 5.9 S	2.1 Installing Singularity' ing. 2.1 Installing Singularity' ing. 3.1 Uniting Singularity' ing. 3.1 Uniting With And Silamet 3.1 Uniting With And Silamet 3.2 Uniting Silamet Sites System. 3.3 Uniting With Society Silamet	14
4.2 Similing 2-regionary image Segularity. 1.3 Uning Wis Mounted File System. 2.3 Uning Wis Mounted File System. 2.3 Uning Wis Mounted File System. 2.3 Uning Wis Mounted File System. 4.3 Install TensorFlow using script. Sample Script. 3.1 TensorFlow Unid script. 5.3 TensorFlow TensorFlow. 5.3 TensorFlow TensorFlow. 5.4 TensorFlow TensorFlow. 5.4 TensorFlow Information TensorFlow. 5.4 TensorFlow Information TensorFlow. 6.5 TensorFlow Import TensorFlow. 6.1 TensorFlow Import TensorFlow. 6.1 Check by running. 6.1 Another Common Error when Importing TensorFlow.	2.4 Building Singularity mange 3.5 Building WE Mounted File System 3.1 Using WE Mounted File System 3.1 Using WE Mounted File System 3.2 Using Survey Scheduler 4. Tensorflow Build Instructions 4. Tensorflow Build Instructions 4.2 Install Tensorflow using script 5.3 Singularity script 5.3 Singularity script 5.3 Revise file for Tensorflow wheel downloaded from a UR 5.3 Regularity script 5.3 Revise file for Tensorflow wheel on local file system 5.3 Regularity rorigits 5.3 Revise file for Tensorflow wheel on local file system 5.3 Regularity rorigits 5.3 Revise file for Tensorflow wheel on local file system 5.3 Regularity rorigits 5.3 Revise file for Tensorflow wheel on local file system 5.3 Regularity rorigits 5.4 Reference rorigits 5.4 Reference rorigits 5.5	14
Using MS And Slum? J. Using MS Another Bit Bystem. J. Using MS Another Bit Bystem. J. Using MS Another Bit Bystem. J. District Scheduler. Tensorflow build instructions. J. Singer Scheduler. S. J. Stems Scipt. J. Stems Scipt. J. Stems Scipt. J. Stems Scipt. J. Stems Scipt. J. Stems Scipt. J. Scheduler Science Scipt. J. Scheduler Science. J. Scheduler Science. J. Scheduler Science. J. Scheduler Science. J. Scheduler Science. J. J. Scheduler Science. J. J. Another Science. J. J. J. Another Science. J. J. Another Science. J. J. J. Another Science. J. J. J. Another J. Science. J. J. J. Another J. J. Science J.	3. Using W15 And Slurm* 3. Using W15 And Slurm* 3. Using Slurm Scheduler. 4. 1 Building TensorFlow . 4.1 Building TensorFlow . 4.1 Building TensorFlow . 5. Sample Scripts. 5. Sample Scripts. 5.3. Singularity cripts 5.3. Singularity cripts 5.3. Recipe file for TensorFlow wheel downloaded from a URL 5.3. Recipe file for TensorFlow wheel downloaded from a URL 5.4. Slignularity cripts 5.4. A lingularity cripts 5.4. Slignularity cripts 5.4. Slign	14
	1.1 Using NFS Hounted File System. 3.2 Using Sums "Scheduler. 4. Tenorflow Italia Instructions. 4.1 Install ConsorFlow. 4.2 Install ConsorFlow. 4.2 Install ConsorFlow Italia System. 5.3 Sample Scripts. 5.3 Engelarity scripts. 5.3 Engelarity scripts. 5.3 Recipe file for TensorFlow wheel downloaded from a URL 5.3 Recipe file for TensorFlow wheel downloaded from a URL 5.3 Recipe file for TensorFlow wheel downloaded from a URL 5.3 Recipe file for TensorFlow wheel downloaded from a URL 5.4 Inference cripts. 5.4 Inference cripts. 5.4 Inference cripts. 5.5 Inference Cript	15
3.2 Using Sturm' Scheduler. Tensorfow Judi Instructions 4.3 Building Tensorfow Learning Script 5.3 Sturms Script 5.4 Sturms Script 5.4 Sturms Script 5.4 Sturms Script 5.4 Sturms Script 5.5 Sturms Script 5.5 Sturms Script 5.4 Sturms Script 5.4 Sturms Script 5.5 Sturms Script 5.5 Sturms Script 5.4 Sturms Script 5.5 Sturms Script 5.5 Sturms Script 5.5 Sturms Script 5.6 Sturms Script 5.6 Sturms Script 5.7 Script Script 5.7 Script Script 5.8 Sturms Script 5.8 Sturms Script 5.8 Sturms Script 5.9 Script Script 5.9 Script Script 5.1 Sturms Script 5.1 Sturms Script 5.1 Sturms Script 5.3 St	3.2 Using Sturn' Scheduler. 4.3 Building TensorFlow 4.3 Building TensorFlow 4.3 Building TensorFlow 5.4 Install TensorFlow Ling Gript 5.5 Sturns origins 5.3 Singularity colpts 5.3 Singularity colpts 5.3.3 Recipe file for TensorFlow wheel downloaded from a URL 5.3.3 Recipe file for TensorFlow wheel on local file system 5.3.4 Singularity con-cript 5.4 Inference colpts 5	15
