

HOW TO MAKE YOUR ZEEK SCRIPTS RUN FASTER

Vern Paxson, Co-Founder and Chief Scientist, Corelight Inc.

Professor of the Graduate School, UC Berkeley

vern@corelight.com

Zeek Workshop | September 10, 2026

Zeek Processing Elements

- Network traffic stream parsing (“event engine”)
 - Work scales w/ # of protocol analyzers
- Script execution
 - Work scales w/ # of packages *and their nature*
- Logging
 - Work scales w/ log volume

Zeek Processing Elements

- Network traffic stream parsing (“event engine”)
 - Work scales w/ # of protocol analyzers
- Script execution
 - Work scales w/ # of packages *and their nature*
- Logging
 - Work scales w/ log volume

Speeding Up Zeek Scripts

1. Using traditional Zeek scripting
 - a) Avoiding repeated expressions
 - b) Comparing perf. with *micro-benchmarks* / *hyperfine*
 - c) Locating perf. issues w/ *zeek --profile-scripts*
2. Using Zeek script optimization
 - a) ZAM
 - b) Using ZAM to produce *flamegraphs*
 - c) Compiling to C++

Speeding Up Zeek Scripts

1. Using traditional Zeek scripting
 - a) Avoiding repeated expressions
 - b) Comparing perf. with *micro-benchmarks* / hyperfine
 - c) Locating perf. issues w/ `zeek --profile-scripts`
2. Using Zeek script optimization
 - a) ZAM
 - b) Using ZAM to produce *flamegraphs*
 - c) Compiling to C++

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    if ( ! f$pe$is_exe )
        return;

    f$pe$os      = os_versions[h$os_version_major, h$os_version_minor];
    f$pe$system  = windows_subsystems[h$system];

    for ( c in h$dll_characteristics )
    {
        if ( dll_characteristics[c] == "DYNAMIC_BASE" )
            f$pe$uses_aslr = T;
        if ( dll_characteristics[c] == "FORCE_INTEGRITY" )
            f$pe$uses_code_integrity = T;
        if ( dll_characteristics[c] == "NX_COMPAT" )
            f$pe$uses_dep = T;
        if ( dll_characteristics[c] == "NO_SEH" )
            f$pe$uses_seh = F;
    }

    f$pe$has_export_table = (|h$table_sizes| > 0 && h$table_sizes[0] > 0);
    f$pe$has_import_table = (|h$table_sizes| > 1 && h$table_sizes[1] > 0);
    f$pe$has_cert_table   = (|h$table_sizes| > 4 && h$table_sizes[4] > 0);
    f$pe$has_debug_data    = (|h$table_sizes| > 6 && h$table_sizes[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    if ( ! f$pe$is_exe )
        return;

    f$pe$os      = os_versions[h$os_version_major, h$os_version_minor];
    f$pe$system  = windows_subsystems[h$system];

    for ( c in h$dll_characteristics )
    {
        if ( dll_characteristics[c] == "DYNAMIC_BASE" )
            f$pe$uses_aslr = T;
        if ( dll_characteristics[c] == "FORCE_INTEGRITY" )
            f$pe$uses_code_integrity = T;
        if ( dll_characteristics[c] == "NX_COMPAT" )
            f$pe$uses_dep = T;
        if ( dll_characteristics[c] == "NO_SEH" )
            f$pe$uses_seh = F;
    }

    f$pe$has_export_table = (|h$table_sizes| > 0 && h$table_sizes[0] > 0);
    f$pe$has_import_table = (|h$table_sizes| > 1 && h$table_sizes[1] > 0);
    f$pe$has_cert_table   = (|h$table_sizes| > 4 && h$table_sizes[4] > 0);
    f$pe$has_debug_data    = (|h$table_sizes| > 6 && h$table_sizes[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    if ( ! f$pe$is_exe )
        return;

    f$pe$os      = os_versions[h$os_version_major, h$os_version_minor];
    f$pe$system  = windows_subsystems[h$system];

    for ( c in h$dll_characteristics )
    {
        if ( dll_characteristics[c] == "DYNAMIC_BASE" )
            f$pe$uses_aslr = T;
        if ( dll_characteristics[c] == "FORCE_INTEGRITY" )
            f$pe$uses_code_integrity = T;
        if ( dll_characteristics[c] == "NX_COMPAT" )
            f$pe$uses_dep = T;
        if ( dll_characteristics[c] == "NO_SEH" )
            f$pe$uses_seh = F;
    }

    f$pe$has_export_table = (h$table_sizes > 0 && h$table_sizes[0] > 0);
    f$pe$has_import_table = (h$table_sizes > 1 && h$table_sizes[1] > 0);
    f$pe$has_cert_table   = (h$table_sizes > 4 && h$table_sizes[4] > 0);
    f$pe$has_debug_data   = (h$table_sizes > 6 && h$table_sizes[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    if ( ! f$pe$is_exe )
        return;

    f$pe$os      = os_versions[h$os_version_major, h$os_version_minor];
    f$pe$system  = windows_subsystems[h$system];

    for ( c in h$dll_characteristics )
    {
        if ( dll_characteristics[c] == "DYNAMIC_BASE" )
            f$pe$uses_aslr = T;
        if ( dll_characteristics[c] == "FORCE_INTEGRITY" )
            f$pe$uses_code_integrity = T;
        if ( dll_characteristics[c] == "NX_COMPAT" )
            f$pe$uses_dep = T;
        if ( dll_characteristics[c] == "NO_SEH" )
            f$pe$uses_seh = F;
    }

    f$pe$has_export_table = (|h$table_sizes| > 0 && h$table_sizes[0] > 0);
    f$pe$has_import_table = (|h$table_sizes| > 1 && h$table_sizes[1] > 0);
    f$pe$has_cert_table   = (|h$table_sizes| > 4 && h$table_sizes[4] > 0);
    f$pe$has_debug_data   = (|h$table_sizes| > 6 && h$table_sizes[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    if ( ! f$pe$is_exe )
        return;

    f$pe$os      = os_versions[h$os_version_major, h$os_version_minor];
    f$pe$system  = windows_subsystems[h$system];

    for ( c in h$dll_characteristics )
    {
        if ( dll_characteristics[c] == "DYNAMIC_BASE" )
            f$pe$uses_aslr = T;
        if ( dll_characteristics[c] == "FORCE_INTEGRITY" )
            f$pe$uses_code_integrity = T;
        if ( dll_characteristics[c] == "NX_COMPAT" )
            f$pe$uses_dep = T;
        if ( dll_characteristics[c] == "NO_SEH" )
            f$pe$uses_seh = F;
    }

    f$pe$has_export_table = (h$table_sizes > 0 && h$table_sizes[0] > 0);
    f$pe$has_import_table = (h$table_sizes > 1 && h$table_sizes[1] > 0);
    f$pe$has_cert_table   = (h$table_sizes > 4 && h$table_sizes[4] > 0);
    f$pe$has_debug_data   = (h$table_sizes > 6 && h$table_sizes[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    local pe = f$pe;
    if ( ! pe$is_exe )
        return;

    pe$os      = os_versions[h$os_version_major, h$os_version_minor];
    pe$ subsystem = windows_subsystems[h$ subsystem];

    for ( c in h$ dll_characteristics )
    {
        if ( dll_characteristics[c] == "DYNAMIC_BASE" )
            pe$uses_aslr = T;
        if ( dll_characteristics[c] == "FORCE_INTEGRITY" )
            pe$uses_code_integrity = T;
        if ( dll_characteristics[c] == "NX_COMPAT" )
            pe$uses_dep = T;
        if ( dll_characteristics[c] == "NO_SEH" )
            pe$uses_seh = F;
    }

    pe$has_export_table = (|h$table_sizes| > 0 && h$table_sizes[0] > 0);
    pe$has_import_table = (|h$table_sizes| > 1 && h$table_sizes[1] > 0);
    pe$has_cert_table   = (|h$table_sizes| > 4 && h$table_sizes[4] > 0);
    pe$has_debug_data   = (|h$table_sizes| > 6 && h$table_sizes[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    local pe = f$pe;
    if ( ! pe$is_exe )
        return;

    pe$os      = os_versions[h$os_version_major, h$os_version_minor];
    pe$subsystem = windows_subsystems[h$subsystem];

    for ( c in h$dll_characteristics )
    {
        if ( dll_characteristics[c] == "DYNAMIC_BASE" )
            pe$uses_aslr = T;
        if ( dll_characteristics[c] == "FORCE_INTEGRITY" )
            pe$uses_code_integrity = T;
        if ( dll_characteristics[c] == "NX_COMPAT" )
            pe$uses_dep = T;
        if ( dll_characteristics[c] == "NO_SEH" )
            pe$uses_seh = F;
    }

    pe$has_export_table = (|h$table_sizes| > 0 && h$table_sizes[0] > 0);
    pe$has_import_table = (|h$table_sizes| > 1 && h$table_sizes[1] > 0);
    pe$has_cert_table   = (|h$table_sizes| > 4 && h$table_sizes[4] > 0);
    pe$has_debug_data   = (|h$table_sizes| > 6 && h$table_sizes[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    local pe = f$pe;
    if ( ! pe$is_exe )
        return;

    pe$os      = os_versions[h$os_version_major, h$os_version_minor];
    pe$ subsystem = windows_subsystems[h$ subsystem];

    for ( c in h$ dll_characteristics )
    {
        if ( dll_characteristics[c] == "DYNAMIC_BASE" )
            pe$uses_aslr = T;
        if ( dll_characteristics[c] == "FORCE_INTEGRITY" )
            pe$uses_code_integrity = T;
        if ( dll_characteristics[c] == "NX_COMPAT" )
            pe$uses_dep = T;
        if ( dll_characteristics[c] == "NO_SEH" )
            pe$uses_seh = F;
    }

    pe$has_export_table = (|h$table_sizes| > 0 && h$table_sizes[0] > 0);
    pe$has_import_table = (|h$table_sizes| > 1 && h$table_sizes[1] > 0);
    pe$has_cert_table   = (|h$table_sizes| > 4 && h$table_sizes[4] > 0);
    pe$has_debug_data   = (|h$table_sizes| > 6 && h$table_sizes[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    local pe = f$pe;
    if ( ! pe$is_exe )
        return;

    pe$os      = os_versions[h$os_version_major, h$os_version_minor];
    pe$ subsystem = windows_subsystems[h$ subsystem];

    for ( c in h$ dll_characteristics )
    {
        if ( dll_characteristics[c] == "DYNAMIC_BASE" )
            pe$uses_aslr = T;
        if ( dll_characteristics[c] == "FORCE_INTEGRITY" )
            pe$uses_code_integrity = T;
        if ( dll_characteristics[c] == "NX_COMPAT" )
            pe$uses_dep = T;
        if ( dll_characteristics[c] == "NO_SEH" )
            pe$uses_seh = F;
    }

    pe$has_export_table = (|h$table_sizes| > 0 && h$table_sizes[0] > 0);
    pe$has_import_table = (|h$table_sizes| > 1 && h$table_sizes[1] > 0);
    pe$has_cert_table    = (|h$table_sizes| > 4 && h$table_sizes[4] > 0);
    pe$has_debug_data    = (|h$table_sizes| > 6 && h$table_sizes[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    local pe = f$pe;
    if ( ! pe$is_exe )
        return;

    pe$os      = os_versions[h$os_version_major, h$os_version_minor];
    pe$ subsystem = windows_subsystems[h$ subsystem];

    for ( c in h$ dll_characteristics )
    {
        if ( dll_characteristics[c] == "DYNAMIC_BASE" )
            pe$uses_aslr = T;
        if ( dll_characteristics[c] == "FORCE_INTEGRITY" )
            pe$uses_code_integrity = T;
        if ( dll_characteristics[c] == "NX_COMPAT" )
            pe$uses_dep = T;
        if ( dll_characteristics[c] == "NO_SEH" )
            pe$uses_seh = F;
    }

    local ts = h$ table_sizes;
    pe$has_export_table = (|ts| > 0 && ts[0] > 0);
    pe$has_import_table = (|ts| > 1 && ts[1] > 0);
    pe$has_cert_table   = (|ts| > 4 && ts[4] > 0);
    pe$has_debug_data   = (|ts| > 6 && ts[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    local pe = f$pe;
    if ( ! pe$is_exe )
        return;

    pe$os          = os_versions[h$os_version_major, h$os_version_minor];
    pe$ subsystem = windows_subsystems[h$ subsystem];

    for ( c in h$dll_characteristics )
    {
        if ( dll_characteristics[c] == "DYNAMIC_BASE" )
            pe$uses_aslr = T;
        if ( dll_characteristics[c] == "FORCE_INTEGRITY" )
            pe$uses_code_integrity = T;
        if ( dll_characteristics[c] == "NX_COMPAT" )
            pe$uses_dep = T;
        if ( dll_characteristics[c] == "NO_SEH" )
            pe$uses_seh = F;
    }

    local ts = h$table_sizes;
    pe$has_export_table = (|ts| > 0 && ts[0] > 0);
    pe$has_import_table = (|ts| > 1 && ts[1] > 0);
    pe$has_cert_table    = (|ts| > 4 && ts[4] > 0);
    pe$has_debug_data    = (|ts| > 6 && ts[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    local pe = f$pe;
    if ( ! pe$is_exe )
        return;

    pe$os      = os_versions[h$os_version_major, h$os_version_minor];
    pe$ subsystem = windows_subsystems[h$ subsystem];

    for ( c in h$dll_characteristics )
    {
        if ( dll_characteristics[c] == "DYNAMIC_BASE" )
            pe$uses_aslr = T;
        if ( dll_characteristics[c] == "FORCE_INTEGRITY" )
            pe$uses_code_integrity = T;
        if ( dll_characteristics[c] == "NX_COMPAT" )
            pe$uses_dep = T;
        if ( dll_characteristics[c] == "NO_SEH" )
            pe$uses_seh = F;
    }

    local ts = h$table_sizes;
    pe$has_export_table = (|ts| > 0 && ts[0] > 0);
    pe$has_import_table = (|ts| > 1 && ts[1] > 0);
    pe$has_cert_table    = (|ts| > 4 && ts[4] > 0);
    pe$has_debug_data    = (|ts| > 6 && ts[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    local pe = f$pe;
    if ( ! pe$is_exe )
        return;

    pe$os      = os_versions[h$os_version_major, h$os_version_minor];
    pe$ subsystem = windows_subsystems[h$ subsystem];

    for ( c in h$dll_characteristics )
    {
        if ( dll_characteristics[c] == "DYNAMIC_BASE" )
            pe$uses_aslr = T;
        if ( dll_characteristics[c] == "FORCE_INTEGRITY" )
            pe$uses_code_integrity = T;
        if ( dll_characteristics[c] == "NX_COMPAT" )
            pe$uses_dep = T;
        if ( dll_characteristics[c] == "NO_SEH" )
            pe$uses_seh = F;
    }

    local ts = h$table_sizes;
    pe$has_export_table = (|ts| > 0 && ts[0] > 0);
    pe$has_import_table = (|ts| > 1 && ts[1] > 0);
    pe$has_cert_table   = (|ts| > 4 && ts[4] > 0);
    pe$has_debug_data   = (|ts| > 6 && ts[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    local pe = f$pe;
    if ( ! pe$is_exe )
        return;

    pe$os      = os_versions[h$os_version_major, h$os_version_minor];
    pe$ subsystem = windows_subsystems[h$ subsystem];

    for ( c in h$dll_characteristics )
    {
        local dll_c = dll_characteristics[c];
        if ( dll_c == "DYNAMIC_BASE" )
            pe$uses_aslr = T;
        if ( dll_c == "FORCE_INTEGRITY" )
            pe$uses_code_integrity = T;
        if ( dll_c == "NX_COMPAT" )
            pe$uses_dep = T;
        if ( dll_c == "NO_SEH" )
            pe$uses_seh = F;
    }

    local ts = h$table_sizes;
    pe$has_export_table = (|ts| > 0 && ts[0] > 0);
    pe$has_import_table = (|ts| > 1 && ts[1] > 0);
    pe$has_cert_table   = (|ts| > 4 && ts[4] > 0);
    pe$has_debug_data   = (|ts| > 6 && ts[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    local pe = f$pe;
    if ( ! pe$is_exe )
        return;

    pe$os      = os_versions[h$os_version_major, h$os_version_minor];
    pe$ subsystem = windows_subsystems[h$ subsystem];

    for ( c in h$dll_characteristics )
    {
        local dll_c = dll_characteristics[c];
        if ( dll_c == "DYNAMIC_BASE" )
            pe$uses_aslr = T;
        if ( dll_c == "FORCE_INTEGRITY" )
            pe$uses_code_integrity = T;
        if ( dll_c == "NX_COMPAT" )
            pe$uses_dep = T;
        if ( dll_c == "NO_SEH" )
            pe$uses_seh = F;
    }

    local ts = h$table_sizes;
    pe$has_export_table = (|ts| > 0 && ts[0] > 0);
    pe$has_import_table = (|ts| > 1 && ts[1] > 0);
    pe$has_cert_table    = (|ts| > 4 && ts[4] > 0);
    pe$has_debug_data    = (|ts| > 6 && ts[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    local pe = f$pe;
    if ( ! pe$is_exe )
        return;

    pe$os      = os_versions[h$os_version_major, h$os_version_minor];
    pe$ subsystem = windows_subsystems[h$ subsystem];

    for ( c in h$dll_characteristics )
    {
        local dll_c = dll_characteristics[c];
        if ( dll_c == "DYNAMIC_BASE" )
            pe$uses_aslr = T;
        if ( dll_c == "FORCE_INTEGRITY" )
            pe$uses_code_integrity = T;
        if ( dll_c == "NX_COMPAT" )
            pe$uses_dep = T;
        if ( dll_c == "NO_SEH" )
            pe$uses_seh = F;
    }

    local ts = h$table_sizes;
    pe$has_export_table = (|ts| > 0 && ts[0] > 0);
    pe$has_import_table = (|ts| > 1 && ts[1] > 0);
    pe$has_cert_table   = (|ts| > 4 && ts[4] > 0);
    pe$has_debug_data   = (|ts| > 6 && ts[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    local pe = f$pe;
    if ( ! pe$is_exe )
        return;

    pe$os      = os_versions[h$os_version_major, h$os_version_minor];
    pe$system  = windows_subsystems[h$system];

    for ( _, dll_c in h$dll_characteristics )
    {
        if ( dll_c == "DYNAMIC_BASE" )
            pe$uses_aslr = T;
        if ( dll_c == "FORCE_INTEGRITY" )
            pe$uses_code_integrity = T;
        if ( dll_c == "NX_COMPAT" )
            pe$uses_dep = T;
        if ( dll_c == "NO_SEH" )
            pe$uses_seh = F;
    }

