In the world of HPC, every increase in processor performance creates an opportunity to accelerate the pace of research and development – and researchers, scientists, and engineers around the world are using high performance computing (HPC) clusters to push the boundaries of knowledge and innovation and they need ever higher performance to solve today’s increasingly complex challenges. From desk-side clusters to the world’s largest supercomputers, Intel® products and technologies for HPC provide powerful and flexible options for optimizing performance across the full range of HPC workloads.

Get deep insights into cutting-edge programming techniques and tools required to achieve the highest performance on Intel® Architecture using C/C++ or Fortran. Learn from technical experts on how to modernize your code legacy or develop brand-new code in order to maximize software performance on current and future Intel® Xeon and Xeon Phi™ processors.

**Translating Multicore Power into Application Performance**

*Intel leads the way to technology innovation. With our processors and software tools and our partners, we strive to bring the know how that you required to parallelize your environment to reap the benefits of today’s competitive world and domain.*

*The workshops that we are conducting are a first step to this and we hope that you as a developer, team leader, strategist, analyst in this environment are able to take full advantage of this offering and find the attached agenda for 2 days.*
<table>
<thead>
<tr>
<th>Duration</th>
<th>Tools</th>
<th>Title</th>
<th>Agenda</th>
</tr>
</thead>
</table>
| 30 min   | NA                         | Parallel Programming Concepts            | Introduction to Parallelism 
Exploiting Parallelism 
Distributed Memory 
Issues in Parallelism 
Improving Performance                                                                                                                                                                                                                                                                  |
| 60 min   | Intel® Advisor XE          | Understanding your code and how to parallelize it | Intel Advisor XE Overview 
Features 
Workflow 
Understanding the results 
Demo 1 : Survey – Determining if your code has opportunities for Parallelism 
Demo 2 : Determining correctness in your code Vectorization Advisor Overview                                                                                                                                                                                                 |
| 60 min   | Intel®C++ Compiler        | Simple way to build your application for Intel Architecture | OS/Architecture Support 
Compatibility 
Compiler Highlights 
Overview of Performance Libraries and Multithreading Support Demo 1 : Simple command line samples accompanying the lecture of optimization/auto-parallelization/auto-vectorization features.                                                                                                                                 |
| 30 min   | Intel® C++ Compiler: Vectorization | Vectorization for Intel® C++ & Fortran Compiler | Introduction to SIMD for Intel® Architecture 
Vector Code Generation 
Compiler & Vectorization 
Validating Vectorization Success 
Reasons for Vectorization Fails 
Vectorization of Special Program Constructs & Loops                                                                                                                                                                                                                                     |
| 30 min   | Intel® Cilk Plus          | Programming with Intel Cilk Plus.         | What is Intel Cilk Plus? 
clik spawn & cilk sync 
cilk for Some programming Notes 
Reducers 
Summary                                                                                                                                                                                                                                                                               |
| 45 min   | OpenMP                     | Industry leading parallelization techniques | What is OpenMP? 
Parallel regions 
Data-Sharing Attribute Clauses 
Worksharing 
OpenMP 3.0 Tasks 
Synchronization 
Runtime functions/environment variables 
what's new in OpenMP 4.0 
Optional Advanced topics                                                                                                                                                                                                                                                                 |
| 60 min   | Intel® VTune™ Amplifier XE | Revealing the performance aspects in your code | Performance tuning methodology 
Quick overview Introduction to Intel® VTune™ Amplifier XE 
Types of performance analysis 
Data collecting technologies Hotspot analysis 
Demo 1: Finding Performance Hotspots Concurrency Analysis Locks and Waits Analysis Command Line Interface, Installation, Remote Collection Event Based Performance Analysis Overview                                                                                                                                                            |
| 30 min   | Intel® Inspector XE        | Correctness checking - identifying memory and threading errors in serial and parallel programs | Intro to Intel® Inspector XE 
Analysis workflow 
Memory problem analysis 
Demo 1: Finding memory errors 
Threading problem Analysis 
Demo 2: Finding threading errors 
Preparing setup for analysis 
Managing analysis results 
Integration with debugger                                                                                                                                                                                                                                                         |
| 30 min   | Intel® TBB                 | Powerful parallel templates library for C++ | Introduction Intel® Threading Building Blocks overview Parallel models comparison Summary                                                                                                                                                                                                                                                                                             |
| 30 min   | Intel® IPP & MKL           | Accelerates math processing routines that increase application performance and reduce development time | Introduction and Overview 
Getting Started with Intel® Math Kernel Library 
Conditional Numerical Reproducibility 
Performance Charts 
Intel® IPP Overview 
What is Intel® IPP 
Intel IPP Benefits 
Intel IPP Domains 
References                                                                                                                                                                                                                                                                 |
## Intel Software Tools Training – HPC Segment