Tensorfow fauld Instructions 4.3 Install TensorFow 4.3 Install TensorFow 4.3 Install TensorFow 4.3 Install TensorFow Laboration 5.4 Second Sec	4. Tensorflow Build Instructions 4.1 Building TensorFlow 4.2 Install TensorFlow using script 5.3 Sample Script 5.3 ForsorFlow build script 5.3 Singularity script 5.3 Singularity script 5.3 Recipe file for TensorFlow wheel downloaded from a URL 5.3 Recipe file for TensorFlow wheel on local file system 5.4 Sample Script 5.4 Singularity run-script 5.5 Singularity run-script 5.4 Singularity run-script 5.4 Singularity run-script 5.5 Singularity run-scri	16
4.2 Install TensorForemap projet 5.3 Sample Scripts 5.4 Sample Scripts 5.5 Sample Scripts 5.5 Sample Scripts 5.3 Sample Scripts 5.4 Sample Scripts 5.4 Sample Scripts 5.5 Sample Scripts 5.4 Sample Script	4. 2. Intral Transforming script 5. Simple Script 5.1 TensofFlow build script 5.3 Simple Script 5.3 Singularity script 5.3.3 Recipe file for TensofFlow wheel downloaded from a URL 5.3.3 Recipe file for TensofFlow wheel downloaded from a URL 5.3.3 Recipe file for TensofFlow wheel downloaded from a URL 5.3.4 Internet scripts 5.4 Internet scripts	
Sample Scripts. 5.1 TensorFow build script. 5.2 Simpularity scripts. 5.3 Singularity is script. 5.3.4 Recipe file for TensorFlow wheel downloaded from a URL 5.3.8 Recipe file for TensorFlow wheel on local file system 5.3.4 Singularity run-script. 5.4 Singularity run-script. 5.4 Singularity run-script. 5.4 Integrating run-script. 5.4 Integrating run-script. 5.4 Integrating run-script. 5.5 Integrating run-script. 5.6 Singurating run-script. 6.1 Theorem run run run run. 6.1.2 Run id tor from the dynamically linked libraries. 6.1.3 Another Common Firor when Importing TensorFlow. 6.1.4 Another Common Firor when Importing TensorFlow.	5. Sample Scripts. 5. Sample Scripts. 5.3 Treasoftwo build script. 5.3 Singularity routint. 5.3 Singularity routint. 5.3 Decipe file for Treasoft Toor wheel downloaded from a URL 5.3 Decipe file for Treasoft Toor wheel on local file system 5.3 A Singularity run-script. 5.4 Interesting.	17
5.1 TensorFive build script 5.2 Sum script 5.3 Singularity script 5.3 Singularity script 5.3 Singularity script 5.3 Recipe file for TensorFlow wheel on local file system 5.3 Singularity in a script 5.4 Singurary run-script 5.4 Singurary files 5.4 Singurary files 5.4 Singurary files 5.1 Singurary files 5.1 Singurary files 5.3 Singurary 5.3 Singurary 5.3 Singurary 5.3 Singurary 5.4 Singurary 5.4 Singurary 5.4 Singurary 5.5 Singur	5.1 TensorFlow build script. 5.2 Siums ocjust. 5.3 Singularity script. 5.3.3 Install script. 5.3.3 Recipe file for TensorFlow whele downtoaded from a URL 5.3.3 Recipe file for TensorFlow whele on local file system. 5.3.4 Inference scripts. 5.4 Inference scripts.	
5.2 Stum scripts 5.3 Install script 5.3 Install script 5.3 Install script 5.3 Recipe file for TensorFlow wheel downtoaded from a URL 5.3 Becipe file for TensorFlow wheel on local file system 5.3 Becipe file for TensorFlow wheel on local file system 5.4 Inference scripts Troblemhoeting 6.5 Importing TensorFlow 6.5 Importing TensorFlow 6.5 Lock by running: 6.1.3 Acheck by running: 6.1.3 Acheck by running: 6.1.4 Achether Common Error when Importing TensorFlow. 6.5 I.4 Another Common Error when Importing TensorFlow.	5.2 Sum societs 5.3 Initiality script 5.3.3 Initiality script 5.3.2 Recipe file for TensorFlow wheel downloaded from a URL 5.3.3 Recipe file for TensorFlow wheel downloaded from a URL 5.3.4 Singuinty run-script 5.4 Inference scripts 5.4 Inference scripts	17
5. Singularity scripts 5. Singularity script 5. Singularity script 5. Singularity script 5. Solution	5.3 Singularity cripts 5.3.1 Install cript 5.3.2 Recipe file for TensorFlow wheel downloaded from a URL 5.3.3 Recipe file for TensorFlow wheel on local file system 5.3.4 Inference cripts 5.4 Inference cripts	21