    local ts = h$table_sizes;
    pe$has_export_table = (|ts| > 0 && ts[0] > 0);
    pe$has_import_table = (|ts| > 1 && ts[1] > 0);
    pe$has_cert_table   = (|ts| > 4 && ts[4] > 0);
    pe$has_debug_data   = (|ts| > 6 && ts[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    local pe = f$pe;
    if ( ! pe$is_exe )
        return;

    pe$os          = os_versions[h$os_version_major, h$os_version_minor];
    pe$ subsystem = windows_subsystems[h$ subsystem];

    for ( _, dll_c in h$dll_characteristics )
    {
        if ( dll_c == "DYNAMIC_BASE" )
            pe$uses_aslr = T;
        if ( dll_c == "FORCE_INTEGRITY" )
            pe$uses_code_integrity = T;
        if ( dll_c == "NX_COMPAT" )
            pe$uses_dep = T;
        if ( dll_c == "NO_SEH" )
            pe$uses_seh = F;
    }

    local ts = h$table_sizes;
    pe$has_export_table = (|ts| > 0 && ts[0] > 0);
    pe$has_import_table = (|ts| > 1 && ts[1] > 0);
    pe$has_cert_table    = (|ts| > 4 && ts[4] > 0);
    pe$has_debug_data    = (|ts| > 6 && ts[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    local pe = f$pe;
    if ( ! pe$is_exe )
        return;

    pe$os          = os_versions[h$os_version_major, h$os_version_minor];
    pe$ subsystem = windows_subsystems[h$ subsystem];

    for ( _, dll_c in h$dll_characteristics )
    {
        if ( dll_c == "DYNAMIC_BASE" )
            pe$uses_aslr = T;
        if ( dll_c == "FORCE_INTEGRITY" )
            pe$uses_code_integrity = T;
        if ( dll_c == "NX_COMPAT" )
            pe$uses_dep = T;
        if ( dll_c == "NO_SEH" )
            pe$uses_seh = F;
    }

    local ts = h$table_sizes;
    pe$has_export_table = (|ts| > 0 && ts[0] > 0);
    pe$has_import_table = (|ts| > 1 && ts[1] > 0);
    pe$has_cert_table    = (|ts| > 4 && ts[4] > 0);
    pe$has_debug_data    = (|ts| > 6 && ts[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    local pe = f$pe;
    if ( ! pe$is_exe )
        return;

    pe$os          = os_versions[h$os_version_major, h$os_version_minor];
    pe$ subsystem = windows_subsystems[h$ subsystem];

    for ( _, dll_c in h$dll_characteristics )
    {
        if ( dll_c == "DYNAMIC_BASE" )
            pe$uses_aslr = T;
        else if ( dll_c == "FORCE_INTEGRITY" )
            pe$uses_code_integrity = T;
        else if ( dll_c == "NX_COMPAT" )
            pe$uses_dep = T;
        else if ( dll_c == "NO_SEH" )
            pe$uses_seh = F;
    }

    local ts = h$table_sizes;
    pe$has_export_table = (|ts| > 0 && ts[0] > 0);
    pe$has_import_table = (|ts| > 1 && ts[1] > 0);
    pe$has_cert_table    = (|ts| > 4 && ts[4] > 0);
    pe$has_debug_data    = (|ts| > 6 && ts[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    local pe = f$pe;
    if ( ! pe$is_exe )
        return;

    pe$os      = os_versions[h$os_version_major, h$os_version_minor];
    pe$system  = windows_subsystems[h$system];

    for ( _, dll_c in h$dll_characteristics )
    {
        if ( dll_c == "DYNAMIC_BASE" )
            pe$uses_aslr = T;
        else if ( dll_c == "FORCE_INTEGRITY" )
            pe$uses_code_integrity = T;
        else if ( dll_c == "NX_COMPAT" )
            pe$uses_dep = T;
        else if ( dll_c == "NO_SEH" )
            pe$uses_seh = F;
    }

    local ts = h$table_sizes;
    pe$has_export_table = (|ts| > 0 && ts[0] > 0);
    pe$has_import_table = (|ts| > 1 && ts[1] > 0);
    pe$has_cert_table    = (|ts| > 4 && ts[4] > 0);
    pe$has_debug_data    = (|ts| > 6 && ts[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    local pe = f$pe;
    if ( ! pe$is_exe )
        return;

    pe$os          = os_versions[h$os_version_major, h$os_version_minor];
    pe$ subsystem = windows_subsystems[h$ subsystem];

    for ( _, dll_c in h$dll_characteristics )
    {
        if ( dll_c == "DYNAMIC_BASE" )
            pe$uses_aslr = T;
        else if ( dll_c == "FORCE_INTEGRITY" )
            pe$uses_code_integrity = T;
        else if ( dll_c == "NX_COMPAT" )
            pe$uses_dep = T;
        else if ( dll_c == "NO_SEH" )
            pe$uses_seh = F;
    }

    local ts = h$table_sizes;
    pe$has_export_table = (|ts| > 0 && ts[0] > 0);
    pe$has_import_table = (|ts| > 1 && ts[1] > 0);
    pe$has_cert_table    = (|ts| > 4 && ts[4] > 0);
    pe$has_debug_data    = (|ts| > 6 && ts[6] > 0);
}

```

```

event pe_optional_header(f: fa_file, h: PE::OptionalHeader) &priority=5
{
    hook set_file(f);

    # Only EXEs have optional headers
    local pe = f$pe;
    if ( ! pe$is_exe )
        return;

    pe$os      = os_versions[h$os_version_major, h$os_version_minor];
    pe$ subsystem = windows_subsystems[h$ subsystem];

    for ( _, dll_c in h$dll_characteristics )
    {
        if ( dll_c == "DYNAMIC_BASE" )
            pe$uses_aslr = T;
        else if ( dll_c == "FORCE_INTEGRITY" )
            pe$uses_code_integrity = T;
        else if ( dll_c == "NX_COMPAT" )
            pe$uses_dep = T;
        else if ( dll_c == "NO_SEH" )
            pe$uses_seh = F;
    }

    local ts = h$table_sizes;
    local n = |ts|;
    pe$has_export_table = (n > 0 && ts[0] > 0);
    pe$has_import_table = (n > 1 && ts[1] > 0);
    pe$has_cert_table    = (n > 4 && ts[4] > 0);
    pe$has_debug_data    = (n > 6 && ts[6] > 0);
}

```

In total
23+ evaluations saved

Speeding Up Zeek Scripts

1. Using traditional Zeek scripting
 - a) Avoiding repeated expressions
 - b) Comparing perf. with *micro-benchmarks* / *hyperfine*
 - c) Locating perf. issues w/ `zeek --profile-scripts`
2. Using Zeek script optimization
 - a) ZAM
 - b) Using ZAM to produce *flamegraphs*
 - c) Compiling to C++

```
perf-examples % cat record-ref1.zee
```

```
type r1: record { a: count; };
```

```
type r2: record { b: r1; };
```

```
event zee_init()
```

```
{
```

```
  local x = r2($b = r1($a = 0));
```

```
  local sum = +0; local i = 0;
```

```
  while ( ++i <= 1000000 )
```

```
  {
```

```
    if ( x$b$a > 10 ) sum += 10;
```

```
    else if ( x$b$a > 5 ) sum += 5;
```

```
    else if ( x$b$a > 3 ) sum += 3;
```

```
    else if ( x$b$a > 1 ) ++sum;
```

```
    else --sum;
```

```
  }
```

```
  print sum;
```

```
}
```

Small script that does
8 record references
(along with other stuff)
a million times

```
perf-examples % cat record-ref1.zee
```

```
type r1: record { a: count; };  
type r2: record { b: r1; };
```

```
event zeek_init()  
{  
  local x = r2($b = r1($a = 0));  
  local sum = +0; local i = 0;
```

```
  while ( ++i <= 1000000 )  
  {  
    if ( x$b$a > 10 )  
    else if ( x$b$a >  
    else if ( x$b$a >  
    else if ( x$b$a >  
    else --sum;  
  }
```

```
  print sum;  
}
```

```
perf-examples % cat record-ref2.zee
```

```
type r1: record { a: count; };  
type r2: record { b: r1; };
```

```
event zeek_init()  
{  
  local x = r2($b = r1($a = 0));  
  local sum = +0; local i = 0;
```

```
  while ( ++i <= 1000000 )  
  {  
    local y = x$b$a;  
    if ( y > 10 ) sum += 10;  
    else if ( y > 5 ) sum += 5;  
    else if ( y > 3 ) sum += 3;  
    else if ( y > 1 ) ++sum;  
    else --sum;  
  }
```

```
  print sum;  
}
```

Same but using a local
variable (y) to reduce to
2 record references

```
perf-examples % hyperfine 'zeek record-ref1.zeek' 'zeek record-ref2.zeek'
```

hyperfine takes a list
of programs to run,
repeatedly times them
& compares

```
perf-examples % hyperfine 'zeek record-ref1.zeek' 'zeek record-ref2.zeek'
```

Benchmark 1: zeek record-ref1.zeek

Time (mean $\pm$ σ): 2.953 s $\pm$ 0.055 s [User: 2.802 s, System: 0.145 s]
Range (min ... max): 2.910 s ... 3.094 s 10 runs

Benchmark 2: zeek record-ref2.zeek

Time (mean $\pm$ σ): 2.704 s $\pm$ 0.014 s [User: 2.553 s, System: 0.144 s]
Range (min ... max): 2.688 s ... 2.736 s 10 runs

Summary

zeek record-ref2.zeek ran

1.09 $\pm$ 0.02 times faster than zeek record-ref1.zeek

For this
micro-benchmark,
using the local is about
9% faster

Speeding Up Zeek Scripts

1. Using traditional Zeek scripting
 - a) Avoiding repeated expressions
 - b) Comparing perf. with *micro-benchmarks* / hyperfine
 - c) Locating perf. issues w/ `zeek --profile-scripts`
2. Using Zeek script optimization
 - a) ZAM
 - b) Using ZAM to produce *flamegraphs*
 - c) Compiling to C++

zeek ... -profile-scripts >prof.out

Hard to make out due to long file names,
let's get rid of them

```
#fields function location type ncall tot_CPU child_CPU tot_Mem child_Mem
#types string string string count interval interval count count
__init /Users/vern/zeek/zlocal/share/zeek/base/init-bare.zeek:04 0
have_spicy_analyzers /Users/vern/zeek/zlocal/share/zeek/base/init-bare.zeek:6850 BiF 3 0.000004 0.000000 0 0
getenv /Users/vern/zeek/zlocal/share/zeek/base/init-bare.zeek:6850 BiF 3 0.000006 0.000000 0 0
Cluster::local_node_type /Users/vern/zeek/zlocal/share/zeek/base/frameworks/cluster/main.zeek:318 function 34 0.000057 0.000000 0 0
__init_secondary_bifs /Users/vern/zeek/zlocal/share/zeek/base/init-bare.zeek:6850 BiF 1 0.000402 0.000000 81920 0
double_to_time /Users/vern/zeek/zlocal/share/zeek/base/init-bare.zeek:6850 BiF 1 0.000025 0.000000 16384 0
zeek_version /Users/vern/zeek/zlocal/share/zeek/base/init-bare.zeek:6850 BiF 1 0.000020 0.000000 16384 0
Version::parse /Users/vern/zeek/zlocal/share/zeek/base/misc/version.zeek:56 function 1 0.000319 0.000148 114688 65536
split_string1 /Users/vern/zeek/zlocal/share/zeek/base/init-bare.zeek:6850 BiF 3 0.000076 0.000000 49152 0
lstrip /Users/vern/zeek/zlocal/share/zeek/base/init-bare.zeek:6850 BiF 6 0.000011 0.000000 0 0
gsub /Users/vern/zeek/zlocal/share/zeek/base/init-bare.zeek:6850 BiF 2 0.000057 0.000000 16384 0
find_last /Users/vern/zeek/zlocal/share/zeek/base/init-bare.zeek:6850 BiF 1 0.000004 0.000000 0 0
Telemetry::register_counter_family /Users/vern/zeek/zlocal/share/zeek/base/frameworks/telemetry/main.zeek:276 function 1 0.000060 0.000054 16384 16384
Telemetry::__counter_family /Users/vern/zeek/zlocal/share/zeek/base/init-bare.zeek:6850 BiF 1 0.000054 0.000000 16384 0
Telemetry::register_gauge_family /Users/vern/zeek/zlocal/share/zeek/base/frameworks/telemetry/main.zeek:335 function 2 0.000040 0.000028 16384 16384
Telemetry::__gauge_family /Users/vern/zeek/zlocal/share/zeek/base/init-bare.zeek:6850 BiF 2 0.000028 0.000000 16384 0
Telemetry::register_histogram_family /Users/vern/zeek/zlocal/share/zeek/base/frameworks/telemetry/main.zeek:403 function 1 0.000026 0.000021 16384 16384
Telemetry::__histogram_family /Users/vern/zeek/zlocal/share/zeek/base/init-bare.zeek:6850 BiF 1 0.000021 0.000000 16384 0
have_spicy_analyzers /Users/vern/zeek/zlocal/share/zeek/base/init-bare.zeek:6850 BiF 5 0.000010 0.000000 0 0
Cluster::local_node_type /Users/vern/zeek/zlocal/share/zeek/base/frameworks/cluster/main.zeek:323 function 9 0.000037 0.000012 0 0
FilteredTraceDetection::should_detect /Users/vern/zeek/zlocal/share/zeek/base/misc/find-filtered-trace.zeek:16 function 1 0.000022 0.000008 0 0
zeek_args /Users/vern/zeek/zlocal/share/zeek/base/init-bare.zeek:6850 BiF 1 0.000008 0.000000 0 0
cat /Users/vern/zeek/zlocal/share/zeek/base/init-bare.zeek:6850 BiF 6 0.000046 0.000000 16384 0
zeek_init /Users/vern/zeek/zlocal/share/zeek/base/frameworks/sumstats/main.zeek:307 event 1 0.000161 0.000132 16384 16384
SumStats::register_observe_plugins /Users/vern/zeek/zlocal/share/zeek/base/frameworks/sumstats/plugins/average.zeek:19 hook 1 0.000032 0.000014 0 0
SumStats::register_observe_plugin /Users/vern/zeek/zlocal/share/zeek/base/frameworks/sumstats/main.zeek:295 function 11 0.000039 0.000000 0 0
SumStats::register_observe_plugins /Users/vern/zeek/zlocal/share/zeek/base/frameworks/sumstats/plugins/hll_unique.zeek:39 hook 1 0.000009 0.000004 0 0
SumStats::register_observe_plugins /Users/vern/zeek/zlocal/share/zeek/base/frameworks/sumstats/plugins/last.zeek:50 hook 1 0.000006 0.000003 0 0
SumStats::register_observe_plugins /Users/vern/zeek/zlocal/share/zeek/base/frameworks/sumstats/plugins/max.zeek:19 hook 1 0.000007 0.000002 0 0
SumStats::register_observe_plugins /Users/vern/zeek/zlocal/share/zeek/base/frameworks/sumstats/plugins/variance.zeek:33 hook 1 0.000025 0.000019 0 0
PacketAnalyzer::register_packet_analyzer /Users/vern/zeek/zlocal/share/zeek/base/init-bare.zeek:6850 BiF 114 0.002361 0.000000 4866048 0
SumStats::add_observe_plugin_dependency /Users/vern/zeek/zlocal/share/zeek/base/frameworks/sumstats/main.zeek:300 function 2 0.000021 0.000000 0 0
SumStats::register_observe_plugins /Users/vern/zeek/zlocal/share/zeek/base/frameworks/sumstats/plugins/std-dev.zeek:26 hook 1 0.000013 0.000007 0 0
zeek_init /Users/vern/zeek/zlocal/share/zeek/base/packet-protocols/root/main.zeek:16 event 1 0.000040 0.000023 0 0
Option::set_change_handler /Users/vern/zeek/zlocal/share/zeek/base/init-bare.zeek:6850 BiF 136 0.000338 0.000000 0 0
```

#fields	function	location	type	ncall	tot_CPU	child_CPU	tot_Mem	child_Mem		
#types	string	string	string	count	interval	interval	count	count		
__init_primary_bifs	<no-location>	BiF	1	0.000704	0.000000	98304	0			
have_spicy	init-bare.zeek:6850	BiF	3	0.000004	0.000000	0	0			
getenv	init-bare.zeek:6850	BiF	3	0.000006	0.000000	0	0			
Cluster::is_enabled	frameworks/cluster/main.zeek:318	function	34	0.000057	0.000000	0	0			
__init_secondary_bifs	init-bare.zeek:6850	BiF	1	0.000402	0.000000	81920	0			
double_to_time	init-bare.zeek:6850	BiF	1	0.000025	0.000000	16384	0			
zeek_version	init-bare.zeek:6850	BiF	1	0.000020	0.000000	16384	0			
Version::parse	misc/version.zeek:56	function	1	0.000319	0.000148	114688	65536			
split_string1	init-bare.zeek:6850	BiF	3	0.000076	0.000000	49152	0			
lstrip	init-bare.zeek:6850	BiF	6	0.000011	0.000000	0	0			
gsub	init-bare.zeek:6850	BiF	2	0.000057	0.000000	16384	0			
find_last	init-bare.zeek:6850	BiF	1	0.000004	0.000000	0	0			
Telemetry::register_counter_family	frameworks/telemetry/main.zeek:276	function	1	0.000060	0.000054	16384	16384			
Telemetry::__counter_family	init-bare.zeek:6850	BiF	1	0.000054	0.000000	16384	0			
Telemetry::register_gauge_family	frameworks/telemetry/main.zeek:335	function	2	0.000040	0.000028	16384	16384			
Telemetry::__gauge_family	init-bare.zeek:6850	BiF	2	0.000028	0.000000	16384	0			
have_spicy_analyzers	init-bare.zeek:6850	BiF	5	0.000010	0.000000	0	0			
Cluster::local_node_type	frameworks/cluster/main.zeek:323	function	9	0.000037	0.000012	0	0			
FilteredTraceDetection::should_detect	misc/find-filtered-trace.zeek:16	function	1	0.000022	0.000008	0	0			
zeek_args	init-bare.zeek:6850	BiF	1	0.000008	0.000000	0	0			
cat	init-bare.zeek:6850	BiF	6	0.000046	0.000000	16384	0			
zeek_init	frameworks/sumstats/main.zeek:307	event	1	0.000161	0.000132	16384	16384			
SumStats::register_observe_plugins	frameworks/sumstats/plugins/average.zeek:19	hook	1	0.000032	0.000014	0	0			
file_state_remove	protocols/smb/main.zeek:249	event	3619	0.004859	0.000000	0	0			
file_state_remove	files/pe/main.zeek:134	event	3619	0.004395	0.000000	0	0			
file_state_remove	frameworks/files/main.zeek:583	event	3619	0.066363	0.032818	0	0			
Files::log_policy	files/x509/main.zeek:199	hook	3619	0.009150	0.000000	0	0			
ssl_established	protocols/ssl/main.zeek:506	event	3788	0.063466	0.044178	0	0			
SSL::ssl_finishing	protocols/ssl/files.zeek:161	hook	3792	0.039774	0.000000	0	0			
ssl_established	protocols/ssl/main.zeek:501	event	3788	0.063466	0.044178	0	0			
ssl_established	protocols/ftp/gridftp.zeek:99	event	3788	0.063466	0.044178	0	0			
ssl_established	protocols/ftp/gridftp.zeek:115	event	3788	0.063466	0.044178	0	0			
GridFTP::data_channel_initial_criteria	protocols/ftp/gridftp.zeek:108	function	3788	0.009557	0.000000	0	0			
ssl_established	protocols/ssl/main.zeek:512	event	3788	0.103337	0.092747	163840	163840			
SSL::finish	protocols/ssl/main.zeek:267	function	7580	0.109984	0.074076	163840	163840			
SSL::log_record	protocols/ssl/main.zeek:240	function	7580	0.061827	0.041825	163840	163840			

Header akin to Zeek TSV logs

Unsorted, by default

#fields	function	location	type	ncall	tot_CPU	child_CPU	tot_Mem	child_Mem		
#types	string	string	string	count	interval	interval	count	count		
non-scripts	<no-location>	TOTAL	258260	8.933789		0.000000	133365760	0		
total	322-locations	TOTAL	639456	4.048020		0.000000	15335424	0		
file_sniff	6-locations	event	21714	0.472209		0.000000	425984	0		
file_sniff	frameworks/files/main.zee:554	event	3619	0.417712		0.339639		114688	114688	
all-BiFs	65-locations	BiF	130510	0.348761		0.000000	6307840	0		
Files::add_analyzer	frameworks/files/main.zee:415	function	14476	0.321759		0.193404		114688	49152	
ssl_handshake_message	protocols/ssl/main.zee:376	event	22057	0.287833		0.156010	0	0		
connection_state_remove	3-locations	event	11379	0.220437		0.000000	81920	0		
ssl_established	5-locations	event	18940	0.196962		0.000000	163840	0		
SSL::add_to_history	protocols/ssl/main.zee:213	function	29291	0.175915		0.036123	0	0		
Conn::set_conn	protocols/conn/main.zee:244	function	7586	0.168133		0.059701	147456	16384		
file_state_remove	6-locations	event	21714	0.150583		0.000000	163840	0		
analyzer_confirmation_info	6-locations	event	22752	0.135853		0.000000	114688	0		
ssl_plaintext_data	protocols/ssl/main.zee:538	event	29291	0.128121		0.030683	0	0		
SSL::set_session	protocols/ssl/main.zee:204	function	79429	0.126430		0.016436	65536	0		
Log::write	frameworks/logging/main.zee:1003	function	11277	0.119019		0.088300	458752	458752		
connection_state_remove	protocols/conn/main.zee:329	event	3793	0.115901		0.103771	0	0		
SSL::finish	protocols/ssl/main.zee:267	function	7580	0.109984		0.074076	163840	163840		
ssl_established	protocols/ssl/main.zee:512	event	3788	0.103337		0.092747	163840	163840		
file_over_new_connection	9-locations	event	32571	0.093994		0.000000	0	0		
ssl_change_cipher_spec	protocols/ssl/main.zee:478	event	7234	0.091254		0.052987	0	0		
Log::__write	init-frameworks-and-bifs.zee:32	BiF	11277	0.088300		0.009150	458752	0		
file_new	frameworks/files/main.zee:534	event	3619	0.085667		0.059141	32768	32768		
Files::__add_analyzer	init-frameworks-and-bifs.zee:32	BiF	14476	0.083714		0.000000	32768	0		
Files::set_info	frameworks/files/main.zee:347	function	14476	0.081698		0.000000	16384	0		
new_connection	2-locations	event	7586	0.080705		0.000000	147456	0		
new_connection	protocols/conn/main.zee:314	event	3793	0.076163		0.064362	147456	147456		
file_state_remove	frameworks/files/main.zee:583	event	3619	0.066363		0.032818	0	0		
ssl_established	protocols/ssl/main.zee:506	event	3788	0.063466		0.044178	0	0		
SSL::log_record	protocols/ssl/main.zee:240	function	7580	0.061827		0.041825	163840	163840		
Files::analyzer_enabled	frameworks/files/main.zee:410	function	14476	0.056661		0.021002	0	0		
analyzer_confirmation_info	protocols/ssl/main.zee:529	event	3792	0.056280		0.041062	65536	65536		
file_hash	2-locations	event	21714	0.055546		0.000000	0	0		
analyzer_confirmation_info	frameworks/analyzer/dpd.zee:31	event	3792	0.055467		0.020920	49152	0		
connection_state_remove	protocols/conn/main.zee:334	event	3793	0.054818		0.043149	81920	81920		
Files::analyzer_name	frameworks/files/main.zee:448	function	14476	0.053029		0.017845	16384	16384		
connection_state_remove	protocols/conn/removal-hooks.zee:74	event	3793	0.049718		0.030939	0	0		

