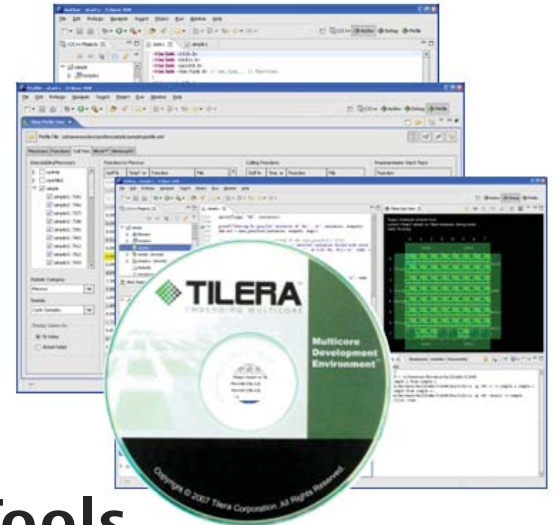


### Complete and Integrated Development Environment for Multicore Processors

Tilera's Multicore Development Environment™ (MDE) is a complete, standards-based multicore programming solution that enables developers to take full advantage of the parallel-processing potential of the Tile Processor™ architecture. Older multicore models required all programming operations be done in a core-by-core fashion, making it impossible to program, debug, or profile more than a handful of cores efficiently. With Tilera's MDE suite, developers can move to larger and more complex multicore applications using its advanced collective views and profiling technologies.

The MDE programming suite reflects a mix of the best of current tools and technology with next-generation solutions designed for the large-scale multicore world. MDE's familiar development environment helps developers get their applications to market quickly, while the new tools help them take full advantage of the technology Tilera® offers.



## The *Best* Industry-Standard Tools and the Most Innovative Next-Generation Solutions for Multicores

**Runtime Software Stack:** The MDE offers a complete software stack including hypervisor, operating system, device drivers, and user space libraries.

- Standard Linux, C, and C++ user space libraries
- Tilera Multicore Components™ (TMC) user library for communication and performance optimization
- Standard operating system using SMP Linux 2.6
- Hypervisor for hardware abstraction and virtualization
- High speed packet processing and load balancing drivers

**Standard and Enhanced Development Tools:** The MDE provides a complete set of multicore development tools including:

- State-of-the-art ANSI C/C++ compiler
- Robust Eclipse Integrated Development Environment (IDE)
- Standard command line tools for GDB, gprof, and oprofile
- Multicore debugger with a collective view of all cores
- Multicore profiler with source line granularity
- Complete system simulator and hardware development platform

Features	Benefits
<ul style="list-style-type: none"> <li>• C/C++ programming model</li> <li>• Supports Linux and SMP Linux</li> <li>• Standard tool chain based on Eclipse and GNU</li> </ul>	<ul style="list-style-type: none"> <li>• Easy to leverage existing code</li> <li>• Take advantage of an extensive body of Open Source applications</li> <li>• Familiar programming environment speeds development</li> </ul>
<ul style="list-style-type: none"> <li>• Support for various programming environments:               <ul style="list-style-type: none"> <li>- Full SMP Linux</li> <li>- Zero Overhead Linux™ (ZOL)</li> <li>- Bare Metal Environment (BME)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Standard and familiar environment to ramp up time</li> <li>• Real time performance results</li> <li>• Integration of control plane and dataplane applications</li> </ul>
<ul style="list-style-type: none"> <li>• Linux user libraries</li> <li>• Standard C and C++ libraries</li> <li>• Software-based high-speed packet processing drivers</li> <li>• TMC libraries for communication and optimization</li> </ul>	<ul style="list-style-type: none"> <li>• Run existing applications with dependencies on standard libraries</li> <li>• Focus customer development on differentiating components</li> <li>• Ability to implement proprietary packet processing protocols</li> <li>• Efficient communication between cores to improve performance</li> </ul>
<ul style="list-style-type: none"> <li>• Multicore-aware profiling and debugging</li> <li>• Standard gdb and gprof interfaces</li> <li>• Eclipse-based graphical interface for data display and organization</li> </ul>	<ul style="list-style-type: none"> <li>• Speed development by accurately identifying functional or performance issues</li> <li>• Familiar and standard interfaces, reducing ramp up time</li> </ul>
<ul style="list-style-type: none"> <li>• Timing accurate simulator</li> <li>• Multiple hardware evaluation platforms including PCIe cards, AMC platforms, and standalone appliances</li> </ul>	<ul style="list-style-type: none"> <li>• Fast software and hardware prototyping</li> <li>• Maximum visibility into processors state to help identify potential issues</li> </ul>

# Multicore Development Environment™ – Product Brief

## Run-Time Software Environment

Tilera provides several operating environments to accommodate the different application needs.

