Eclipse Linux Tools Project

Linux Symposium Tracing Summit 2009

Andrew Overholt
Red Hat
Agenda

- Background
- Today's focus: tools for C/C++ developers built on CDT
- Memory profiling
- Call profiling
- Work areas
- Future plans and how to get involved
Agenda

- Background
- Today's focus: tools for C/C++ developers built on CDT
- Memory profiling
- Call profiling
- Work areas
- Future plans and how to get involved
Agenda

- Background
- Today's focus: tools for C/C++ developers built on CDT
- Memory profiling
- Call profiling
- Work areas
- Future plans and how to get involved
C/C++ Tools

- Developer-focused
- Sane defaults
- Integrate with CDT functionality
GNU Autotools
Libhover

```c
void *blah = malloc;

if (display_name)
    display = gdk_display_open;
else
{
    GList *display = NULL;
    const char *period = str;
    if (period)
    {
        gulong n;
        ...
    }
}
```

This function returns a pointer to a newly allocated block of size bytes long, or a null pointer if the block could not be allocated.
```
probe kernel.function("vfs_read").return {
    reads[execname()] += $return
}

probe kernel.function("vfs_write").return {
    writes[execname()] += $return
}

probe timer.s(1) {
    foreach (p in reads)
        total_io[p] += reads[p]
    foreach (p in writes)
        total_io[p] += writes[p]
    foreach (p in total_io[10])
        printf("%15s r: %8d KiB w: %8d KiB\n",
            p, reads[p]/1024,
            writes[p]/1024)
    printf("\n")
    # Note we don't zero out reads, writes and total_io,
```
Agenda

- Background
- Today's focus: tools for C/C++ developers built on CDT
- Memory profiling
- Call profiling
- Work areas
- Future plans and how to get involved
Valgrind

```c
int *foo() {
    return (int *)malloc(SIZE);
}

void bar(int *ptr) {
    free(ptr);
}
```

Valgrind test (memcheck) [memcheck] /usr/bin/valgrind (17/03/09 3:07 PM)

- 4,000 bytes in 100 blocks are definitely lost in loss record 1 of 1 [pid: 28810 / tid: 3]
  - at 0x4A073F: malloc (vg_replace_malloc.c:207)
  - by 0x400589: foo (valgrindtest.c:21)
  - by 0x400541: main (valgrindtest.c:12)
Valgrind
# Valgrind

![Valgrind](image)

<table>
<thead>
<tr>
<th>Location</th>
<th>Ir</th>
<th>I2mr</th>
<th>I2mr</th>
<th>Dt</th>
<th>D1mr</th>
<th>D2mr</th>
<th>Dv</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total [PID: 23963]</td>
<td>1,419,193</td>
<td>2,873,636</td>
<td>68,515</td>
<td>387,367,57</td>
<td>2,717,263</td>
<td>600,210</td>
<td>13</td>
</tr>
<tr>
<td>m_trampoline.S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>egg-editable-toolbar.c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>egg-toolbars-model.c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eggtypebuiltins.c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>totem-scsaver.c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eog-application.c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eog-ciebug.c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eog-image.c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eog-job-queue.c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Agenda

- Background
- Today's focus: tools for C/C++ developers built on CDT
- Memory profiling
- Call profiling
- Work areas
- Future plans and how to get involved
OProfile

```cpp
#include <iostream>

int factorial1(int n) {
    if (n <= 1)
        return 1;
    else
        return n * factorial1(n-1);
}

unsigned long long factorial2(unsigned n) {
    unsigned long long ret = 1;
    for (unsigned int i = 1; i <= n; i++)
        ret *= i;
    return ret;
}

int main() {
    std::cout << factorial2(10) << std::endl;
    return 0;
}
```
Custom

Time spent in System Calls (ns)

- fstat: 294,273
- mprotect: 77,235
- mmap: 142,015
- open: 133,611
- munmap: 32,977
- access: 20,344
- read: 24,034
- write: 251,502
- Others: 20,946
Custom
Custom
Agenda

- Background
- Today's focus: tools for C/C++ developers built on CDT
- Memory profiling
- Call profiling
- Work areas
- Future plans and how to get involved
Work areas

- User interaction
  - Profile As menu?
- APIs
- Data presentation and visualization
- Functionality
- Integration with source code
Agenda

- Background
- Today's focus: tools for C/C++ developers built on CDT
- Memory profiling
- Call profiling
- Work areas
- Future plans and how to get involved
Future (near / distant)

- SystemTapGui integration
- gprof integration
- LTTng tooling
- Languages other than C/C++
Join us

• We welcome contributors of all forms
  • Plug-in testers
  • Plug-in developers
  • Web designers
  • Documentation authors
  • Graphic designers
  • Commercial adopters
http://www.eclipse.org/linuxtools