```
sort -nr -k5
most CPU first
```

#fields	function	location	type	ncall	tot_CPU	child_CPU	tot_Mem	child_Mem
#types	string	string	count	interval	interval	count	count	
non-scripts	<no-location>	TOTAL	258260	8.933789	0.000000	133365760	0	
total	322-locations	TOTAL	639456	4.048020	0.000000	15335424	0	
file_sniff	6-locations	event	21714	0.472209	0.000000	425984	0	
file_sniff	frameworks/files/main.zee:554	event	3619	0.417712	0.339639	114688	114688	
all-BiFs			48761	0.000000	6307840	0		
Files::add			415	function	14476	0.321759	0.193404	114688 49152
ssl_handsh				event	22057	0.287833	0.156010	0 0
connection			11379	0.220437	0.000000	81920	0	
ssl_establ			40	0.196962	0.000000	163840	0	
SSL::add_to_history	protocols/ssl/main.zee:215	function			29291	0.175915	0.036123	0 0
Conn::set_conn	protocols/conn/main.zee:244	function			7586	0.168133	0.059701	147456 16384
file_state_remove	6-locations	event	21714	0.150583	0.000000	163840	0	
analyzer_confirmation_info	6-locations	event	22752	0.135853	0.000000	114688	0	
ssl_plaintext_data	protocols/ssl/main.zee:538	event	29291	0.128121	0.030683	0	0	
SSL::set_session	protocols/ssl/main.zee:204	function			79429	0.126430	0.016436	65536 0
Log::write	frameworks/logging/main.zee:1003	function			11277	0.119019	0.088300	458752 458752
connection_state_remove	protocols/conn/main.zee:329	event	3793	0.115901	0.103771	0	0	
SSL::finish	protocols/ssl/main.zee:267	function			7580	0.109984	0.074076	163840 163840
ssl_established	protocols/ssl/main.zee:512	event	3788	0.103337	0.092747	163840	163840	
file_over_new_connection	9-locations	event	32571	0.093994	0.000000	0	0	
ssl_change_cipher_spec	protocols/ssl/main.zee:478	event	7234	0.091254	0.052987	0	0	
Log::__write	init-frameworks-and-bifs.zee:32	BiF	11277	0.088300	0.009150	458752	0	
file_new	frameworks/files/main.zee:534	event	3619	0.085667	0.059141	32768	32768	
Files::__add_analyzer	init-frameworks-and-bifs.zee:32	BiF			14476	0.083714	0.000000	32768 0
Files::set_info	frameworks/files/main.zee:347	function			14476	0.081698	0.000000	16384 0
new_connection	2-locations	event	7586	0.080705	0.000000	147456	0	
new_connection	protocols/conn/main.zee:314	event	3793	0.076163	0.064362	147456	147456	
file_state_remove	frameworks/files/main.zee:583	event	3619	0.066363	0.032818	0	0	
ssl_established	protocols/ssl/main.zee:506	event	3788	0.063466	0.044178	0	0	
SSL::log_record	protocols/ssl/main.zee:240	function			7580	0.061827	0.041825	163840 163840
Files::analyzer_enabled	frameworks/files/main.zee:410	function			14476	0.056661	0.021002	0 0
analyzer_confirmation_info	protocols/ssl/main.zee:529	event	3792	0.056280	0.041062	65536	65536	
file_hash	2-locations	event	21714	0.055546	0.000000	0	0	
analyzer_confirmation_info	frameworks/analyzer/dpd.zee:31	event	3792	0.055467	0.020920	49152	0	
connection_state_remove	protocols/conn/main.zee:334	event	3793	0.054818	0.043149	81920	81920	
Files::analyzer_name	frameworks/files/main.zee:448	function			14476	0.053029	0.017845	16384 16384
connection_state_remove	protocols/conn/removal-hooks.zee:74	event	3793	0.049718	0.030939	0	0	

sort -nr -k5
most CPU first

Zeek execution separate
from scripts
(event engine, logging)

#fields	function	location	type	ncall	tot_CPU	child_CPU	tot_Mem	child_Mem
#types	string	string	string	count	interval	interval	count	count
non-scripts	<no-location>	TOTAL	258260	8.933789	0.000000	133365760	0	
total	322-locations	TOTAL	639456	4.048020	0.000000	15335424	0	
file_sniff	6-locations	event	21714	0.472209	0.000000	425984	0	
file_sniff	frameworks/files/main.zeek:554	event	3619	0.417712	0.339639	114688	114688	
all-BiFs	65-locations	BiF	130510	0.348761	0.000000	6307840	0	
Files::add	main.zeek:415	function	14476	0.321759	0.193404	114688	49152	
ssl_handsh	.zeek:376	event	22057	0.287833	0.156010	0	0	
connection_state_remove	3-locations	event	11379	0.220437	0.000000	81920	0	
ssl_established	5-locations	event	18940	0.196962	0.000000	163840	0	
SSL::add_to_history	protocols/ssl/main.zeek:213	function	29291	0.175915	0.036123	0	0	
Conn::set_conn	protocols/conn/main.zeek:244	function	7586	0.168133	0.059701	147456	16384	
file_state_remove	6-locations	event	21714	0.150583	0.000000	163840	0	
analyzer_confirmation_info	6-locations	event	22752	0.135853	0.000000	114688	0	
ssl_plaintext_data	protocols/ssl/main.zeek:538	event	29291	0.128121	0.030683	0	0	
SSL::set_session	protocols/ssl/main.zeek:204	function	79429	0.126430	0.016436	65536	0	
Log::write	frameworks/logging/main.zeek:1003	function	11277	0.119019	0.088300	458752	458752	
connection_state_remove	protocols/conn/main.zeek:329	event	3793	0.115901	0.103771	0	0	
SSL::finish	protocols/ssl/main.zeek:267	function	7580	0.109984	0.074076	163840	163840	
ssl_established	protocols/ssl/main.zeek:512	event	3788	0.103337	0.092747	163840	163840	
file_over_new_connection	9-locations	event	32571	0.093994	0.000000	0	0	
ssl_change_cipher_spec	protocols/ssl/main.zeek:478	event	7234	0.091254	0.052987	0	0	
Log::__write	init-frameworks-and-bifs.zeek:32	BiF	11277	0.088300	0.009150	458752	0	
file_new	frameworks/files/main.zeek:534	event	3619	0.085667	0.059141	32768	32768	
Files::__add_analyzer	init-frameworks-and-bifs.zeek:32	BiF	14476	0.083714	0.000000	32768	0	
Files::set_info	frameworks/files/main.zeek:347	function	14476	0.081698	0.000000	16384	0	
new_connection	2-locations	event	7586	0.080705	0.000000	147456	0	
new_connection	protocols/conn/main.zeek:314	event	3793	0.076163	0.064362	147456	147456	
file_state_remove	frameworks/files/main.zeek:583	event	3619	0.066363	0.032818	0	0	
ssl_established	protocols/ssl/main.zeek:506	event	3788	0.063466	0.044178	0	0	
SSL::log_record	protocols/ssl/main.zeek:240	function	7580	0.061827	0.041825	163840	163840	
Files::analyzer_enabled	frameworks/files/main.zeek:410	function	14476	0.056661	0.021002	0	0	
analyzer_confirmation_info	protocols/ssl/main.zeek:529	event	3792	0.056280	0.041062	65536	65536	
file_hash	2-locations	event	21714	0.055546	0.000000	0	0	
analyzer_confirmation_info	frameworks/analyzer/dpd.zeek:31	event	3792	0.055467	0.020920	49152	0	
connection_state_remove	protocols/conn/main.zeek:334	event	3793	0.054818	0.043149	81920	81920	
Files::analyzer_name	frameworks/files/main.zeek:448	function	14476	0.053029	0.017845	16384	16384	
connection_state_remove	protocols/conn/removal-hooks.zeek:74	event	3793	0.049718	0.030939	0	0	

sort -nr -k5
most CPU first

All scripts together

#fields	function	location	type	ncall	tot_CPU	child_CPU	tot_Mem	child_Mem
#types	string	string	count	interval	interval	count	count	
non-scripts	<no-location>	TOTAL	258260	8.933789	0.000000	133365760	0	
total	322-locations	TOTAL	639456	4.048020	0.000000	15335424	0	
file_sniff	6-locations	event	21714	0.472209	0.000000	425984	0	
file_sniff	frameworks/files/main.zee	event	3619	0.417712	0.339639	114688	114688	
all-BiFs	65-locations	BiF	130510	0.348761	0.000000	6307840	0	
Files::add_analyzer	frameworks/files/main.zee	function	14476	0.321759	0.193404	114688	49152	
ssl_handsh	.zee	event	22057	0.287833	0.156010	0	0	
connection	s	event	11379	0.220437	0.000000	81920	0	
ssl_establ	ent	event	18940	0.196962	0.000000	163840	0	
SSL::add_t	.zee	function	29291	0.175915	0.036123	0	0	
Conn::set	44	function	7586	0.168133	0.059701	147456	16384	
file_state_remove	6-locations	event	21714	0.150583	0.000000	163840	0	
analyzer_confirmation_info	6-locations	event	22752	0.135853	0.000000	114688	0	
ssl_plaintext_data	protocols/ssl/main.zee	event	29291	0.128121	0.030683	0	0	
SSL::set_session	protocols/ssl/main.zee	function	79429	0.126430	0.016436	65536	0	
Log::write	frameworks/logging/main.zee	function	11277	0.119019	0.088300	458752	458752	
connection_state_remove	protocols/conn/main.zee	event	3793	0.115901	0.103771	0	0	
SSL::finish	protocols/ssl/main.zee	function	7580	0.109984	0.074076	163840	163840	
ssl_established	protocols/ssl/main.zee	event	3788	0.103337	0.092747	163840	163840	
file_over_new_connection	9-locations	event	32571	0.093994	0.000000	0	0	
ssl_change_cipher_spec	protocols/ssl/main.zee	event	7234	0.091254	0.052987	0	0	
Log::__write	init-frameworks-and-bifs.zee	BiF	11277	0.088300	0.009150	458752	0	
file_new	frameworks/files/main.zee	event	3619	0.085667	0.059141	32768	32768	
Files::__add_analyzer	init-frameworks-and-bifs.zee	BiF	14476	0.083714	0.000000	32768	0	
Files::set_info	frameworks/files/main.zee	function	14476	0.081698	0.000000	16384	0	
new_connection	2-locations	event	7586	0.080705	0.000000	147456	0	
new_connection	protocols/conn/main.zee	event	3793	0.076163	0.064362	147456	147456	
file_state_remove	frameworks/files/main.zee	event	3619	0.066363	0.032818	0	0	
ssl_established	protocols/ssl/main.zee	event	3788	0.063466	0.044178	0	0	
SSL::log_record	protocols/ssl/main.zee	function	7580	0.061827	0.041825	163840	163840	
Files::analyzer_enabled	frameworks/files/main.zee	function	14476	0.056661	0.021002	0	0	
analyzer_confirmation_info	protocols/ssl/main.zee	event	3792	0.056280	0.041062	65536	65536	
file_hash	2-locations	event	21714	0.055546	0.000000	0	0	
analyzer_confirmation_info	frameworks/analyzer/dpd.zee	event	3792	0.055467	0.020920	49152	0	
connection_state_remove	protocols/conn/main.zee	event	3793	0.054818	0.043149	81920	81920	
Files::analyzer_name	frameworks/files/main.zee	function	14476	0.053029	0.017845	16384	16384	
connection_state_remove	protocols/conn/removal-hooks.zee	event	3793	0.049718	0.030939	0	0	

sort -nr -k5
most CPU first

All 6 of the
file_sniff event
handlers

#fields	function	location	type	ncall	tot_CPU	child_CPU	tot_Mem	child_Mem	
#types	string	string	count	interval	interval	count	count		
non-scripts	<no-location>	TOTAL	258260	8.933789	0.000000	133365760	0		
total	322-locations	TOTAL	639456	4.048020	0.000000	15335424	0		
file_sniff	6-locations	event	21714	0.472209	0.000000	425984	0		
file_sniff	frameworks/files/main.zeeb:554	event	3619	0.417712	0.339639	114688	114688		
all-BiFs	65-locations	BiF	130510	0.348761	0.000000	6307840	0		
Files::add_analyzer	frameworks/files/main.zeeb:415	function	14476	0.321759	0.193404	114688	49152		
ssl_handshake	protocols/ssl/main.zeeb:376	event	22057	0.287833	0.156010	0	0		
connect	s	event	11379	0.220437	0.000000	81920	0		
ssl_est	ent	18940	0.196962	0.000000	163840	0			
SSL::ac	.zeeb:213	function	29291	0.175915	0.036123	0	0		
Conn::s	44	function	7586	0.168133	0.059701	147456	16384		
file_st	ent	21714	0.150583	0.000000	163840	0			
analyzer_confirmation_info	6-locations	event	22752	0.135853	0.000000	114688	0		
ssl_plaintext_data	protocols/ssl/main.zeeb:538	event	29291	0.128121	0.030683	0	0		
SSL::set_session	protocols/ssl/main.zeeb:204	function	79429	0.126430	0.016436	65536	0		
Log::write	frameworks/logging/main.zeeb:1003	function	11277	0.119019	0.088300	458752	458752		
connection_state_remove	protocols/conn/main.zeeb:329	event	3793	0.115901	0.103771	0	0		
SSL::finish	protocols/ssl/main.zeeb:267	function	7580	0.109984	0.074076	163840	163840		
ssl_established	protocols/ssl/main.zeeb:512	event	3788	0.103337	0.092747	163840	163840		
file_over_new_connection	9-locations	event	32571	0.093994	0.000000	0	0		
ssl_change_cipher_spec	protocols/ssl/main.zeeb:478	event	7234	0.091254	0.052987	0	0		
Log::__write	init-frameworks-and-bifs.zeeb:32	BiF	11277	0.088300	0.009150	458752	0		
file_new	frameworks/files/main.zeeb:534	event	3619	0.085667	0.059141	32768	32768		
Files::__add_analyzer	init-frameworks-and-bifs.zeeb:32	BiF	14476	0.083714	0.000000	32768	0		
Files::set_info	frameworks/files/main.zeeb:347	function	14476	0.081698	0.000000	16384	0		
new_connection	2-locations	event	7586	0.080705	0.000000	147456	0		
new_connection	protocols/conn/main.zeeb:314	event	3793	0.076163	0.064362	147456	147456		
file_state_remove	frameworks/files/main.zeeb:583	event	3619	0.066363	0.032818	0	0		
ssl_established	protocols/ssl/main.zeeb:506	event	3788	0.063466	0.044178	0	0		
SSL::log_record	protocols/ssl/main.zeeb:240	function	7580	0.061827	0.041825	163840	163840		
Files::analyzer_enabled	frameworks/files/main.zeeb:410	function	14476	0.056661	0.021002	0	0		
analyzer_confirmation_info	protocols/ssl/main.zeeb:529	event	3792	0.056280	0.041062	65536	65536		
file_hash	2-locations	event	21714	0.055546	0.000000	0	0		
analyzer_confirmation_info	frameworks/analyzer/dpd.zeeb:31	event	3792	0.055467	0.020920	49152	0		
connection_state_remove	protocols/conn/main.zeeb:334	event	3793	0.054818	0.043149	81920	81920		
Files::analyzer_name	frameworks/files/main.zeeb:448	function	14476	0.053029	0.017845	16384	16384		
connection_state_remove	protocols/conn/removal-hooks.zeeb:74	event	3793	0.049718	0.030939	0	0		

sort -nr -k5
most CPU first

The most expensive
file_sniff handler

#fields	function	location	type	ncall	tot_CPU	child_CPU	tot_Mem	child_Mem
#types	string	string	string	count	interval	interval	count	count
all-BiFs	65-locations	BiF	130510	0.348761	0.000000	6307840	0	
Log::__write	init-frameworks-and-bifs.zeek:32	BiF	11277	0.088300	0.00915			
Files::__add_analyzer	init-frameworks-and-bifs.zeek:32	BiF	14476	0.083714				
to_lower	init-bare.zeek:6850	BiF	22057	0.026889	0.000000	0		
Files::__analyzer_enabled	init-frameworks-and-bifs.zeek:32	BiF	14476	0.021002	0.000000	0	0	
Files::__analyzer_name	init-frameworks-and-bifs.zeek:32	BiF	14476	0.017845	0.000000	16384	0	
join_string_set	init-bare.zeek:6850	BiF	3792	0.016530	0.000000	0	0	
to_upper	init-bare.zeek:6850	BiF	11026	0.013930	0.000000	0	0	
x509_get_certificate_string	<no-location>	BiF	3619	0.013036	0.000000	245760	0	
disable_analyzer	init-bare.zeek:6850	BiF	3788	0.012249	0.000000	0	0	
get_port_transport_proto	init-bare.zeek:6850	BiF	7586	0.011069	0.000000	16384	0	
sha256_hash	init-bare.zeek:6850	BiF	3619	0.007212	0.000000	16384	0	
is_protocol_analyzer	init-bare.zeek:6850	BiF	3792	0.005886	0.000000	0	0	
Files::__enable_reassembly	init-frameworks-and-bifs.zeek:32	BiF	3619	0.005673	0.000000	0	0	
network_time	init-bare.zeek:6850	BiF	4216	0.005122	0.000000	0	0	
Analyzer::__name	init-frameworks-and-bifs.zeek:32	BiF	3792	0.005022	0.000000	0	0	
Files::__set_reassembly_buffer	init-frameworks-and-bifs.zeek:32	BiF	3619	0.004997	0.000000	16384	0	
PacketAnalyzer::register_packet_analyzer	init-bare.zeek:6850	BiF	114	0.002361	0.000000	4866048	0	
fmt	init-bare.zeek:6850	BiF	423	0.001677	0.000000	16384	0	
Broker::__flush_logs	init-frameworks-and-bifs.zeek:32	BiF	336	0.001615	0.000000	16384	0	
Log::__add_filter	init-frameworks-and-bifs.zeek:32	BiF	59	0.000888	0.000000	65536	0	
__init_primary_bifs	<no-location>	BiF	1	0.000704	0.000000	98304	0	
global_options	init-bare.zeek:6850	BiF	1	0.000570	0.000000	16384	0	
__init_secondary_bifs	init-bare.zeek:6850	BiF	1	0.000402	0.000000	81920	0	
Log::__create_stream	init-frameworks-and-bifs.zeek:32	BiF	58	0.000369	0.000000	32768	0	
Option::set_change_handler	init-bare.zeek:6850	BiF	136	0.000338	0.000000	0	0	
Cluster::publish	init-frameworks-and-bifs.zeek:32	BiF	1	0.000279	0.000000	131072	0	
Analyzer::__register_for_port	init-frameworks-and-bifs.zeek:32	BiF	71	0.000202	0.000000	0	0	
Option::set	init-bare.zeek:6850	BiF	1	0.000136	0.000085	0	0	
Cluster::__subscribe	init-frameworks-and-bifs.zeek:32	BiF	2	0.000125	0.000096	0	0	
string_to_pattern	init-bare.zeek:6850	BiF	4	0.000105	0.000000	0	0	
split_string1	init-bare.zeek:6850	BiF	3	0.000076	0.000000	49152	0	
gsub	init-bare.zeek:6850	BiF	2	0.000057	0.000000	16384	0	
Telemetry::__counter_family	init-bare.zeek:6850	BiF	1	0.000054	0.000000	16384	0	
cat	init-bare.zeek:6850	BiF	6	0.000046	0.000000	16384	0	

sort -nr -k5 | grep BiF
BiFs with most CPU

#fields	function		location		type	ncall	tot_CPU	child_CPU	tot_Mem	child_Mem
#types	string	string	string	count	interval	interval	count	count		
all-BiFs	65-locations	BiF	130510	0.348761		0.000000	6307840	0		
Log::__write	init-frameworks-and-bifs.zeek:32	BiF	11277	0.088300		0.00915				
Files::__add_analyzer	init-frameworks-and-bifs.zeek:32	BiF	14476	0.083714						
to_lower	init-bare.zeek:6850	BiF	22057	0.026889		0.000000	0			
Files::__analyzer_enabled	init-frameworks-and-bifs.zeek:32	BiF	14476	0.021002		0.000000	0	0		
Files::__analyzer_name	init-frameworks-and-bifs.zeek:32	BiF	14476	0.017845		0.000000	16384	0		
join_string_set	init-bare.zeek:6850	BiF	3792	0.016530		0.000000	0	0		
to_upper	init-bare.zeek:6850	BiF	11026	0.013930		0.000000	0	0		
x509_get_certificate_string	<no-location>	BiF	3619	0.013036		0.000000	245760	0		
disable_analyzer	init-bare.zeek:6850	BiF	3788	0.012249		0.000000	0	0		
get_port_transport_proto	init-bare.zeek:6850	BiF	7586	0.011069		0.000000	16384	0		
sha256_hash	init-bare.zeek:6850	BiF	3619	0.007212		0.000000	16384	0		
is_protocol_analyzer	init-bare.zeek:6850	BiF	3792	0.005886		0.000000	0	0		
Files::__enable_reassembly	init-frameworks-and-bifs.zeek:32	BiF	3619	0.005673		0.000000	0	0		
network_time	init-bare.zeek:6850	BiF	4216	0.005122		0.000000	0	0		
Analyzer::__name	init-frameworks-and-bifs.zeek:32	BiF	3792	0.005022		0.000000	0	0		
Files::__set_reassembly_buffer	init-frameworks-and-bifs.zeek:32	BiF	3619	0.004997		0.000000	16384	0		
PacketAnalyzer::register_packet_analyzer	init-bare.zeek:6850	BiF	114	0.002361		0.000000	4866048	0		
fmt	init-bare.zeek:6850	BiF	423	0.001677		0.000000	16384	0		
Broker::__flush_logs	init-frameworks-and-bifs.zeek:32	BiF	336	0.001615		0.000000	16384	0		
Log::__add_filter	init-frameworks-and-bifs.zeek:32	BiF	59	0.000888		0.000000	65536	0		
__init_primary_bifs	<no-location>	BiF	1	0.000704		0.000000	98304	0		
global_options	init-bare.zeek:6850	BiF	1	0.000570		0.000000	16384	0		
__init_secondary_bifs	init-bare.zeek:6850	BiF	1	0.000402		0.000000	81920	0		
Log::__create_stream	init-frameworks-and-bifs.zeek:32	BiF	58	0.000369		0.000000	32768	0		
Option::set_change_handler	init-bare.zeek:6850	BiF	136	0.000338		0.000000	0	0		
Cluster::publish	init-frameworks-and-bifs.zeek:32	BiF	1	0.000279		0.000000	131072	0		
Analyzer::__register_for_port	init-frameworks-and-bifs.zeek:32	BiF	71	0.000202		0.000000	0	0		
Option::set	init-bare.zeek:6850	BiF	1	0.000136		0.000085	0	0		
Cluster::__subscribe	init-frameworks-and-bifs.zeek:32	BiF	2	0.000125		0.000096	0	0		
string_to_pattern	init-bare.zeek:6850	BiF	4	0.000105		0.000000	0	0		
split_string1	init-bare.zeek:6850	BiF	3	0.000076		0.000000	49152	0		
gsub	init-bare.zeek:6850	BiF	2	0.000057		0.000000	16384	0		
Telemetry::__counter_family	init-bare.zeek:6850	BiF	1	0.000054		0.000000	16384	0		
cat	init-bare.zeek:6850	BiF	6	0.000046		0.000000	16384	0		