53 finksii Gript 53 fickepi flor TensorFlow wheel downloaded from a UR. 53 Recipe file for TensorFlow wheel on local file system 54 Singdurity run-script 54 Singdurity run-script Torolarisationating 55 files for the system 55 files for the system 55 files for the system of the system 55 for system of the system of the system 55 for system of the system of the system 55 for system of the system of the system 55 for system of the system of the system 55 for system of the system of the system 55 for system of the system of the system 55 for system of the system of the system 55 for system of the system of the system 55 for system of the system of the system 55 for system of the system of the system 55 for system of the system of the system of the system 55 for system of the system of the system of the system 55 for system of the system of the system of the system 55 for system of the system of the system of the system 55 for system of the system of the system of the system 55 for system of the system of the system of the system 55 for system of the system of the system 55 for system of the system of the system 55 for system of the system of the system 55 for system of the system of the system 55 for system of the system of the system 55 for system of the system of the system 55 for system of the system 55 for system of the system 55 for system of the system of the system 55 for system 55 f	5.3.1 Initial script 5.3.2 Recipe file for TensorFlow wheel downloaded from a URL 5.3.3 Recipe file for TensorFlow wheel on local file system 5.4 Afreences excipts 5.4 Inferences excipts 5.4 Inferences excipts	24
S.1.3 Recipe file for Tensorflow wheel on local file system S.4. Brecipe file for Tensorflow integration of the system S.4. Brecipe file for Tensorflow S.4. Brecipe file for the system S.4. Brecipe file from the system S.5. Tensorflow integrating from the system S.5. Check by running: S.1.4. Another Common Error when Importing Tensorflow.	2.3.3 Recipe the for tensorFlow inneed ownsource round volu. 5.3.3 Recipe the for tensorFlow wheel on local file system 5.4 Recipe and they rune-cript 5.4 Interweight 6.4 Interweight	
S.A.S fingularity run-script S.A.S fingularity run-script S.A. Inference cripts Troubleshooting S.O. Tensor/Four import Issues S.1. Thmporting Tensor/Flow S.1.2 Run Ide to find the dynamically linked libraries S.1.3 Run Ider Common Error when Importing Tensor/Flow S.1.4 Another Common Error when Importing Tensor/Flow	5.3.4 Singularity run-script 5.4. Inference scripts. 6.4. Inference scripts.	
5.4. Inference acripts Troubleshooting Tesselfow 6.1.7. Insport Insures 6.1.2. Clack by running. 6.3.1. Clack by running. 6.3.1. Clack by running. 6.3.4. Achter Common Firer when Importing TensorFlow.	5.4 Inference scripts	
Troubleshooting 6.1 TensorFlow 6.31 Importing TensorFlow 6.3.1 Importing TensorFlow 6.3.2 Rui dot for find the dynamically linked libraries 6.3.3 Check by running: 6.3.3 Check hyr running: 7.3.4 Another Common Error when Importing TensorFlow. 6.3.4 Another Common Error when Importing TensorFlow.	6 Troubleshoating	29
6.1 Theorem TensorFlow Import Insues 6.1.1 Importing TensorFlow 6.1.2 Check by running: 6.3.1 Check by running: 6.3.4 Anether Common Firer when Importing TensorFlow.	e. nouveshooting	31
0.1 Importing Innorrient 6.1.2 Run Ido find the dynamically linked libraries 6.1.3 Check by running: 6.1.4 Another Common Error when Importing TensorFlow	6.1 TensorFlow Import Issues	31
6.1.3 Check by running: 6.1.4 Another Common Error when Importing TensorFlow	6.1.2 Bun led to find the dynamically linked libraries	
6.1,4 Another Common Error when Importing TensorFlow.	6.1.3 Check by running:	
	6.1.4 Another Common Error when Importing TensorFlow.	33
6.1.5 Verify that TensorFlow is Using right the version of GCC.	6.1.5 Verify that TensorFlow is Using right the version of GCC.	34