### Day 1

<table>
<thead>
<tr>
<th>Duration</th>
<th>Tools</th>
<th>Title</th>
<th>Agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td>60 min</td>
<td>NA</td>
<td>Intel® Architecture for Software Developers</td>
<td>Brief introduction to 4th Generation Intel® Core™ architecture Intel® Xeon Phi™ coprocessor architecture overview</td>
</tr>
<tr>
<td>30 min</td>
<td>NA</td>
<td>Intel® Xeon Phi™ Coprocessor</td>
<td>Intel® Xeon Phi™ coprocessor architecture Cache hierarchy, interconnect and communications architecture</td>
</tr>
<tr>
<td>30 min</td>
<td>NA</td>
<td>Parallel Programming Concepts</td>
<td>Introduction to Parallelism Exploiting Parallelism Shared Memory Distributed Memory Issues in Parallelism Limitations Improving Performance</td>
</tr>
<tr>
<td>30 min</td>
<td>Intel® Advisor XE</td>
<td>Understanding your code and how to parallelize it</td>
<td>Intel Advisor XE Overview Features Workflow Understanding the results Lab 1: Survey – Determining if your code has opportunities for Parallelism Lab 2: Determining correctness in your code</td>
</tr>
<tr>
<td>60 min</td>
<td>Intel® Advisor XE – Vectorization Advisor</td>
<td>Understanding vectorization and how it impacts performance</td>
<td>Vectorization Advisor Overview Features Workflow Understanding the results Examples &amp; Lab</td>
</tr>
<tr>
<td>45 min</td>
<td>Intel® C++ Compiler</td>
<td>Simple way to build your application for Intel Architecture</td>
<td>OS/Architecture Support Compatibility Compiler Highlights Overview of Performance Libraries and Multithreading Support Lab 1: Simple command line samples accompanying the lecture of optimization/auto-parallelization/auto-vectorization features.</td>
</tr>
<tr>
<td>30 min</td>
<td>Intel® C++ Compiler : Vectorization</td>
<td>Vectorization for Intel® C++ &amp; Fortran Compiler</td>
<td>Introduction to SIMD for Intel® Architecture Vector Code Generation Compiler &amp; Vectorization Validating Vectorization Success Reasons for Vectorization Fails Vectorization of Special Program Constructs &amp; Loops</td>
</tr>
<tr>
<td>45 min</td>
<td>Intel® Cilk Plus</td>
<td>Programming with Intel Cilk Plus.</td>
<td>What is Intel Cilk Plus? cilk spawn &amp; cilk_sync LAB 1 - cilk spawn and cilk_sync cilk for Some programming Notes LAB Activity 2 - Parallelizing using cilk_for Reducers Summary</td>
</tr>
<tr>
<td>30 min</td>
<td>OpenMP</td>
<td>Industry leading parallelization techniques</td>
<td>What is OpenMP? Parallel regions Data-Sharing Attribute Clauses Worksharing OpenMP 3.0 Tasks Synchronization Runtime functions/environment variables what's new in OpenMP 4.0 Optional Advanced topics</td>
</tr>
<tr>
<td>75 min</td>
<td>Intel® VTune™ Amplifier XE (Generics)</td>
<td>Revealing the performance aspects in your code</td>
<td>Performance tuning methodology Quick overview Introduction to Intel® VTune™ Amplifier XE Types of performance analysis Data collecting technologies Hotspot analysis Lab 1: Finding Performance Hotspots Concurrency Analysis Lab 2: Analyzing Parallelism Locks and Waits Analysis Introduction to Performance Monitoring Unit Event Based Performance Analysis Overview</td>
</tr>
<tr>
<td>Duration</td>
<td>Tools</td>
<td>Title</td>
<td>Agenda</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
<td>----------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>45 min</td>