```
sort -nr -k5 | grep BiF
BiFs with most CPU
```

#fields	function	location	type	ncall	tot_CPU	child_CPU	tot_Mem	child_Mem
#types	string	string	string	count	interval	interval	count	count
ssl_handshake_message	protocols/ssl/main.zee	376	event	22057	0.287833			
ssl_established	5-locations	event	18940	0.196962	0.000000		163840	
SSL::add_to_history	protocols/ssl/main.zee	213	function		29291	0.17591		
ssl_plaintext_data	protocols/ssl/main.zee	538	event	29291	0.128121			
SSL::set_session	protocols/ssl/main.zee	204	function		79429	0.126430		
SSL::finish	protocols/ssl/main.zee	267	function		7580	0.109984	0.074076	163840 163840
ssl_established	protocols/ssl/main.zee	512	event	3788	0.103337	0.092747	163840	163840
ssl_change_cipher_spec	protocols/ssl/main.zee	478	event	7234	0.091254	0.052987	0	0
ssl_established	protocols/ssl/main.zee	506	event	3788	0.063466	0.044178	0	0
SSL::log_record	protocols/ssl/main.zee	240	function		7580	0.061827	0.041825	163840 163840
analyzer_confirmation_info	protocols/ssl/main.zee	529	event	3792	0.056280	0.041062		65536 65536
SSL::ssl_finishing	protocols/ssl/files.zee	161	hook	3792	0.039774	0.000000	0	0
ssl_server_hello	2-locations	event	7584	0.039292	0.000000	0	0	
file_sniff	protocols/ssl/files.zee	126	event	3619	0.035886	0.000000	311296	0
ssl_server_hello	protocols/ssl/main.zee	294	event	3792	0.034685	0.004553	0	0
SSL::finalize_ssl	protocols/ssl/main.zee	517	hook	3792	0.030939	0.017275	0	0
ssl_client_hello	2-locations	event	7584	0.024450	0.000000	0	0	
ssl_ecdh_server_params	protocols/ssl/main.zee	330	event	3619	0.023113	0.004409	0	0
ssl_established	protocols/ftp/gridftp.zee	115	event	3788	0.020225	0.009557	0	0
ssl_client_hello	protocols/ssl/main.zee	275	event	3792	0.020015	0.004002	0	0
ssl_extension	protocols/ssl/main.zee	462	event	1548	0.008999	0.003606	0	0
ssl_established	protocols/ssl/main.zee	501	event	3788	0.005085	0.000000	0	0
ssl_established	protocols/ftp/gridftp.zee	99	event	3788	0.004849	0.000000	0	0
ssl_server_hello	protocols/quic/main.zee	268	event	3792	0.004607	0.000000	0	0
ssl_client_hello	protocols/quic/main.zee	260	event	3792	0.004435	0.000000	0	0
ssl_extension_supported_versions	protocols/ssl/main.zee	319	event	344	0.001810	0.000213		0
ssl_extension_server_name	2-locations	event	344	0.001344	0.000000	0	0	
ssl_extension_key_share	protocols/ssl/main.zee	337	event	344	0.001306	0.000189	0	0
ssl_extension_server_name	protocols/ssl/main.zee	346	event	172	0.001045	0.000189	0	0
ssl_extension_server_name	protocols/quic/main.zee	241	event	172	0.000299	0.000000	0	0
zeek_init	protocols/ssl/main.zee	197	event	1	0.000250	0.000234	0	0
zeek_init	protocols/ssl/files.zee	97	event	1	0.000078	0.000053	0	0

```
sort -nr -k5 | grep -i ssl
functions etc. relating to
TLS analysis with most CPU
```

Speeding Up Zeek Scripts

1. Using traditional Zeek scripting
 - a) Avoiding repeated expressions
 - b) Comparing perf. with *micro-benchmarks* / hyperfine
 - c) Locating perf. issues w/ `zeek --profile-scripts`

2. Using Zeek script optimization
 - a) ZAM
 - b) Using ZAM to produce *flamegraphs*
 - c) Compiling to C++

ZAM: compile scripts to optimized IL

- *ZAM (Zeek Abstract Machine)*

ZAM: compile scripts to optimized IL

- ZAM (*Zeek Abstract Machine*)
- Applies a number of optimizations
(Inlining, constant propagation, folding, CSE)

ZAM: compile scripts to optimized IL

- ZAM (*Zeek Abstract Machine*)
- Applies a number of optimizations
(Inlining, constant propagation, folding, CSE)
- Workflow: use `zeek -O ZAM`

```
perf-examples % cat record-ref1.zee
```

```
type r1: record { a: count; };
```

```
type r2: record { b: r1; };
```

```
event zee_init()
```

```
{
```

```
  local x = r2($b = r1($a = 0));
```

```
  local sum = +0; local i = 0;
```

```
  while ( ++i <= 1000000 )
```

```
  {
```

```
    if ( x$b$a > 10 ) sum += 10;
```

```
    else if ( x$b$a > 5 ) sum += 5;
```

```
    else if ( x$b$a > 3 ) sum += 3;
```

```
    else if ( x$b$a > 1 ) ++sum;
```

```
    else --sum;
```

```
  }
```

```
  print sum;
```

```
}
```

Small script that does
8 record references
(along with other stuff)
a million times

Frame for zeek_init:

frame[0] = <internal>::#0 x

frame[1] = sum

frame[2] = i

frame[3] = <internal>::#7

frame[4] = <internal>::#8

Final code for zeek_init:

0: construct-known-record-V 0 (<internal>::#0) aux: map=1 elem-0:<constant> // zeek_init;zeek_init:6

1: construct-known-record-V 0 (x) aux: map=1 elem-0:<slot:0> // zeek_init;zeek_init:6

2: assign-VC-I 1 (sum), 0 // zeek_init;zeek_init:7

3: assign-VC-U 2 (i), 0 // zeek_init;zeek_init:8

4: incru-V 2 (i) aux: CFT: loop // zeek_init;zeek_init:10

5: le-VCb-cond-U 2 (i), 22, 1000000 aux: CFT: loop-cond // zeek_init;zeek_init:10

6: field-RVi-R 3 (<internal>::#7), 0 (x), 0 aux: types=1 // zeek_init;zeek_init:12

7: field-RVi-U 4 (<internal>::#8), 3 (<internal>::#7), 0 aux: types=1 // zeek_init;zeek_init:12

8: lt-CVb-cond-U 4 (<internal>::#8), 11, 10 aux: CFT: if // zeek_init;zeek_init:12

9: add-VVC-I 1 (sum), 1 (sum), 10 aux: CFT: block-end // zeek_init;zeek_init:12

10: goto-b 4 // zeek_init;zeek_init:12

11: lt-CVb-cond-U 4 (<internal>::#8), 14, 5 aux: CFT: if/else // zeek_init;zeek_init:13

12: add-VVC-I 1 (sum), 1 (sum), 5 aux: CFT: block-end // zeek_init;zeek_init:13

13: goto-b 4 // zeek_init;zeek_init:13

14: lt-CVb-cond-U 4 (<internal>::#8), 17, 3 aux: CFT: if/else // zeek_init;zeek_init:14

15: add-VVC-I 1 (sum), 1 (sum), 3 aux: CFT: block-end // zeek_init;zeek_init:14

16: goto-b 4 // zeek_init;zeek_init:14

17: lt-CVb-cond-U 4 (<internal>::#8), 20, 1 aux: CFT: if/else // zeek_init;zeek_init:15

18: incru-V 1 (sum) aux: CFT: block-end // zeek_init;zeek_init:15

19: goto-b 4 // zeek_init;zeek_init:15

20: decru-V 1 (sum) aux: CFT: block-end/else/loop-end // zeek_init;zeek_init:16

21: goto-b 4 // zeek_init;zeek_init:16

22: print1-V 1 (sum) // zeek_init;zeek_init:19

Frame for zeek_init:

```
frame[0] = <internal>::#0 x
frame[1] = sum
frame[2] = i
frame[3] = <internal>::#7
frame[4] = <internal>::#8
```

Frame holding
variables/temporaries

Final code for zeek_init:

```
0: construct-known-record-V 0 (<internal>::#0) aux: map=1 elem-0:<constant> // zeek_init;zeek_init:6
1: construct-known-record-V 0 (x) aux: map=1 elem-0:<slot:0> // zeek_init;zeek_init:6
2: assign-VC-I 1 (sum), 0 // zeek_init;zeek_init:7
3: assign-VC-U 2 (i), 0 // zeek_init;zeek_init:8
4: incru-V 2 (i) aux: CFT: loop // zeek_init;zeek_init:10
5: le-VCb-cond-U 2 (i), 22, 1000000 aux: CFT: loop-cond // zeek_init;zeek_init:10
6: field-RVi-R 3 (<internal>::#7), 0 (x), 0 aux: types=1 // zeek_init;zeek_init:12
7: field-RVi-U 4 (<internal>::#8), 3 (<internal>::#7), 0 aux: types=1 // zeek_init;zeek_init:12
8: lt-CVb-cond-U 4 (<internal>::#8), 11, 10 aux: CFT: if // zeek_init;zeek_init:12
9: add-VVC-I 1 (sum), 1 (sum), 10 aux: CFT: block-end // zeek_init;zeek_init:12
10: goto-b 4 // zeek_init;zeek_init:12
11: lt-CVb-cond-U 4 (<internal>::#8), 14, 5 aux: CFT: if/else // zeek_init;zeek_init:13
12: add-VVC-I 1 (sum), 1 (sum), 5 aux: CFT: block-end // zeek_init;zeek_init:13
13: goto-b 4 // zeek_init;zeek_init:13
14: lt-CVb-cond-U 4 (<internal>::#8), 17, 3 aux: CFT: if/else // zeek_init;zeek_init:14
15: add-VVC-I 1 (sum), 1 (sum), 3 aux: CFT: block-end // zeek_init;zeek_init:14
16: goto-b 4 // zeek_init;zeek_init:14
17: lt-CVb-cond-U 4 (<internal>::#8), 20, 1 aux: CFT: if/else // zeek_init;zeek_init:15
18: incu-V 1 (sum) aux: CFT: block-end // zeek_init;zeek_init:15
19: goto-b 4 // zeek_init;zeek_init:15
20: decu-V 1 (sum) aux: CFT: block-end/else/loop-end // zeek_init;zeek_init:16
21: goto-b 4 // zeek_init;zeek_init:16
22: print1-V 1 (sum) // zeek_init;zeek_init:19
```

Frame for zeek_init:

frame[0] = <internal>::#0 x

frame[1] = sum

frame[2] = i

frame[3] = <internal>::#7

frame[4] = <internal>::#8

Final code for zeek_init:

```
0: construct-known-record-V 0 (<internal>::#0) aux: map=1 elem-0:<constant> // zeek_init;zeek_init:6
1: construct-known-record-V 0 (x) aux: map=1 elem-0:<slot:0> // zeek_init;zeek_init:6
2: assign-VC-I 1 (sum), 0 // zeek_init;zeek_init:7
3: assign-VC-U 2 (i), 0 // zeek_init;zeek_init:8
4: incru-V 2 (i) aux: CFT: loop // zeek_init;zeek_init:10
5: le-VCb-cond-U 2 (i), 22, 1000000 aux: CFT: loop-cond // zeek_init;zeek_init:10
6: field-RVi-R 3 (<internal>::#7), 0 (x), 0 aux: types=1 // zeek_init;zeek_init:12
7: field-RVi-U 4 (<internal>::#8), 3 (<internal>::#7), 0 aux: types=1 // zeek_init;zeek_init:12
8: lt-CVb-cond-U 4 (<internal>::#8), 11, 10 aux: CFT: if // zeek_init;zeek_init:12
9: add-VVC-I 1 (sum), 1 (sum), 5 aux: CFT: block-end // zeek_init;zeek_init:12
10: goto-b 4 // zeek_init;zeek_init:12
11: lt-CVb-cond-U 4 (<internal>::#8), 17, 3 aux: CFT: if/else // zeek_init;zeek_init:13
12: add-VVC-I 1 (sum), 1 (sum), 5 aux: CFT: block-end // zeek_init;zeek_init:13
13: goto-b 4 // zeek_init;zeek_init:13
14: lt-CVb-cond-U 4 (<internal>::#8), 17, 3 aux: CFT: if/else // zeek_init;zeek_init:14
15: add-VVC-I 1 (sum), 1 (sum), 3 aux: CFT: block-end // zeek_init;zeek_init:14
16: goto-b 4 // zeek_init;zeek_init:14
17: lt-CVb-cond-U 4 (<internal>::#8), 20, 1 aux: CFT: if/else // zeek_init;zeek_init:15
18: incri-V 1 (sum) aux: CFT: block-end // zeek_init;zeek_init:15
19: goto-b 4 // zeek_init;zeek_init:15
20: decr-V 1 (sum) aux: CFT: block-end/else/loop-end // zeek_init;zeek_init:16
21: goto-b 4 // zeek_init;zeek_init:16
22: print1-V 1 (sum) // zeek_init;zeek_init:19
```

Low-level
execution IL

```
perf-examples % hyperfine 'zeek record-ref1.zeek' 'zeek -0 ZAM record-ref1.zeek'
```

How do we fare using
ZAM instead of regular
script execution?

```
perf-examples % hyperfine 'zeek record-ref1.zeek' 'zeek -0 ZAM record-ref1.zeek'
```

Benchmark 1: zeek record-ref1.zeek

Time (mean $\pm$ σ): 2.923 s $\pm$ 0.010 s [User: 2.776 s, System: 0.143 s]
Range (min ... max): 2.906 s ... 2.939 s 10 runs

Benchmark 2: zeek -0 ZAM record-ref1.zeek

Time (mean $\pm$ σ): 5.959 s $\pm$ 0.051 s [User: 5.782 s, System: 0.172 s]
Range (min ... max): 5.900 s ... 6.045 s 10 runs

Summary

zeek record-ref1.zeek ran
2.04 $\pm$ 0.02 times faster than zeek -0 ZAM record-ref1.zeek

Uh-oh!
It takes twice as long!

```
[perf-examples % hyperfine 'zeek -b record-ref1.zeek' 'zeek -b -0 ZAM record-ref1.zeek'
```

Hmmm maybe it's compilation overhead. Let's use **-b** so there's much less for ZAM to compile ...

```
[perf-examples % hyperfine 'zeek -b record-ref1.zeek' 'zeek -b -0 ZAM record-ref1.zeek']
```

Benchmark 1: zeek -b record-ref1.zeek

Time (mean $\pm$ σ): 2.505 s $\pm$ 0.013 s [User: 2.410 s, System: 0.088 s]
Range (min ... max): 2.486 s ... 2.537 s 10 runs

Benchmark 2: zeek -b -0 ZAM record-ref1.zeek

Time (mean $\pm$ σ): 890.1 ms $\pm$ 12.6 ms [User: 799.3 ms, System: 84.4 ms]
Range (min ... max): 877.4 ms ... 915.9 ms 10 runs

Summary

zeek -b -0 ZAM record-ref1.zeek ran

2.81 $\pm$ 0.04 times faster than zeek -b record-ref1.zeek

ZAM is 2.8x faster!
Are we done?

```
[perf-examples % hyperfine 'zeek -b record-ref1.zeek' 'zeek -b -0 ZAM record-ref1.zeek']
```

Benchmark 1: zeek -b record-ref1.zeek

Time (**mean** $\pm$ **σ**): **2.505 s** $\pm$ **0.013 s** [User: **2.410 s**, System: **0.088 s**]
Range (**min** ... **max**): **2.486 s** ... **2.537 s** 10 runs

Benchmark 2: zeek -b -0 ZAM record-ref1.zeek

Time (**mean** $\pm$ **σ**): **890.1 ms** $\pm$ **12.6 ms** [User: **799.3 ms**, System: **84.4 ms**]
Range (**min** ... **max**): **877.4 ms** ... **915.9 ms** 10 runs

Summary

zeek -b -0 ZAM record-ref1.zeek ran
2.81 $\pm$ 0.04 times faster than **zeek -b record-ref1.zeek**

Hmmm how much of
ZAM's 890 ms is still
compilation overhead?

```
perf-examples % hyperfine 'zeek -b /dev/null' 'zeek -b -0 ZAM /dev/null'
```

Let's run without any scripts

```
perf-examples % hyperfine 'zeek -b /dev/null' 'zeek -b -0 ZAM /dev/null'
```

Benchmark 1: zeek -b /dev/null

Time (mean $\pm$ σ): **574.9 ms** $\pm$ 9.6 ms [User: 489.6 ms, System: 80.0 ms]
Range (min ... max): 566.8 ms ... 597.9 ms 10 runs

Benchmark 2: zeek -b -0 ZAM /dev/null

Time (mean $\pm$ σ): **797.5 ms** $\pm$ 12.6 ms [User: 708.8 ms, System: 82.4 ms]
Range (min ... max): 786.2 ms ... 830.1 ms 10 runs

Summary

zeek -b /dev/null ran

1.39 $\pm$ **0.03** times faster than zeek -b -0 ZAM /dev/null

ZAM is slower, but we see about 220 ms
of *additional* compilation overhead ...

```
[perf-examples % hyperfine 'zeek -b record-ref1.zeek' 'zeek -b -0 ZAM record-ref1.zeek']
```

Benchmark 1: zeek -b record-ref1.zeek

Time (mean $\pm$ σ): 2.505 s $\pm$ 0.013 s [User: 2.410 s, System: 0.088 s]
Range (min ... max): 2.486 s ... 2.537 s 10 runs

Benchmark 2: zeek -b -0 ZAM record-ref1.zeek

Time (mean $\pm$ σ): 890.1 ms $\pm$ 12.6 ms [User: 799.3 ms, System: 84.4 ms]
Range (min ... max): 877.4 ms ... 915.9 ms 10 runs

Summary

zeek -b -0 ZAM record-ref1.zeek ran

2.81 $\pm$ 0.04 times faster than zeek -b record-ref1.zeek

So adjusted for startup overhead,
non-ZAM is 2505 ms – 575 ms = 1930 ms
ZAM is 890 ms – 797 = 93 ms (20:1!)

```
perf-examples % hyperfine 'zeek -b record-ref10.zeek' 'zeek -b -0 ZAM record-ref  
10.zeek'
```

Scripts do 10M loops rather than 1M loops
Larger inputs lessen startup/exit effects

```
perf-examples % hyperfine 'zeek -b record-ref10.zeek' 'zeek -b -0 ZAM record-ref10.zeek'
```

Benchmark 1: zeek -b record-ref10.zeek

Time (mean $\pm$ σ): 19.723 s $\pm$ 0.053 s [User: 19.580 s, System: 0.118 s]

Range (min ... max): 19.660 s ... 19.811 s 10 runs

Benchmark 2: zeek -b -0 ZAM record-ref10.zeek

Time (mean $\pm$ σ): 1.640 s $\pm$ 0.017 s [User: 1.546 s, System: 0.085 s]

Range (min ... max): 1.625 s ... 1.677 s 10 runs

Summary

zeek -b -0 ZAM record-ref10.zeek ran

12.03 $\pm$ 0.13 times faster than zeek -b record-ref10.zeek

That's more like it!
(though still not the asymptotic 20:1)

```
perf-examples % hyperfine 'zeek -b -0 ZAM record-ref10.zeek' 'zeek -b -0 ZAM record-ref20.zeek'
```

How does the version that uses a local to reduce record references fare w/ ZAM?

```
perf-examples % hyperfine 'zeek -b -0 ZAM record-ref10.zeek' 'zeek -b -0 ZAM record-ref20.zeek'
```

Benchmark 1: zeek -b -0 ZAM record-ref10.zeek

Time (mean $\pm$ σ): 1.639 s $\pm$ 0.013 s [User: 1.547 s, System: 0.085 s]

Range (min ... max): 1.628 s ... 1.670 s 10 runs

Benchmark 2: zeek -b -0 ZAM record-ref20.zeek

Time (mean $\pm$ σ): 1.637 s $\pm$ 0.009 s [User: 1.546 s, System: 0.084 s]

Range (min ... max): 1.627 s ... 1.656 s 10 runs

Summary

zeek -b -0 ZAM record-ref20.zeek ran

1.00 $\pm$ 0.01 times faster than zeek -b -0 ZAM record-ref10.zeek

No difference!

Because ZAM avoids the extra record references in the original, too

ZAM: compile scripts to optimized IL

- ZAM (*Zeek Abstract Machine*)
- Applies a number of optimizations
(Inlining, constant propagation, folding, CSE)
- Workflow: add `-O` ZAM command-line arg.
- Compatibility: full (other than some error message formats)

ZAM: compile scripts to optimized IL

- ZAM (*Zeek Abstract Machine*)
- Applies a number of optimizations
(Inlining, constant propagation, folding, CSE)
- Workflow: add `-O` ZAM command-line arg.
- Compatibility: full (other than some error message formats)
- Notes:
 - Increases startup time by a few seconds
 - Supports per-script line profiling via sampling

Speeding Up Zeek Scripts

1. Using traditional Zeek scripting
 - a) Avoiding repeated expressions
 - b) Comparing perf. with *micro-benchmarks* / hyperfine
 - c) Locating perf. issues w/ `zeek --profile-scripts`
2. Using Zeek script optimization
 - a) ZAM
 - b) Using ZAM to produce *flamegraphs*
 - c) Compiling to C++

```
ZAM-prof % wc prof.out  
89457 960663 44164300 prof.out
```

```
zeek ... -profile-ZAM >prof.out
```

```
ZAM-prof % wc prof.out
89457 960663 44164300 prof.out
```

```
ZAM-prof % head -25 prof.out
```

```
Profiling overhead = 1148 nsec/instruction
```

assign-VC-U	365072	0.113231
assign-VC-D	239634	0.072290
assign-VC-S	205837	0.083474
assign-VC-I	370940	0.125720
assign-VV-R	849907	0.450144
assign-VV-V	157886	0.058435
assign-VV-U	419066	0.123931
assign-VV-D	37663	0.014494
assign-VV-T	140933	0.066382
assign-VV-N	411909	0.154010
assign-VV-A	193404	0.075263
assign-VV-F	4193	0.001717
assign-VV-S	32629	0.012210
assign-VV-I	636106	0.253592
field-lhs-assign-FC-U	49353	0.038644
field-lhs-assign-FC-D	7	0.000002
field-lhs-assign-FC-S	2781	0.002084
field-lhs-assign-FC-I	94499	0.051255
field-lhs-assign-FV-R	755771	0.558813
field-lhs-assign-FV-O	177	0.000091
field-lhs-assign-FV-V	41250	0.031484
field-lhs-assign-FV-U	172051	0.076822
field-lhs-assign-FV-D	279369	0.118326
field-lhs-assign-FV-T	29813	0.029092

```
zeek ... --profile-ZAM >prof.out
```

Conn:: CPU time 16.447924, 15379377 instructions
Known:: CPU time 13.737616, 12082246 instructions
SMB2:: CPU time 10.435512, 7870606 instructions
Log:: CPU time 8.135690, 252345 instructions
CorelightBurst:: CPU time 5.722345, 5313349 instructions
ConnThreshold:: CPU time 5.111487, 4656336 instructions
Files:: CPU time 4.548738, 3017925 instructions
LocalSubnets:: CPU time 4.405429, 3648133 instructions
AppID:: CPU time 3.368948, 4244155 instructions
SSL:: CPU time 3.183886, 3218384 instructions
HTTP_C2:: CPU time 3.144021, 4724353 instructions
ConnPolling:: CPU time 2.734802, 2601174 instructions
genisys:: CPU time 2.323668, 2687197 instructions
HTTP:: CPU time 2.226542, 2406172 instructions
SMB:: CPU time 1.658496, 1318371 instructions
Weird:: CPU time 1.626456, 865347 instructions
Common:: CPU time 1.464670, 974033 instructions
DGA:: CPU time 1.335080, 1131320 instructions
Site:: CPU time 1.276879, 1410304 instructions
Namecache:: CPU time 1.237501, 1341690 instructions
DCE_RPC:: CPU time 1.191557, 1078247 instructions
RDP:: CPU time 1.189367, 530751 instructions
Viz:: CPU time 0.925224, 897615 instructions
JA3_RES_402:: CPU time 0.924991, 1094714 instructions
PacketAnalyzer:: CPU time 0.910821, 507858 instructions
MeterpreterDetection_HTTPHeader:: CPU time 0.887101, 719295 instructions
DOH:: CPU time 0.861172, 837047 instructions
X509:: CPU time 0.685712, 765979 instructions
CVE_2021_44228:: CPU time 0.665066, 1612880 instructions

Per-module counts of
instructions, CPU time

129 Modules

connection_state_remove CPU time 21.251043, 84563 calls, 2769169 instructions
smb2_message CPU time 18.882096, 140948 calls, 6515194 instructions
partial_connection CPU time 16.155029, 45193 calls, 1522981 instructions
connection_state_remove CPU time 13.890685, 84563 calls, 5718901 instructions
LocalSubnets::infer_current_subnets_evt CPU time 12.462723, 1 calls, 19 instructions
LocalSubnets::infer_current_subnets CPU time 12.282389, 1 calls, 156915 instructions
content_gap CPU time 10.175484, 268977 calls, 4887437 instructions
ConnPolling::check CPU time 8.423532, 84566 calls, 848140 instructions
ConnThreshold::set_current_threshold CPU time 7.778922, 89380 calls, 3217680 instructions
connection_state_remove CPU time 7.035430, 84563 calls, 422815 instructions
ssl_established CPU time 6.502396, 1875 calls, 5625 instructions
finalize_logging CPU time 6.263308, 1 calls, 108099 instructions
genisys::msg CPU time 6.205011, 247 calls, 2650817 instructions
get_file_handle CPU time 5.022808, 49513 calls, 495130 instructions
connection_established CPU time 4.216216, 13752 calls, 1655279 instructions
LocalSubnets::update_local_nets CPU time 4.066560, 1 calls, 1877008 instructions
Known::track_services CPU time 3.918525, 14902 calls, 1200384 instructions
new_connection CPU time 3.898936, 84563 calls, 1527841 instructions
connection_established CPU time 3.738258, 13752 calls, 588096 instructions
http_message_done CPU time 3.430272, 8870 calls, 1827016 instructions
Conn::LongConnection::long_callback CPU time 3.288609, 68147 calls, 1635528 instructions
conn_weird CPU time 3.200444, 25542 calls, 953296 instructions
SSL::ssl_finishing CPU time 3.186272, 2592 calls, 1271200 instructions
file_sniff CPU time 2.975979, 9951 calls, 1024274 instructions
connection_state_remove CPU time 2.861411, 84563 calls, 338252 instructions
connection_state_remove CPU time 2.719214, 84563 calls, 262758 instructions
connection_viz_summary CPU time 2.705907, 77988 calls, 909617 instructions
LocalSubnets::simplify_subnets CPU time 2.552753, 1 calls, 1135653 instructions
http_header CPU time 2.494208, 42966 calls, 1279912 instructions

Per-function counts of calls,
instructions, CPU time

660 functions ran
1,007 “did not execute”