10



73

Intel distribution of Caffe

A fork of BVLC Caffe* maintained by Intel (Github)

The best-performing CPU framework for CNNs

<u>Supports low-precision inference</u> on Intel Xeon Scalable Processors (formerly known as Skylake)



CROSS PLATFORM INFERENCE

Optimization Notice Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

Intel Confidential under CNDA



OpenVINO[™] - TOOLKIT Machine Learning / Deep Learning Inference



Computer Vision SDK Deep Learning (DL) Deployment Toolkit Deep Learning Algorithms Optimized DL Frameworks

Optimization Notice Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

INTEL CONFIDENTIAL



INTEL® DISTRIBUTION OF OPENVINO TOOLKIT

Take your computer vision solutions to a new level with deep learning inference intelligence.

What it is

A toolkit to accelerate development of **high performance computer vision** & **deep learning into vision applications** from device to cloud. It enables deep learning on hardware accelerators and easy deployment across multiple types of Intel[®] platforms.

Who needs this product?

- Computer vision/deep learning software developers
- Data scientists
- OEMs, ISVs, System Integrators

Usages

Security surveillance, robotics, retail, healthcare, AI, office automation, transportation, non-vision use cases (speech, text) & more.



HIGH PERFORMANCE, PERFORM AI AT THE EDGE



STREAMLINED & OPTIMIZED DEEP LEARNING INFERENCE



HETEROGENEOUS, CROSS-PLATFORM FLEXIBILITY

Free Download ▶ software.intel.com/openvino-toolkit Open Source version ▶ 01.org/openvinotoolkit

Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others. Latest version is 2018 R5



What's Inside Intel[®] Distribution of OpenVINO[™] toolkit



OS Support: CentOS* 7.4 (64 bit), Ubuntu* 16.04.3 LTS (64 bit), Microsoft Windows* 10 (64 bit), Yocto Project* version Poky Jethro v2.0.3 (64 bit)



An open source version is available at 01.org/openvinotoolkit (some deep learning functions support Intel CPU/GPU only).

Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others. OpenVX and the OpenVX logo are trademarks of the Khronos Group Inc. OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos



Intel[®] Deep Learning Deployment Toolkit

For Deep Learning Inference

Model Optimizer

Trained

- What it is: A python based tool to import trained models and convert them to Intermediate representation.
- Why important: Optimizes for performance/space with conservative topology transformations; biggest boost is from conversion to data types matching hardware.

Inference Engine

- What it is: High-level inference API
- Why important: Interface is implemented as dynamically loaded plugins for each hardware type. Delivers best performance for each type without requiring users to implement and maintain multiple code pathways.



