V8 engine of Node.js on IA: JavaScript-JITTED x86 machine code mapping profiling support and X87 Quark processor enabling

chunyang.dai@intel.com, Intel Company
Agenda

• JavaScript-JITTED x86 machine code mapping profiling support in VTune
  • Background: Scan of current profiling tools for Node.js
  • What’s Intel VTune and the JavaScript code / machine code mapping
  • How to use it in Node.js

• X87 Quark processor enabling for V8 / Node.js
  • Background
  • Intel’s effort for X87 Quark processor enabling
  • How to build Node.js for Quark processor

• Q / A?
Scan of Current Node.js profiling for JavaScript code

- **Existing Hierarchy of Node.js profiling**
  - Node.js' C++ level profiling – any platform-dependent profiling tool works
  - Node.js' V8 profiling with limited JavaScript function information
    - Pass `--prof` flag to node and generate the V8 log file
      - Provided function level profiling data.
      - No detailed profiling information for JavaScript code
    - “perf” and v8 “--ll_prof” support on Linux etc
      - Provided function level profiling data.
      - Provided machine code level profiling data
    - No detailed profiling information for JavaScript code

- **A big missing**
  - Node.js' V8 profiling with detailed JavaScript source code mapping to JITTED assemble machine code
Intel VTune profiler and JITTED code mapping

• What’s Intel VTune profiler?
  • A commercial application for software performance analysis.
  • It has both GUI and command line interfaces.
  • It is available for Linux, Windows and Android operating systems.
  • It provides functionality(API) to profile runtime generated code.
  • It runs on top of Intel processors (please visit [Intel VTune Amplifier](https://software.intel.com/en-us/vtune) for details)

• VTune’s JITTED code and source code mapping API:
  • Easy to use.
  • Provide detailed dynamic generated code line <-> machine code level profiling/mapping capacity
Enable Node.js’ JavaScript-JITTED x86 machine code mapping profiling support in VTune

- **Upstream Status:**
  - This feature is landed in Node.js by pull request #3785.
  - Thanks for Ben Noordhuis and others’ timely review
  - It should be available since Node.js V5.2.0.

- **How to build a Vtune-enable Node.js:**
  - Step 1: Download Node.js code from github.
  - Step 2: Compile Node.js with special flag:
    - on Windows OS:
      ```
      ./vcbuild.bat --enable-vtune [other flags]
      ```
    - on Linux and android:
      ```
      ./configure --enable-vtune-profiling [other flags]
      ./make
      ```
Profile JavaScript in Vtune (1)

• How to do Node.js profiling in Vtune:
  • Step 1: Get/Install Vtune.
  • Step 2: Open Intel VTune and create one project in it.
  • Step 3: Configure VTune project.
Profile JavaScript in Vtune (2)

- Step 4: configure profiling type and start profiling

- Step 5: show result do analysis.
Profile JavaScript in Vtune (3)

- Step 5: show result do analysis.
Profile JavaScript in Vtune (4)

• Step 5: show result do analysis.
Agenda

• JavaScript-JITTED x86 machine code mapping profiling support in VTune
  • Background: Scan of current profiling tools for Node.js
  • What’s Intel VTune and the JavaScript code / machine code mapping
  • How to use it in Node.js

• X87 Quark processor enabling for V8 / Node.js
  • Background
  • Intel’s effort for X87 Quark processor enabling
  • How to build Node.js for Quark processor
• Q / A?
X87 Quark processor enabling for V8 / Node.js

**Background**

- **Intel X87:**
  - x87 is only floating point ISA of the x86 architecture instruction set before SSE debut
  - x87 co-exist with SSE enable processor
- **Intel Quark Family processor:**
  - Quark processor is designed for ubiquitous computing markets and the Internet of Things (IoT), from automotive to industrial to wearables
  - Quark processor’s float point currently is x87-only ISA
- **Google V8’s original roadmap of IA32 support:**
  - SSE/none-SSE V8 co-existing in unified IA32/X64 V8 main port before V8 v3.26
  - SSE/none-SSE V8 port is forked since V8 3.27
  - V8 embedded applications such as Node.js needs a dedicated V8 port for X87-only platform after V8 3.26

<table>
<thead>
<tr>
<th></th>
<th>&lt;= 3.26 release</th>
<th>&gt; 3.26 release</th>
</tr>
</thead>
<tbody>
<tr>
<td>With SSE2 (IA32/X64)</td>
<td>Yes. With optimized compilers</td>
<td>Yes. With optimized compilers</td>
</tr>
<tr>
<td>Non-SSE (including X87-only Quark processor)</td>
<td>Yes. But Low priority. No optimized compiler support.</td>
<td>Not in major x86 master port (Need Intel’s work on X87 port)</td>
</tr>
</tbody>
</table>
X87 Quark processor enabling for V8 / Node.js

• Intel’s effort for Node.js enabling on Quark:
  
  • Functionality
    • Intel created and being maintainer of V8 X87 port inside V8 upstream code repository
    • From V8 v3.27 to current V8 4.8’s release branch and future V8 release branch
    • Intel is making sure every release of Node.js works well on Quark since node.js 0.12
  
  • Performance
    • Intel implemented optimized compilers for v8 x87: 1) Crankshaft compiler since V8 release v3.27 2) Turbofan compiler since V8 release v4.5
X87 Quark processor enabling for V8 / Node.js

• How to build Node.js on Intel Quark processor specifically.

Step 1: download node.js source code.
Step 2: do configuration
   ./configure –dest-cpu=ia32
Step 3: modify the config.gypi, add one line as below (marked Red)
   'target_arch': 'ia32',

                           ......................
   'v8_no_strict_aliasing': 1,
   'v8_optimized_debug': 0,
   'v8_random_seed': 0,
   'v8_use_snapshot': 'true',
   'v8_target_arch': 'x87',
   'want_separate_host_toolset': 1}}

Step 4: ./make
Q / A.

• You can also reach me by email: chunyang.dai@intel.com