```
content_gap CPU time 10.175484, 268977 calls, 4887437 instructions
content_gap 0 268977 0.119125 load-val-VV-R 0 (Conn::c), interpreter frame[0]
content_gap 1 268977 0.103368 load-val-VV-U 1 (Conn::length), interpreter frame[3]
content_gap 2 268977 0.103948 not-has-field-cond-VV 0 (Conn::c), 15, 10
content_gap 3 6488 0.003557 field-RVi-R 2 (<internal>::#2), 0 (Conn::c), 0
content_gap 4 6488 0.003948 field-RVi-U 3 (<internal>::#3), 2 (<internal>::#2), 3
content_gap 5 6488 0.002480 get-port-transport-proto-VV 3 (Conn::p.2), 3 (<internal>::#3)
content_gap 6 6488 0.002433 field-RVi-D 4 (<internal>::#4), 0 (Conn::c), 3
content_gap 7 6488 0.003177 field-RVi-S 2 (<internal>::#5), 0 (Conn::c), 7
content_gap 8 6488 0.044150 construct-known-record-V 2 (<internal>::#6)
content_gap 9 6488 0.003821 field-lhs-assign-FV-R 0 (Conn::c), 2 (<internal>::#7)
content_gap 10 268977 0.116475 field-RVi-R 2 (<internal>::#7), 0 (Conn::c), 15
content_gap 11 268977 0.138658 field-lhs-assign-FVi-R 2 (<internal>::#7), 0 (Conn::c), 15
content_gap 12 268977 0.103678 has-field-cond-VV 0 (Conn::c), 8, 25
content_gap 13 41 0.000016 field-RVi-V 5 (<internal>::#20), 0 (Conn::c), 8
content_gap 14 41 0.000014 size-VV-V 3 (<internal>::#21), 5 (<internal>::#20)
content_gap 15 41 0.000013 lt-VCV-cond-U 3 (<internal>::#21), 25, 0
content_gap 16 41 0.000014 not-has-field-cond-VV 2 (<internal>::#7), 17, 20
content_gap 17 22 0.000066 construct-set-VV 6 (<internal>::#9), 0
content_gap 18 22 0.000054 table-coerce-VV 6 (<internal>::#10), 6 (<internal>::#9)
content_gap 19 22 0.000011 field-lhs-assign-FV-T 2 (<internal>::#7), 6 (<internal>::#10), 17
content_gap 20 41 0.000014 field-RVi-T 6 (<internal>::#13), 2 (<internal>::#7), 17
content_gap 21 41 0.000014 sub-VVC-U 3 (<internal>::#17), 3 (<internal>::#21), 1
content_gap 22 41 0.000034 index-vec-VV 5 (<internal>::#18), 5 (<internal>::#20), 3 (<internal>::#17)
content_gap 23 41 0.000015 field-RVi-S 5 (<internal>::#19), 5 (<internal>::#18), 2
content_gap 24 41 0.000176 addstmt1-VV 6 (<internal>::#13), 5 (<internal>::#19)
content_gap 25 268977 0.125867 load-global-VV-T 5 (Site::local_nets), 0
content_gap 26 268977 0.099600 size-VV-T 3 (<internal>::#30), 5 (Site::local_nets)
content_gap 27 268977 0.090798 lt-VCV-cond-U 3 (<internal>::#30), 35, 0
```

Per-ZAM instruction
counts and CPU time

content_gap CPU time 10.175484, 268977 calls, 4887437 instructions

content_gap 0 268977 0.119125 load-val-VV-R 0 (Conn::c), interpreter frame[0]

content_gap 1 268977 0.103368 load-val-VV-U 1 (Conn::length), interpreter frame[3]

content_gap 2 268977 0.103948 not-has-field-cond-VV 0 (Conn::c), 15, 10

content_gap 3 6488 0.003557 field-RVi-R 2 (<internal>::#2), 0 (Conn::c), 0

content_gap 4 6488 0.003948 field-RVi-U 3 (<internal>::#2), 2 (<internal>::#2), 2

content_gap 5 6488 0.002480 get-port-transp 3 (<internal>::#2), 2 (<internal>::#2), 3

content_gap 6 6488 0.002433 field-RVi-D 4 (<internal>::#2), 2 (<internal>::#2), 2

content_gap 7 6488 0.003177 field-RVi-S 2 (<internal>::#5), 0 (Conn::c), 7

content_gap 8 6488 0.044150 construct-known-record-V 2 (<internal>::#6)

content_gap 9 6488 0.003821 field-lhs-assign-FV-R 0 (Conn::c), 2 (<internal>::#6), 15

content_gap 10 268977 0.116475 field-RVi-R 2 (<internal>::#7), 0 (Conn::c), 15

content_gap 11 268977 0.138658 field-lhs-assign-FVi-R 2 (<internal>::#7), 0 (Conn::c), 0, 2

content_gap 12 268977 0.103678 has-field-cond-VV 0 (Conn::c), 8, 25

content_gap 13 41 0.000016 field-RVi-V 5 (<internal>::#20), 0 (Conn::c), 8

content_gap 14 41 0.000014 size-VV-V 3 (<internal>::#21), 5 (<internal>::#20)

content_gap 15 41 0.000013 lt-VCV-cond-U 3 (<internal>::#21), 25, 0

content_gap 16 41 0.000014 not-has-field-cond-VV 2 (<internal>::#7), 17, 20

content_gap 17 22 0.000066 construct-set-VV 6 (<internal>::#9), 0

content_gap 18 22 0.000054 table-coerce-VV 6 (<internal>::#10), 6 (<internal>::#9)

content_gap 19 22 0.000011 field-lhs-assign-FV-T 2 (<internal>::#7), 6 (<internal>::#10), 17

content_gap 20 41 0.000014 field-RVi-T 6 (<internal>::#13), 2 (<internal>::#7), 17

content_gap 21 41 0.000014 sub-VVC-U 3 (<internal>::#17), 3 (<internal>::#21), 1

content_gap 22 41 0.000034 index-vec-VV 5 (<internal>::#18), 5 (<internal>::#20), 3 (<internal>::#17)

content_gap 23 41 0.000015 field-RVi-S 5 (<internal>::#19), 5 (<internal>::#18), 2

content_gap 24 41 0.000176 addstmt1-VV 6 (<internal>::#13), 5 (<internal>::#19)

content_gap 25 268977 0.125867 load-global-VV-T 5 (Site::local_nets), 0

content_gap 26 268977 0.099600 size-VV-T 3 (<internal>::#30), 5 (Site::local_nets)

content_gap 27 268977 0.090798 lt-VCV-cond-U 3 (<internal>::#30), 35, 0

This code always executes

```
content_gap CPU time 10.175484, 268977 calls, 4887437 instructions
content_gap 0 268977 0.119125 load-val-VV-R 0 (Conn::c), interpreter frame[0]
content_gap 1 268977 0.103368 load-val-VV-U 1 (Conn::length), interpreter frame[3]
content_gap 2 268977 0.103948 not-has-field-cond-VV 0 (Conn::c), 15, 10
content_gap 3 6488 0.003557 field-RVi-R 2 (<internal>::#2), 0 (Conn::c), 0
content_gap 4 6488 0.003948 field-RVi-U 3 (<internal>::#3), 2 (<internal>::#2), 3
content_gap 5 6488 0.002480 get-port-transport-proto-VV 3 (Conn::p.2), 3 (<internal>::#3)
content_gap 6 6488 0.002433 field-RVi-D 4 (<internal>::#4), 0 (Conn::c), 3
content_gap 7 6488 0.003177 field-RVi-S 2 (<internal>::#5), 0 (Conn::c), 7
content_gap 8 6488 0.044150 construct-known-record-V 2 (<internal>::#6)
content_gap 9 6488 0.003821 field-lhs-assign-FV-R 0 (Conn::c), 2 (<internal>::#6), 15
content_gap 10 268977 0.116475 field-RVi-R 2 (<internal>::#7), 0 (Conn::c), 15
content_gap 11 268977 0.138658 field-lhs-assign-FVi-R 2 (<internal>::#7), 0 (Conn::c), 0, 2
content_gap 12 268977 0.103678 field-lhs-assign-FV-U 0 (Conn::c), 8, 25
content_gap 13 41 0.000016 field-lhs-assign-FV-R 2 (<internal>::#7), 0 (Conn::c), 8
content_gap 14 41 0.000014 field-lhs-assign-FV-U 0 (Conn::c), 8, 25
content_gap 15 41 0.000013 lt-VCV-cond-U 3 (<internal>::#21), 25, 0
content_gap 16 41 0.000014 not-has-field-cond-VV 2 (<internal>::#7), 17, 20
content_gap 17 22 0.000066 construct-set-VV 6 (<internal>::#9), 0
content_gap 18 22 0.000054 table-coerce-VV 6 (<internal>::#10), 6 (<internal>::#9)
content_gap 19 22 0.000011 field-lhs-assign-FV-T 2 (<internal>::#7), 6 (<internal>::#10), 17
content_gap 20 41 0.000014 field-RVi-T 6 (<internal>::#13), 2 (<internal>::#7), 17
content_gap 21 41 0.000014 sub-VVC-U 3 (<internal>::#17), 3 (<internal>::#21), 1
content_gap 22 41 0.000034 index-vec-VV 5 (<internal>::#18), 5 (<internal>::#20), 3 (<internal>::#17)
content_gap 23 41 0.000015 field-RVi-S 5 (<internal>::#19), 5 (<internal>::#18), 2
content_gap 24 41 0.000176 addstmt1-VV 6 (<internal>::#13), 5 (<internal>::#19)
content_gap 25 268977 0.125867 load-global-VV-T 5 (Site::local_nets), 0
content_gap 26 268977 0.099600 size-VV-T 3 (<internal>::#30), 5 (Site::local_nets)
content_gap 27 268977 0.090798 lt-VCV-cond-U 3 (<internal>::#30), 35, 0
```

This code rarely does

```

content_gap CPU time 10.175484, 268977 calls, 4887437 instructions
content_gap 0 268977 0.119125 load-val-VV-R 0 (Conn::c), interpreter frame[0]
content_gap 1 268977 0.103368 load-val-VV-U 1 (Conn::length), interpreter frame[3]
content_gap 2 268977 0.103948 not-has-field-cond-VV 0 (Conn::c), 15, 10
content_gap 3 6488 0.003557 field-RVi-R 2 (<internal>::#2), 0 (Conn::c), 0
content_gap 4 6488 0.003948 field-RVi-U 3 (<internal>::#3), 2 (<internal>::#2), 3
content_gap 5 6488 0.002480 get-port-transport-proto-VV 3 (Conn::p.2), 3 (<internal>::#3)
content_gap 6 6488 0.002433 field-RVi-D 4 (<internal>::#4), 0 (Conn::c), 3
content_gap 7 6488 0.003177 field-RVi-S 2 (<internal>::#5), 0 (Conn::c), 7
content_gap 8 6488 0.044150 construct-known-record-V 2 (<internal>::#6)
content_gap 9 6488 0.003821 field-lhs-assign-FV-R 0 (Conn::c), 2 (<internal>::#6), 15
content_gap 10 268977 0.116475 field-RVi-R 2 (<internal>::#7), 0 (Conn::c), 15
content_gap 11 268977 0.138658 field-lhs-assign-FVi-R 2 (<internal>::#7), 0 (Conn::c), 0, 2
content_gap 12 268977 0.103678 has-field-cond-VV 0 (Conn::c), 8, 25
content_gap 13 41 0.000016 field-RVi-V 5 (<internal>::#20), 0 (Conn::c), 8
content_gap 14 41 0.000014 size-VV-V 3 (<internal>::#21), 5 (<internal>::#20)
content_gap 15 41 0.000013 lt-VCV-cond-U 3 (<internal>::#21), 25, 0
content_gap 16 41 0.000014 not-has-field-cond-VV 2 (<internal>::#7), 17, 20
content_gap 17 22 0.000066 construct-set-VV 6 (<internal>::#9), 0
content_gap 18 22 0.000054 table-coerce-VV 6 (<internal>::#10), 6 (<internal>::#9)
content_gap 19 22 0.000011 field-lhs-assign-FV-T 2 (<internal>::#7), 6 (<internal>::#10), 17
content_gap 20 41 0.000014 field-RVi-T 6 (<internal>::#13), 2 (<internal>::#7), 17
content_gap 21 41 0.000014 sub-VVC-U 3 (<internal>::#17), 3 (<internal>::#21), 1
content_gap 22 41 0.000034 index-vec-VV 5 (<internal>::#18), 5 (<internal>::#20), 3 (<internal>::#17)
content_gap 23 41 0.000015 field-RVi-S 5 (<internal>::#19), 5 (<internal>::#18), 2
content_gap 24 41 0.000176 addstmt1-VV 6 (<internal>::#13), 5 (<internal>::#19)
content_gap 25 268977 0.125867 load-global-VV-T 5 (Site::local_nets), 0
content_gap 26 268977 0.099600 s :local_nets)
content_gap 27 268977 0.090798 l ... and this almost 0

```

never

```
content_gap CPU time 10.175484, 268977 calls, 4887437 instructions
content_gap 0 268977 0.119125 load-val-VV-R 0 (Conn::c), interpreter frame[0] // Conn::content_gap;Conn::content_gap:286-290
content_gap 1 268977 0.103368 load-val-VV-U 1 (Conn::length), interpreter frame[3] // Conn::content_gap;Conn::content_gap:286-290
content_gap 2 268977 0.103948 not-has-field-cond-VVV 0 (Conn::c), 15, 10 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242
content_gap 3 6488 0.003557 field-RVi-R 2 (<internal>::#2), 0 (Conn::c), 0 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 4 6488 0.003948 field-RVi-U 3 (<internal>::#3), 2 (<internal>::#2), 3 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 5 6488 0.002480 get-port-transport-proto-VV 3 (Conn::p.2), 3 (<internal>::#3) // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 6 6488 0.002433 field-RVi-D 4 (<internal>::#4), 0 (Conn::c), 3 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 7 6488 0.003177 field-RVi-S 2 (<internal>::#5), 0 (Conn::c), 7 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 8 6488 0.044150 construct-known-record-V 2 (<internal>::#6) // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 9 6488 0.003821 field-lhs-assign-FV-R 0 (Conn::c), 2 (<internal>::#6), 15 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 10 268977 0.116475 field-RVi-R 2 (<internal>::#7), 0 (Conn::c), 15 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:245
content_gap 11 268977 0.138658 field-lhs-assign-FVi-R 2 (<internal>::#7), 0 (Conn::c), 0, 2 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:245
content_gap 12 268977 0.103678 has-field-cond-VVV 0 (Conn::c), 8, 25 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:246-250
```

```
content_gap CPU time 10.175484, 268977 calls, 4887437 instructions
content_gap 0 268977 0.119125 load-val-VV-R 0 (Conn::c), interpreter frame[0] // Conn::content_gap;Conn::content_gap:286-290
content_gap 1 268977 0.103368 load-val-VV-U 1 (Conn::length), interpreter frame[3] // Conn::content_gap;Conn::content_gap:286-290
content_gap 2 268977 0.103948 not-has-field-cond-VVV 0 (Conn::c), 15, 10 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242
content_gap 3 6488 0.003557 field-RVi-R 2 (<internal>::#2), 0 (Conn::c), 0 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 4 6488 0.003948 field-RVi-U 3 (<internal>::#3), 2 (<internal>::#2), 3 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 5 6488 0.002480 get-port-transport-proto-VV 3 (Conn::p.2), 3 (<internal>::#3) // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 6 6488 0.002433 field-RVi-D 4 (<internal>::#4), 0 (Conn::c), 3 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 7 6488 0.003177 field-RVi-S 2 (<internal>::#5), 0 (Conn::c), 7 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 8 6488 0.044150 construct-known-record-V 2 (<internal>::#6) // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 9 6488 0.003821 field-lhs-assign-FV-R 0 (Conn::c), 2 (<internal>::#6), 15 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 10 268977 0.116475 field-RVi-R 2 (<internal>::#7), 0 (Conn::c), 15 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:245
content_gap 11 268977 0.138658 field-lhs-assign-FVi-R 2 (<internal>::#7), 0 (Conn::c), 0, 2 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:245
content_gap 12 268977 0.103678 has-field-cond-VVV 0 (Conn::c), 8, 25 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:246-250
```

```
content_gap CPU time 10.175484, 268977 calls, 4887437 instructions
content_gap 0 268977 0.119125 load-val-VV-R 0 (Conn::c), interpreter frame[0] // Conn::content_gap;Conn::content_gap:286-290
content_gap 1 268977 0.103368 load-val-VV-U 1 (Conn::length), interpreter frame[3] // Conn::content_gap;Conn::content_gap:286-290
content_gap 2 268977 0.103948 not-has-field-cond-VVV 0 (Conn::c), 15, 10 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242
content_gap 3 6488 0.003557 field-RVi-R 2 (<internal>::#2), 0 (Conn::c), 0 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 4 6488 0.003948 field-RVi-U 3 (<internal>::#3), 2 (<internal>::#2), 3 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 5 6488 0.002480 get-port-transport-proto-VV 3 (Conn::p.2), 3 (<internal>::#3) // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 6 6488 0.002433 field-RVi-D 4 (<internal>::#4), 0 (Conn::c), 3 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 7 6488 0.003177 field-RVi-S 2 (<internal>::#5), 0 (Conn::c), 7 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 8 6488 0.044150 construct-known-record-V 2 (<internal>::#6) // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 9 6488 0.003821 field-lhs-assign-FV-R 0 (Conn::c), 2 (<internal>::#6), 15 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 10 268977 0.116475 field-RVi-R 2 (<internal>::#7), 0 (Conn::c), 15 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:245
content_gap 11 268977 0.138658 field-lhs-assign-FVi-R 2 (<internal>::#7), 0 (Conn::c), 0, 2 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:245
content_gap 12 268977 0.103678 has-field-cond-VVV 0 (Conn::c), 8, 25 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:246-250
```

```
content_gap CPU time 10.175484, 268977 calls, 4887437 instructions
content_gap 0 268977 0.119125 load-val-VV-R 0 (Conn::c), interpreter frame[0] // Conn::content_gap;Conn::content_gap:286-290
content_gap 1 268977 0.103368 load-val-VV-U 1 (Conn::length), interpreter frame[3] // Conn::content_gap;Conn::content_gap:286-290
content_gap 2 268977 0.103948 not-has-field-cond-VVV 0 (Conn::c), 15, 10 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242
content_gap 3 6488 0.003557 field-RVi-R 2 (<internal>::#2), 0 (Conn::c), 0 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 4 6488 0.003948 field-RVi-U 3 (<internal>::#3), 2 (<internal>::#2), 3 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 5 6488 0.002480 get-port-transport-proto-VV 3 (Conn::p.2), 3 (<internal>::#3) // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 6 6488 0.002433 field-RVi-D 4 (<internal>::#4), 0 (Conn::c), 3 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 7 6488 0.003177 field-RVi-S 2 (<internal>::#5), 0 (Conn::c), 7 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 8 6488 0.044150 construct-known-record-V 2 (<internal>::#6) // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 9 6488 0.003821 field-lhs-assign-FV-R 0 (Conn::c), 2 (<internal>::#6), 15 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 10 268977 0.116475 field-RVi-R 2 (<internal>::#7), 0 (Conn::c), 15 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:245
content_gap 11 268977 0.138658 field-lhs-assign-FVi-R 2 (<internal>::#7), 0 (Conn::c), 0, 2 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:245
content_gap 12 268977 0.103678 has-field-cond-VVV 0 (Conn::c), 8, 25 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:246-250
```