GPU = Intel CPU with integrated graphics processing unit/Intel $^{\circ}$ Processor Graphics

Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

Improve Performance with Model Optimizer



- Easy to use, Python*-based workflow does not require rebuilding frameworks.
- Import Models from various supported frameworks Caffe*, TensorFlow*, MXNet*, ONNX*, Kaldi*.
- 100+ models for Caffe, MXNet and TensorFlow validated. All public models on ONNX* model zoo supported.
- With support of LSTM and 3D Convolutional based networks and Kaldi framework / Kaldi Nnet2*, the model optimizer extends inferencing for non-vision networks.
- IR files for models using standard layers or user-provided custom layers do not require Caffe.
- Fallback to original framework is possible in cases of unsupported layers, but requires original framework.

Optimal Model Performance Using the Inference Engine

- Simple & Unified API for Inference across all Intel[®] architecture
- Optimized inference on large IA hardware targets (CPU/GEN/FPGA)
- Heterogeneity support allows execution of layers across hardware types
- Asynchronous execution improves performance
- Futureproof/scale your development for future Intel[®] processors



Transform Models & Data into Results & Intelligence

GPU = Intel CPU with integrated graphics/Intel® Processor Graphics/GEN GNA = Gaussian mixture model and Neural Network Accelerator

Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others. OpenVX and the OpenVX logo are trademarks of the Khronos Group Inc. OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos



Speed Deployment with Intel Optimized Pre-trained Models

OpenVINO[™] toolkit includes optimized pre-trained models to expedite development and improve deep learning inference on Intel[®] processors. Use these models for development & production deployment without the need to search for or to train your own models.

- Age & Gender
- Face Detection standard & enhanced
- Head Position
- Human Detection eye-level & high-angle detection
- Detect People, Vehicles & Bikes
- License Plate Detection: small & front facing
- Vehicle Metadata
- Human Pose Estimation
- Text Detection

Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

Pre-Trained Models

- Vehicle Detection
- Retail Environment
- Pedestrian Detection
- Pedestrian & Vehicle Detection
- Person Attributes Recognition Crossroad
- Emotion Recognition
- Identify Someone from Different Videos – standard & enhanced
- Facial Landmarks

Identify Roadside objects

- Advanced Roadside Identification
- Person Detection & Action Recognition
- Person Re-identification ultra small/ultra fast
- Face Re-identification
- Landmarks Regression
- Smart Classroom Use Cases
- Single image Super Resolution

Intel Confidential



Save Time with Deep Learning Samples & Computer Vision Algorithms

Samples

Use Model Optimizer & Inference Engine for public models & Intel pretrained models.

- Object Detection
- Standard & Pipelined Image Classification
- Security Barrier
- Object Detection for Single Shot Multibox Detector (SSD) using Asynch API
- Object Detection SSD
- Neural Style Transfer
- Hello Infer Classification
- Interactive Face Detection
- Image Segmentation
- Validation Application
- Multi-channel Face Detection

Computer Vision Algorithms

Start quickly with highly-optimized, ready-todeploy, custom-built algorithms using Intel pretrained models.

- Face Detector
- Age & Gender Recognizer
- Camera Tampering Detector
- Emotions Recognizer
- Person Re-identification
- Crossroad Object Detector
- License Plate Recognition
- Vehicle Attributes Classification
- Pedestrian Attributes Classification



Starting with OpenCV* & OpenVX*

- Intel-optimized functions for faster performance on Intel hardware
- Basic building blocks to speed performance, cut development time & allow customization
- All-in-one package



- Well-established, open source, computer vision library
- Wide variety of algorithms and functions available



- Targeted at real time, low power applications
- Graph-based representation, optimization & execution
- 11 samples included

Optimization Notice

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.



Legal Disclaimer & Optimization Notice

Performance results are based on testing as of August 2017 to September 2018 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure.

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. For more complete information visit <u>www.intel.com/benchmarks</u>.

INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS". NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Copyright © 2018, Intel Corporation. All rights reserved. Intel, the Intel logo, Pentium, Xeon, Core, VTune, OpenVINO, Cilk, are trademarks of Intel Corporation or its subsidiaries in the U.S. and other countries.

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



Software