### Standard SMP Linux Environment

- Standard operating environment with a full SMP Linux OS, a hypervisor, and application libraries
- For applications and control plane code requiring operating system services

### Zero Overhead Linux (ZOL)

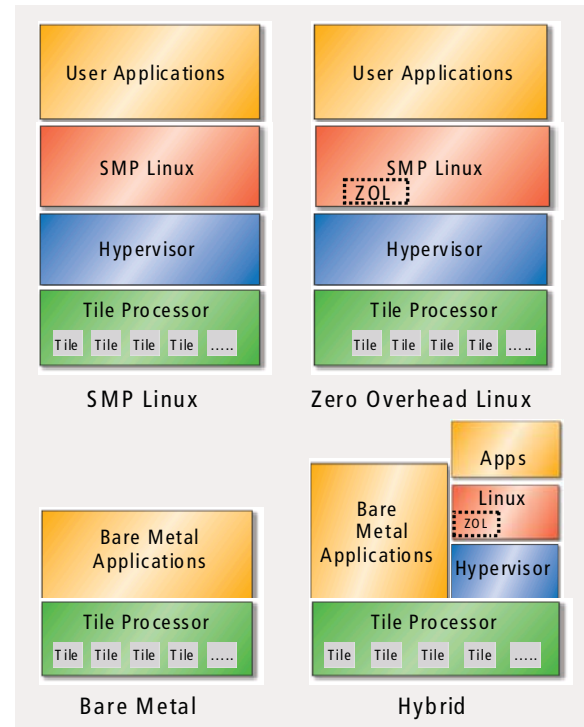
- Similar to standard SMP environment
- Avoid Linux overhead: OS interrupts, timer ticks, TLB shoot downs, etc.
- Transparent to programmer – no software change required
- For high performance real-time dataplane applications

### Bare Metal Environment

- Full control of the hardware
- Rich set of libraries for memory, communication, and I/O
- For embedded applications requiring fine grain control of memory and I/O
- For customers who want to port their own OS

### Hybrid Environment

- Using two or all three of the models listed above
- Each environment can be run on one or more tiles
- Ideal for customers aggregating dataplane and control plane software on one chip



## Development Tools

Tilera provides familiar, industry-standard development tools, as well as new tools developed specifically for the multicore environment.

### Standard ANSI C/C++ Compiler

- Full ANSI C/C++ support
- GNU extensions
- State-of-the-art optimizations (inlining, unrolling, profile guided optimization)
- SIMD intrinsics

### Eclipse-Based IDE

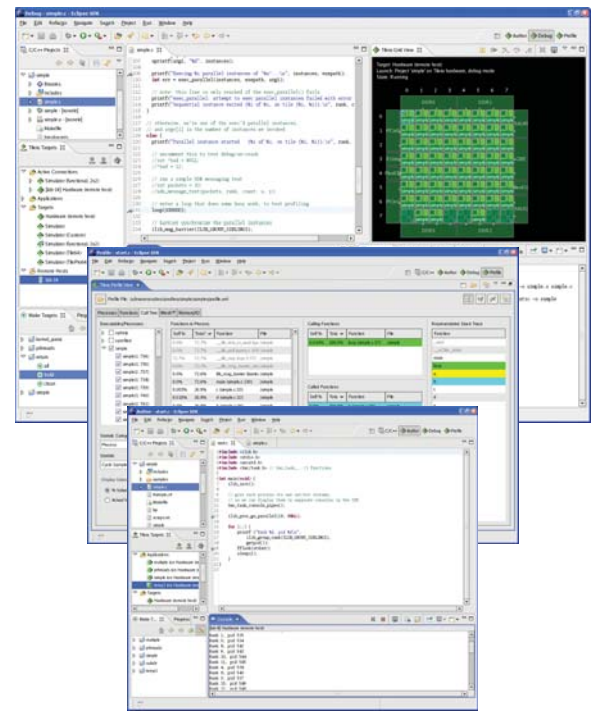
- Graphics-based authoring environment
- Interactive debugger and profiler
- Grid View of all tiles for aggregate control

### Parallel Debugger

- “Whole-application” debugging provides control over all tiles running an application
- Control over the entire processor or individual cores
- Visual representation of the state of all cores.
- Standard debug functions: breakpoints, stepping, and stopping

### Parallel Profiler

- Graphical IDE for navigation
- Standard gprof for basic function-level profiling
- Detailed run-time information
  - Application statistics per process
  - Source-level tracking per function
  - Representative stack trace
  - Utilization details per CPU, cache, memory, and network



For more information on Tilera products, visit [www.tilera.com](http://www.tilera.com).

Tilera Corporation  
2333 Zanker Road  
San Jose, CA 95131

Phone: (408) 383-9292  
Fax: (408) 383-9225  
[www.tilera.com](http://www.tilera.com)