```

        }

        if ( |c$service| > 0 )
            c$conn$service=to_lower(join_string_set(c$service, ","));

        c$conn$conn_state=conn_state(c, get_port_transport_proto(c$id$resp_p));

        if ( c$history != "" )
            c$conn$history=c$history;
    }
}

event content_gap(c: connection, is_orig: bool, seq: count, length: count) &priority=5
{
    set_conn(c, F);

    c$conn$missed_bytes = c$conn$missed_bytes + length;
}

event tunnel_changed(c: connection, e: EncapsulatingConnVector) &priority=5
{
    set_conn(c, F);
    if ( |e| > 0 )
    {
        if ( ! c$conn?$tunnel_parents )
            c$conn$tunnel_parents = set();
        add c$conn$tunnel_parents[e[|e|-1]$uid];
    }
    c$tunnel = e;
}
/Users/vern/zeek/zeek-master.06Feb24/scripts/base/protocols/conn/main.zeek: unmodified: line 288

```

```
content_gap CPU time 10.175484, 268977 calls, 4887437 instructions
content_gap 0 268977 0.119125 load-val-VV-R 0 (Conn::c), interpreter frame[0] // Conn::content_gap;Conn::content_gap:286-290
content_gap 1 268977 0.103368 load-val-VV-U 1 (Conn::length), interpreter frame[3] // Conn::content_gap;Conn::content_gap:286-290
content_gap 2 268977 0.103948 not-has-field-cond-VVV 0 (Conn::c), 15, 10 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242
content_gap 3 6488 0.003557 field-RVi-R 2 (<internal>::#2), 0 (Conn::c), 0 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 4 6488 0.003948 field-RVi-U 3 (<internal>::#3), 2 (<internal>::#2), 3 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 5 6488 0.002480 get-port-transport-proto-VV 3 (Conn::p.2), 3 (<internal>::#3) // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 6 6488 0.002433 field-RVi-D 4 (<internal>::#4), 0 (Conn::c), 3 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 7 6488 0.003177 field-RVi-S 2 (<internal>::#5), 0 (Conn::c), 7 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 8 6488 0.044150 construct-known-record-V 2 (<internal>::#6) // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 9 6488 0.003821 field-lhs-assign-FV-R 0 (Conn::c), 2 (<internal>::#6), 15 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 10 268977 0.116475 field-RVi-R 2 (<internal>::#7), 0 (Conn::c), 15 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:245
content_gap 11 268977 0.138658 field-lhs-assign-FVi-R 2 (<internal>::#7), 0 (Conn::c), 0, 2 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:245
content_gap 12 268977 0.103678 has-field-cond-VVV 0 (Conn::c), 8, 25 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:246-250
```

```
content_gap CPU time 10.175484, 268977 calls, 4887437 instructions
content_gap 0 268977 0.119125 load-val-VV-R 0 (Conn::c), interpreter frame[0] // Conn::content_gap;Conn::content_gap:286-290
content_gap 1 268977 0.103368 load-val-VV-U 1 (Conn::length), interpreter frame[3] // Conn::content_gap;Conn::content_gap:286-290
content_gap 2 268977 0.103948 not-has-field-cond-VVW 0 (Conn::c), 15, 10 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242
content_gap 3 6488 0.003557 field-RVi-R 2 (<internal>::#2), 0 (Conn::c), 0 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 4 6488 0.003948 field-RVi-U 3 (<internal>::#3), 2 (<internal>::#2), 3 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 5 6488 0.002480 get-port-transport-proto-VV 3 (Conn::p.2), 3 (<internal>::#3) // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 6 6488 0.002433 field-RVi-D 4 (<internal>::#4), 0 (Conn::c), 3 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 7 6488 0.003177 field-RVi-S 2 (<internal>::#5), 0 (Conn::c), 7 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 8 6488 0.044150 construct-known-record-V 2 (<internal>::#6) // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 9 6488 0.003821 field-lhs-assign-FV-R 0 (Conn::c), 2 (<internal>::#6), 15 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 10 268977 0.116475 field-RVi-R 2 (<internal>::#7), 0 (Conn::c), 15 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:245
content_gap 11 268977 0.138658 field-lhs-assign-FVi-R 2 (<internal>::#7), 0 (Conn::c), 0, 2 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:245
content_gap 12 268977 0.103678 has-field-cond-VVW 0 (Conn::c), 8, 25 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:246-250
```

```
content_gap CPU time 10.175484, 268977 calls, 4887437 instructions
content_gap 0 268977 0.119125 load-val-VV-R 0 (Conn::c), interpreter frame[0] // Conn::content_gap;Conn::content_gap:286-290
content_gap 1 268977 0.103368 load-val-VV-U 1 (Conn::length), interpreter frame[3] // Conn::content_gap;Conn::content_gap:286-290
content_gap 2 268977 0.103948 not-has-field-cond-VVV 0 (Conn::c), 15, 10 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242
content_gap 3 6488 0.003557 field-RVi-R 2 (<internal>::#2), 0 (Conn::c), 0 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 4 6488 0.003948 field-RVi-U 3 (<internal>::#3), 2 (<internal>::#2), 3 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 5 6488 0.002480 get-port-transport-proto-VV 3 (Conn::p.2), 3 (<internal>::#3) // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:241
content_gap 6 6488 0.002433 field-RVi-D 4 (<internal>::#4), 0 (Conn::c), 3 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 7 6488 0.003177 field-RVi-S 2 (<internal>::#5), 0 (Conn::c), 7 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 8 6488 0.044150 construct-known-record-V 2 (<internal>::#6) // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 9 6488 0.003821 field-lhs-assign-FV-R 0 (Conn::c), 2 (<internal>::#6), 15 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:239-242;Conn::set_conn:242
content_gap 10 268977 0.116475 field-RVi-R 2 (<internal>::#7), 0 (Conn::c), 15 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:245
content_gap 11 268977 0.138658 field-lhs-assign-FVi-R 2 (<internal>::#7), 0 (Conn::c), 0, 2 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:245
content_gap 12 268977 0.103678 has-field-cond-VVV 0 (Conn::c), 8, 25 // Conn::content_gap;Conn::content_gap:286-290;Conn::content_gap:288;Conn::set_conn;Conn::set_conn:237-283;Conn::set_conn:246-250
```

```

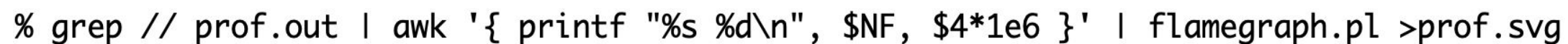
        else
            return rs == UDP_ACTIVE ? "SHR" : "OTH";
        }

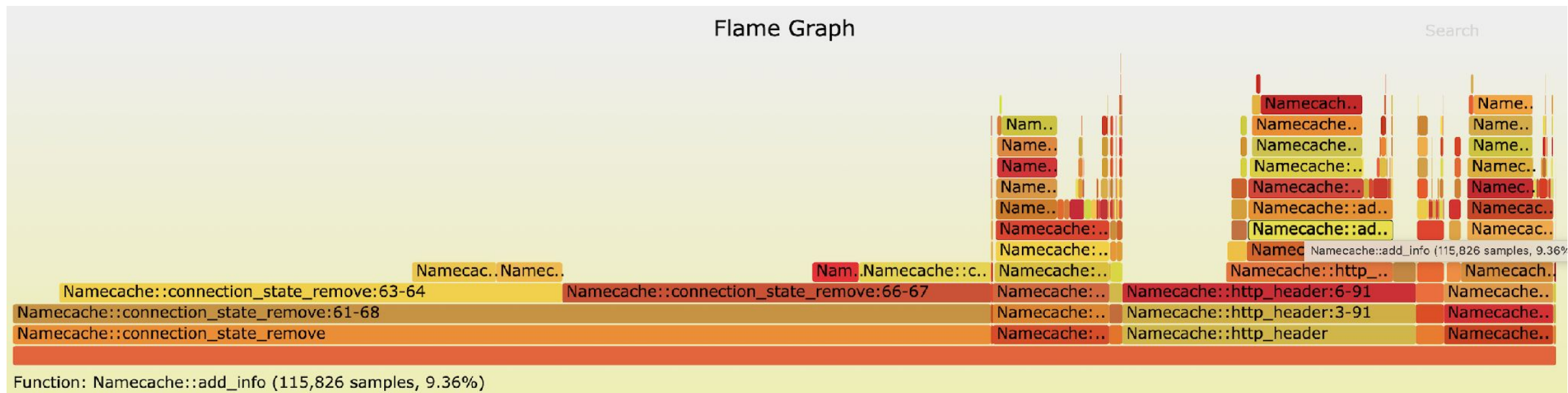
    else
        return "OTH";
    }

## Fill out the c$conn record for logging
function set_conn(c: connection, eoc: bool)
{
    if ( ! c?$conn )
    {
        local p = get_port_transport_proto(c$id$resp_p);
        c$conn = Info($ts=c$start_time, $uid=c$uid, $proto=p);
    }

    c$conn$id=c$id:
    if ( c?$tunnel && |c$tunnel| > 0 )
    {
        if ( ! c$conn?$tunnel_parents )
            c$conn$tunnel_parents = set();
        add c$conn$tunnel_parents[c$tunnel[|c$tunnel|-1]$uid];
    }
    if( |Site::local_nets| > 0 )
    {
        c$conn$local_orig=Site::is_local_addr(c$id$orig_h);
        c$conn$local_resp=Site::is_local_addr(c$id$resp_h);
    }
}

```





```
% grep // prof.out | grep Namecache | awk '{ printf "%s %d\n", $NF, $4 * 1e6 }' | flamegraph.pl >nc.prof.svg
```

Speeding Up Zeek Scripts

1. Using traditional Zeek scripting
 - a) Avoiding repeated expressions
 - b) Comparing perf. with *micro-benchmarks* / hyperfine
 - c) Locating perf. issues w/ `zeek --profile-scripts`
2. Using Zeek script optimization
 - a) ZAM
 - b) Using ZAM to produce *flamegraphs*
 - c) **Compiling to C++**

Compiling scripts to C++

Compiling scripts to C++

- Leverages native C++ optimization

```
532
533 event connection_pause(c: connection, was_orig_restart: bool, dt: interval)
534     {
535         ++statistics_info$total_pauses_seen;
536         ++statistics_info$interval_pauses_seen;
537
538         if ( is_white_listed(c) )
539             return;
540
541         local conn_data = ConnData($ts=network_time(), $id=c$id,
542                                     $start_time=c$start_time, $uid=c$uid);
543
544         if ( ! is_worker )
545             { # Handle locally.
546                 handle_connection_pause(conn_data, was_orig_restart, dt);
547                 return;
548             }
549
550         if ( |sent_to_manager| > max_conns_worker && c$id !in sent_to_manager )
551             {
552                 #@ BEGIN-NOT-TESTED
553                 ++statistics_info$total_max_conn_pauses_discarded;
554
555                 local discarded_max =
556                     statistics_info$total_max_conn_pauses_discarded;
557                 if ( discarded_max > 0 && discarded_max % 10000 == 0 )
558                     Reporter::warning(fmt("Discarded %s/%s pauses (max_conns reached)",
559                                           discarded_max, statistics_info$total_pauses_seen));
560                 return;
561                 #@ END-NOT-TESTED
562             }
563
564         add sent_to_manager[c$id];
565
566         local e = Broker::make_event(SteppingStones::manager_connection_pause,
567                                     conn_data, was_orig_restart, dt);
568         Broker::publish(Cluster::manager_topic, e);
569     }
```

```

532
533 event connection_pause(c: connection, was_orig_restart: bool)
534 {
535     ++statistics_info$total_pauses_seen;
536     ++statistics_info$interval_pauses_seen;
537
538     if ( is_white_listed(c) )
539         return;
540
541     local conn_data = ConnData($ts=network_time(), $id=c$id,
542                               $start_time=c$start_time);
543
544     if ( ! is_worker )
545     { # Handle locally.
546         handle_connection_pause(conn_data, was_orig_restart);
547         return;
548     }
549
550     if ( |sent_to_manager| > max_conns_worker && c$id != "" )
551     {
552         #@ BEGIN-NOT-TESTED
553         ++statistics_info$total_max_conn_pauses_discarded;
554
555         local discarded_max =
556             statistics_info$total_max_conn_pauses_discarded;
557         if ( discarded_max > 0 && discarded_max % 100 == 0 )
558             Reporter::warning(fmt("Discarded %s conn pauses",
559                                   discarded_max, statistics_info$total_max_conn_pauses_discarded));
560         return;
561         #@ END-NOT-TESTED
562     }
563
564     add sent_to_manager[c$id];
565
566     local e = Broker::make_event(SteppingStones::manager_connection_pause__gl_ev,
567                                 conn_data, was_orig_restart);
568     Broker::publish(Cluster::manager_topic, e);
569 }

```

```

void scripts_Corelight_SteppingStones_main_zEEK__connection_pause__zf(const RecordValPtr& c_
, bool was_orig_restart_, double dt_, Frame* f__CPP)
{
{
    static bool did_init = false;
    if ( ! did_init )
    {
        if ( event_registry->Lookup("SteppingStones::manager_connection_pause") )
        {
            SteppingStones__manager_connection_pause__gl_ev = event_registry->Register("SteppingStones::manager_connection_pause");
        }
        did_init = true;
    }
}
zeek_uint_t discarded_max_;
RecordValPtr e_;
RecordValPtr conn_data_;

// ./scripts/Corelight/SteppingStones/main.zEEK:535
cast_intrusive<RecordVal>(SteppingStones__statistics_info__gl->GetVal())->Assign(0,
(field_access_zeek_uint_t__CPP(cast_intrusive<RecordVal>(SteppingStones__statistics_info__gl->GetVal()), 0))+1));
// ./scripts/Corelight/SteppingStones/main.zEEK:536
cast_intrusive<RecordVal>(SteppingStones__statistics_info__gl->GetVal())->Assign(1,
(field_access_zeek_uint_t__CPP(cast_intrusive<RecordVal>(SteppingStones__statistics_info__gl->GetVal()), 1))+1));
// ./scripts/Corelight/SteppingStones/main.zEEK:538
if ( scripts_Corelight_SteppingStones_main_zEEK__SteppingStones__is_white_listed__zf(c_, f__CPP) )
{
    // ./scripts/Corelight/SteppingStones/main.zEEK:539
    return;
}
// ./scripts/Corelight/SteppingStones/main.zEEK:541

```

```

532
533 event connection_pause(c: connection, was_orig_restart: bool)
534 {
535     ++statistics_info$total_pauses_seen;
536     ++statistics_info$interval_pauses_seen;
537
538     if ( is_white_listed(c) )
539         return;
540
541     local conn_data = ConnData($ts=network_time(), $id=c$id,
542                               $start_time=c$start_time);
543
544     if ( ! is_worker )
545     { # Handle locally.
546         handle_connection_pause(conn_data, was_orig_restart);
547         return;
548     }
549
550     if ( |sent_to_manager| > max_conns_worker && c$id != $id )
551     {
552         #@ BEGIN-NOT-TESTED
553         ++statistics_info$total_max_conn_pauses_discarded;
554
555         local discarded_max =
556             statistics_info$total_max_conn_pauses_discarded;
557         if ( discarded_max > 0 && discarded_max % 100 == 0 )
558             Reporter::warning(fmt("Discarded %s conn pauses",
559                                   discarded_max, statistics_info$total_max_conn_pauses_discarded));
560         return;
561         #@ END-NOT-TESTED
562     }
563
564     add sent_to_manager[c$id];
565
566     local e = Broker::make_event(SteppingStones::manager_connection_pause,
567                                conn_data, was_orig_restart);
568     Broker::publish(Cluster::manager_topic, e);
569 }

```

```

void scripts_Corelight_SteppingStones_main_zEEK__connection_pause__zf(const RecordValPtr& c_
, bool was_orig_restart_, double dt_, Frame* f__CPP)
{
{
    static bool did_init = false;
    if ( ! did_init )
    {
        if ( event_registry->Lookup("SteppingStones::manager_connection_pause") )
        {
            SteppingStones__manager_connection_pause__gl_ev = event_registry->Register("SteppingStones::manager_connection_pause");
        }
        did_init = true;
    }
}

zeek_uint_t discarded_max_;
RecordValPtr e_;
RecordValPtr conn_data_;

// ./scripts/Corelight/SteppingStones/main.zEEK:535
cast_intrusive<RecordVal>(SteppingStones__statistics_info__gl->GetVal())->Assign(0,
(field_access_zeek_uint_t__CPP(cast_intrusive<RecordVal>(SteppingStones__statistics_info__gl->GetVal()), 0))+1));
// ./scripts/Corelight/SteppingStones/main.zEEK:536
cast_intrusive<RecordVal>(SteppingStones__statistics_info__gl->GetVal())->Assign(1,
(field_access_zeek_uint_t__CPP(cast_intrusive<RecordVal>(SteppingStones__statistics_info__gl->GetVal()), 1))+1));
// ./scripts/Corelight/SteppingStones/main.zEEK:538
if ( scripts_Corelight_SteppingStones_main_zEEK__SteppingStones__is_white_listed__zf(c_, f__CPP) )
{
    // ./scripts/Corelight/SteppingStones/main.zEEK:539
    return;
}
// ./scripts/Corelight/SteppingStones/main.zEEK:541

```

Compiling scripts to C++

- Leverages native C++ optimization
- Automatically provides script-level profiling via C++ tools

Compiling scripts to C++

- Leverages native C++ optimization
- Automatically provides script-level profiling via C++ tools
- Usual workflow: compile a set of scripts to a Zeek plugin
(also doable for packages that already have plugins)

Compiling scripts to C++

- Leverages native C++ optimization
- Automatically provides script-level profiling via C++ tools
- Usual workflow: compile a set of scripts to a Zeek plugin (also doable for packages that already have plugins)
- Another workflow: compile Zeek's default scripts into C++

Compiling scripts to C++

- Leverages native C++ optimization
- Automatically provides script-level profiling via C++ tools
- Usual workflow: compile a set of scripts to a Zeek plugin (also doable for packages that already have plugins)
- Another workflow: compile Zeek's default scripts into C++
- **Compatibility: full (other than some error message formats)**
 - However: will skip in presence of `@ifdef cond. code`

Speeding up Zeek Scripts - How Much Faster?

Speeding up Zeek Scripts - How Much Faster?

- Hugely depends on what a script does ...
... and *Amdahl's Law*

Speeding up Zeek Scripts - How Much Faster?

- Hugely depends on what a script does ...
... and *Amdahl's Law*
- Typical **total** run-time gain: ~5-10%

Speeding up Zeek Scripts - How Much Faster?

- Hugely depends on what a script does ...
... and *Amdahl's Law*
- Typical **total** run-time gain: ~5-10%
- Gets better with the more packages you use!

HOW TO MAKE YOUR ZEEK SCRIPTS RUN FASTER

Vern Paxson, Co-Founder and Chief Scientist, Corelight Inc.

Professor of the Graduate School, UC Berkeley

vern@corelight.com

Zeek Workshop | September 10, 2026


```
function foo(url: string)
{
  if ( ends_with(url, ".google.com") || ends_with(url, ".google.net") || ends_with(url, ".google.co") )
    set_url_category(url, "google");

  else if ( ends_with(url, ".fb.com") || ends_with(url, ".fb.net") || ends_with(url, ".fb.co") )
    set_url_category(url, "facebook");

  else if ( ends_with(url, ".aws.com") || ends_with(url, ".aws.net") || ends_with(url, ".aws.co") )
    set_url_category(url, "amazon");

  else if ( ends_with(url, ".azure.com") || ends_with(url, ".azure.net") || ends_with(url, ".azure.co") )
    set_url_category(url, "azure");

  else if ...