<td>Intel® Inspector XE</td>
<td>Correctness checking - Identifying memory and threading errors in serial and parallel programs</td>
<td>Intro to Intel® Inspector XE Analysis workflow Memory problem analysis Lab 1. Finding memory errors Threading problem Analysis Lab 2. Finding threading errors Preparing setup for analysis Managing analysis results Integration with debugger Automated</td>
</tr>
<tr>
<td>30 min</td>
<td>Intel® TBB</td>
<td>Powerful parallel templates library for C++</td>
<td>Introduction Intel® Threading Building Blocks overview Tasks concept Generic parallel algorithms Task-based Programming Performance Tuning Parallel pipeline Concurrent Containers Scalable memory allocator Synchronization Primitives Parallel models comparison Summary Labs Lab 1: Parallel For</td>
</tr>
<tr>
<td>30 min</td>
<td>Intel® MKL</td>
<td>Accelerates math processing routines that increase application performance and reduce development time</td>
<td>Introduction and Overview Getting Started with Intel® Math Kernel Library Conditional Numerical Reproducibility Performance Charts References</td>
</tr>
<tr>
<td>30 min</td>
<td>Intel® IPP</td>
<td></td>
<td>Intel® IPP Overview What is Intel® IPP Intel IPP Benefits Intel IPP Domains Intel IPP Naming and Linkage</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OPTIONAL</td>
</tr>
<tr>
<td>60 min</td>
<td>ITAC</td>
<td></td>
<td>Advanced ITAC Overview – Effective usage of the ITAC API functions: user timing system available and C++ class – Limiting trace file size by filtering unimportant events – Limiting trace file size by further API functions: switching tracing on and off – Case Study BGQD Lab1: ITAC default: timing baseline for Poisson with and without ITAC Lab2: Tcollect: using the --tcollect compiler flag Lab3: Filter: applying filtering while tracing and while compilation Lab4: ON-OFF: switching tracing on and off by using API functions Lab5: Timing: how to use an existing timing system for effective instrumentation</td>
</tr>
<tr>
<td>90 min</td>
<td>MPI</td>
<td>Analysis of MPI Programs</td>
<td>Analysis of MPI programs Overview: Scaling analysis: speedup and efficiency for whole program and compute part Optimal choice of the process grid and mapping on the data grid Ideal network simulator Load balancing Message passing profile and process mapping Bandwidth analysis</td>
</tr>
<tr>
<td>90 min</td>
<td>MPI</td>
<td>MPI Correctness Checking and Debugging</td>
<td>MPI correctness checking overview Common MPI problems Correctness checking goals Command line or GUI output Supported checks, configuration, and usage ITAC failsafe mode for crashing applications Using Inspector XE for hybrid Intel MPI codes Intel MPI build-in debug output Debugging Intel MPI codes Lab Activity 1 Use Correctness Checking on an erroneous MPI program Activity 2 Use the ITAC failsafe mode to enforce writing of a trace file Activity 3 Use the ITAC failsafe mode to trace an erroneous program Activity 4 Use the Intel MPI debug output Activity 5 Use the Intel MPI debugger support</td>
</tr>
<tr>
<td>90 min</td>
<td>MPI</td>
<td>MPI Tuning</td>
<td>Motivation Automatic Tuning mptune o Allgather mptune process mptune parameters Manual Tuning Basic Advanced Black belt Labs</td>
</tr>
</tbody>
</table>