}
```

```
function foo(url: string)
{
    if ( ends_with(url, ".google.com") || ends_with(url, ".google.net") || ends_with(url, ".google.co") )
        set_url_category(url, "google");

    else if ( ends_with(url, ".fb.com") || ends_with(url, ".fb.net") || ends_with(url, ".fb.co") )
        set_url_category(url, "facebook");

    else if ( ends_with(url, ".aws.com") || ends_with(url, ".aws.net") || ends_with(url, ".aws.co") )
        set_url_category(url, "amazon");

    else if ( ends_with(url, ".azure.com") || ends_with(url, ".azure.net") || ends_with(url, ".azure.co") )
        set_url_category(url, "azure");

    else if ...
}
```

```
function foo(url: string)
{
    if ( url == /*\google\.(com|net|co)/ )
        set_url_category(url, "google");

    else if ( url == /*\fb\.(com|net|co)/ )
        set_url_category(url, "facebook");

    else if ( url == /*\aws\.(com|net|co)/ )
        set_url_category(url, "amazon");

    else if ( url == /*\azure\.(com|net|co)/ )
        set_url_category(url, "azure");

    else if ...

}
```

```
function foo(url: string)
{
    if ( ends_with(url, ".google.com") || ends_with(url, ".google.net") || ends_with(url, ".google.co") )
        set_url_category(url, "google");

    else if ( ends_with(url, ".fb.com") || ends_with(url, ".fb.net") || ends_with(url, ".fb.co") )
        set_url_category(url, "facebook");

    else if ( ends_with(url, ".aws.com") || ends_with(url, ".aws.net") || ends_with(url, ".aws.co") )
        set_url_category(url, "amazon");

    else if ( ends_with(url, ".azure.com") || ends_with(url, ".azure.net") || ends_with(url, ".azure.co") )
        set_url_category(url, "azure");
```

else if ...

```
const url_categories = table(
    [ /*.*\google\.(com|net|co)/ ] = "google",
    [ /*.*\fb\.(com|net|co)/ ] = "facebook",
    [ /*.*\aws\.(com|net|co)/ ] = "amazon",
    [ /*.*\azure\.(com|net|co)/ ] = "azure",
);
```

```
function foo(url: string)
{
    local category = url_categories[url];
    if ( |category| > 0 )
        set_url_category(url, category[0]);
}
```

```
function foo(url: string)
```

```
{
    if ( url == /*.*\google\.(com|net|co)/ )
        set_url_category(url, "google");

    else if ( url == /*.*\fb\.(com|net|co)/ )
        set_url_category(url, "facebook");

    else if ( url == /*.*\aws\.(com|net|co)/ )
        set_url_category(url, "amazon");

    else if ( url == /*.*\azure\.(com|net|co)/ )
        set_url_category(url, "azure");

    else if ...

}
```

Speeding Up Zeek Scripts

1. For script writers
 - a) Avoiding repeated expressions
 - b) Profiling to find issues

2. For script users
 - a) ZAM optimization
 - b) Comparing performance with *hyperfine*
 - c) Using ZAM to produce *flamegraphs*
 - d) Compiling to C++

What About
Function Call Overhead?

```
perf-examples % cat record-ref10.zeek
```

```
type r1: record { a: count; };
```

```
type r2: record { b: r1; };
```

```
event zeek_init()
```

```
{
```

```
  local x = r2($b = r1($a = 0));
```

```
  local sum = +0; local i = 0;
```

```
  while ( ++i <= 10000000 )
```

```
  {
```

```
    if ( x$b$a > 10 ) sum += 10;
```

```
    else if ( x$b$a > 5 ) sum += 5;
```

```
    else if ( x$b$a > 3 ) sum += 3;
```

```
    else if ( x$b$a > 1 ) ++sum;
```

```
    else --sum;
```

```
  }
```

```
  print sum;
```

```
}
```

Version of our original script
that does 10M iterations

```
perf-examples % cat record-ref10.zee
```

```
type r1: record { a: count; };
```

```
type r2: record { b: r1; };
```

```
event zee_init()
```

```
{
```

```
  local x = r2($b = r1($a = 0));
```

```
  local sum = +0; local i = 0;
```

```
  while ( ++i <= 10000000 )
```

```
  {
```

```
    if ( x$b$a > 10 ) sum += 10;
```

```
    else if ( x$b$a > 5 ) sum += 5;
```

```
    else if ( x$b$a > 3 ) sum += 3;
```

```
    else if ( x$b$a > 1 ) ++sum;
```

```
    else --sum;
```

```
  }
```

```
  print sum;
```

```
}
```

Same, except uses a function
call to compute them

```
perf-examples % cat record-ref10.call.zee
```

```
type r1: record { a: count; };
```

```
type r2: record { b: r1; };
```

```
function sum_it(v: r2): int
```

```
{
```

```
  local sum = +0;
```

```
  if ( v$b$a > 10 ) sum += 10;
```

```
  else if ( v$b$a > 5 ) sum += 5;
```

```
  else if ( v$b$a > 3 ) sum += 3;
```

```
  else if ( v$b$a > 1 ) ++sum;
```

```
  else --sum;
```

```
  return sum;
```

```
}
```

```
event zee_init()
```

```
{
```

```
  local x = r2($b = r1($a = 0));
```

```
  local sum = +0; local i = 0;
```

```
  while ( ++i <= 10000000 )
```

```
    sum += sum_it(x);
```

```
  print sum;
```

```
}
```

```
[perf-examples % hyperfine 'zeek -b record-ref10.zeek' 'zeek -b record-ref10.call  
.zeek'
```

```
perf-examples % hyperfine 'zeek -b record-ref10.zeek' 'zeek -b record-ref10.call.zeek'
```

Benchmark 1: zeek -b record-ref10.zeek

Time (mean $\pm$ σ): 19.703 s $\pm$ 0.242 s [User: 19.517 s, System: 0.131 s]
Range (min ... max): 19.409 s ... 20.238 s 10 runs

Benchmark 2: zeek -b record-ref10.call.zeek

Time (mean $\pm$ σ): 35.341 s $\pm$ 0.490 s [User: 35.157 s, System: 0.128 s]
Range (min ... max): 34.863 s ... 36.210 s 10 runs

Summary

zeek -b record-ref10.zeek ran

1.79 $\pm$ 0.03 times faster than zeek -b record-ref10.call.zeek

Significant function call overhead
(when doing millions!)

```
perf-examples % hyperfine 'zeek -b -0 ZAM record-ref10.zeek' 'zeek -b -0 ZAM record-ref10.call.zeek'
```

How about for ZAM?

```
perf-examples % hyperfine 'zeek -b -0 ZAM record-ref10.zeek' 'zeek -b -0 ZAM record-ref10.call.zeek'
```

Benchmark 1: zeek -b -0 ZAM record-ref10.zeek

Time (mean $\pm$ σ): 1.629 s $\pm$ 0.009 s [User: 1.538 s, System: 0.082 s]
Range (min ... max): 1.619 s ... 1.645 s 10 runs

Benchmark 2: zeek -b -0 ZAM record-ref10.call.zeek

Time (mean $\pm$ σ): 1.874 s $\pm$ 0.006 s [User: 1.785 s, System: 0.082 s]
Range (min ... max): 1.865 s ... 1.881 s 10 runs

Summary

zeek -b -0 ZAM record-ref10.zeek ran

1.15 $\pm$ 0.01 times faster than zeek -b -0 ZAM record-ref10.call.zeek

Much more modest
(ideally, inliner would make it 0%)

```
perf-examples % hyperfine 'zeek -b record-ref10.call.zeek' 'zeek -b -O ZAM record-ref10.zeek'
```

Let's finish with a full comparison,
regular vs. ZAM, 10M iterations ...

```
perf-examples % hyperfine 'zeek -b record-ref10.call.zeek' 'zeek -b -0 ZAM record-ref10.zeek'
```

Benchmark 1: zeek -b record-ref10.call.zeek

Time (mean $\pm$ σ): 35.554 s $\pm$ 0.203 s [User: 35.266 s, System: 0.191 s]

Range (min ... max): 35.306 s ... 35.934 s 10 runs

Benchmark 2: zeek -b -0 ZAM record-ref10.zeek

Time (mean $\pm$ σ): 1.666 s $\pm$ 0.024 s [User: 1.566 s, System: 0.090 s]

Range (min ... max): 1.641 s ... 1.722 s 10 runs

Summary

zeek -b -0 ZAM record-ref10.zeek ran

21.34 $\pm$ 0.33 times faster than zeek -b record-ref10.call.zeek

#fields	function	location	type	ncall	tot_CPU	child_CPU	tot_Mem	child_Mem
#types	string	string	string	count	interval	interval	count	count
total	322-locations	TOTAL	639456	4.048020	0.000000	15335424	0	
non-scripts	<no-location>	TOTAL	258260	8.933789	0.000000	133365760	0	
all-BiFs	65-locations	BiF	130510	0.348761	0.000000	6307840	0	
SSL::set_session	protocols/ssl/main.zeek:204	function	79429	0.126430	0.016436	65536	0	
file_over_new_connection	9-locations	event	32571	0.093994	0.000000	0	0	
ssl_plaintext_data	protocols/ssl/main.zeek:538	event	29291	0.128121	0.030683	0	0	
SSL::add_to_history	protocols/ssl/main.zeek:213	function	29291	0.175915	0.036123	0	0	
analyzer_confirmation_info	6-locations	event	22752	0.135853	0.000000	114688	0	
to_lower	init-bare.zeek:6850	BiF	22057	0.026889	0.000000	0	0	
ssl_handshake_message	protocols/ssl/main.zeek:376	event	22057	0.287833	0.156010	0	0	
file_state_remove	6-locations	event	21714	0.150583	0.000000	163840	0	
file_sniff	6-locations	event	21714	0.472209	0.000000	425984	0	
file_hash	2-locations	event	21714	0.055546	0.000000	0	0	
ssl_established	5-locations	event	18940	0.196962	0.000000	163840	0	
Files::set_info	frameworks/files/main.zeek:347	function	14476	0.081698	0.000000	16384	0	
Files::analyzer_name	frameworks/files/main.zeek:448	function	14476	0.053029	0.017845	16384	16384	
Files::analyzer_enabled	frameworks/files/				0.021002	0	0	
Files::add_analyzer	frameworks/files/main.zeek:3404				114688	49152		
Files::__analyzer_name	init-frameworks-and-bifs.zeek:32	BiF	14476	0.017845	0.000000	16384	0	
Files::__analyzer_enabled	init-frameworks-and-bifs.zeek:32	BiF	14476	0.021002	0.000000	0	0	
Files::__add_analyzer	init-frameworks-and-bifs.zeek:32	BiF	14476	0.083714	0.000000	32768	0	
connection_state_remove	3-locations	event	11379	0.220437	0.000000	81920	0	
Log::write	frameworks/logging/main.zeek:1003	function	11277	0.119019	0.088300	458752	458752	
Log::__write	init-frameworks-and-bifs.zeek:32	BiF	11277	0.088300	0.009150	458752	0	
Site::is_local_addr	utils/site.zeek:194	function	11205	0.017372	0.000000	0	0	
to_upper	init-bare.zeek:6850	BiF	11026	0.013930	0.000000	0	0	
file_hash	files/x509/certificate-event-cache.zeek:100	event	10857	0.032775	0.000000	0	0	
file_hash	files/hash/main.zeek:19	event	10857	0.022771	0.000000	0	0	
new_connection	2-locations	event	7586	0.080705	0.000000	147456	0	
get_port_transport_proto	init-bare.zeek:6850	BiF	7586	0.011069	0.000000	16384	0	
connection_established	2-locations	event	7586	0.020229	0.000000	0	0	
Conn::set_conn	protocols/conn/main.zeek:244	function	7586	0.168133	0.059701	147456	16384	
ssl_server_hello	2-locations	event	7584	0.039292	0.000000	0	0	
ssl_client_hello	2-locations	event	7584	0.024450	0.000000	0	0	
SSL::log_record	protocols/ssl/main.zeek:240	function	7580	0.061827	0.041825	163840	163840	
SSL::finish	protocols/ssl/main.zeek:267	function	7580	0.109984	0.074076	163840	163840	

sort -nr -k4 : most calls first

#fields	function	location	type	ncall	tot_CPU	child_CPU	tot_Mem	child_Mem
#types	string	string	count	interval	interval	count	count	
file_sniff	6-locations	event	21714	0.472209	0.000000	425984	0	
file_sniff	frameworks/files/main.zeek:554	event	3619	0.417712	0.339639	114688	114688	
Files::add_analyzer	frameworks/files/main.zeek:415	function			14476	0.321759	0.193404	114688 49152
file_state_remove	6-locations	event	21714	0.150583	0.000000	163840	0	
file_over_new_connection	9-locations	event	32571	0.093994	0.000000	0	0	
file_new	frameworks/files/main.zeek:534	event	3619	0.085667	0.059141	32768	32768	
Files::__add_analyzer	init-frameworks-and-bifs.zeek:32	BiF			14476	0.083714	0.000000	32768 0
Files::set_info	frameworks/files/main.zeek:347	function			14476	0.081698	0.000000	16384 0
file_state_remove	frameworks/files/main.zeek:583	event	3619	0.066363	0.032818	0	0	
Files::analyzer_enabled	frameworks/files/main.zeek:410	function			14476	0.056661	0.021002	0 0
file_hash	2-locations	event	21714	0.055546	0.000000	0	0	
Files::analyzer_name	frameworks/files/main.zeek:448	function			14476	0.053029	0.017845	16384 16384
x509_certificate	files/x509/main.zeek:213	event	3619	0.047529	0.020248	262144	262144	
file_over_new_connection	frameworks/files/main.zeek:545	event	3619	0.045210	0.023310	0	0	
SSL::ssl_finishing	protocols/ssl/files.zeek:161	hook	3792	0.039774	0.000000	0	0	
file_sniff	protocols/ssl/files.zeek:126	event	3619	0.035886	0.000000	311296	0	
file_state_remove	files/x509/main.zeek:303	event	3619	0.035473	0.011179	163840	163840	
file_hash	files/x509/certificate							
file_state_remove	frameworks/fil							
file_hash	files/hash/main.zeek:1							
Files::__analyzer_enabled	init-f							
Files::__analyzer_name	init-framework							
Files::enable_reassembly	framew							
Files::set_reassembly_buffer_size	frameworks/files/main.zeek:395	function			3619	0.014951	0.004997	16384 16384
file_state_remove	files/x509/certificate-event-cache.zeek:84	event	3619	0.011482	0.000000	0	0	
X509::create_deduplication_index	files/x509/main.zeek:205	hook	3619	0.010725	0.000000	0	0	
X509::x509_certificate_cache_replay	files/x509/certificate-event-cache.zeek:53	hook	3609	0.009783	0.000000	0	0	
file_over_new_connection	protocols/ftp/files.zeek:50	event	3619	0.009575	0.000000	0	0	
Files::log_policy	files/x509/main.zeek:199	hook	3619	0.009150	0.000000	0	0	
file_over_new_connection	protocols/irc/files.zeek:32	event	3619	0.008383	0.000000	0	0	
file_over_new_connection	protocols/http/entities.zeek:105	event	3619	0.006506	0.000000	0	0	
file_over_new_connection	protocols/krb/files.zeek:75	event	3619	0.005841	0.000000	0	0	
Files::__enable_reassembly	init-frameworks-and-bifs.zeek:32	BiF	3619	0.005673	0.000000	0	0	
file_sniff	protocols/http/entities.zeek:166	event	3619	0.005420	0.000000	0	0	
Files::__set_reassembly_buffer	init-frameworks-and-bifs.zeek:32	BiF	3619	0.004997	0.000000	16384	0	
file_over_new_connection	protocols/smtp/entities.zeek:37	event	3619	0.004990	0.000000	0	0	
file_state_remove	protocols/smb/main.zeek:249	event	3619	0.004859	0.000000	0	0	

sort -nr -k5 | grep files:
functions etc. relating to file extraction with most CPU

```
ZAM-prof % awk 'NF == 3' prof.out | sort -nr -k2 | head -28
```

```
field-RVi-R      7967908 3.868006
load-val-VV-R    6210445 3.304763
has-field-cond-VVV      3964927 2.110147
field-RVi-U      3562537 1.637321
load-global-VV-T      3059054 1.618656
goto-V  2403554 0.751501
field-RVi-S      2239168 1.113405
if-VV  2228240 1.130058
if-not-VV      2204416 0.805226
field-RVi-T      2188778 1.395426
field-RVi-A      2057921 0.925958
load-val-VV-S    1571471 0.631652
not-has-field-cond-VVV 1427434 0.636913
return-X      1407866 0.528629
load-val-VV-I    1373366 0.531527
if-else-VV      1294917 0.411416
val-is-in-table-cond-VVV      1284116 2.231540
index-vec-VVV    1190026 0.605244
load-global-VV-I    1127531 0.843917
field-RVi-D      1044804 0.462559
lt-VCV-cond-U    1028826 0.000000
next-vector-iter-VVV    1026272 0.398328
eq-VCV-cond-I    987319 0.000000
construct-known-record-V      952094 4.663268
val-is-in-table-VVV    909478 1.117178
table-index1-VVV-R    853300 2.522969
assign-VV-R      849907 0.450144
eq-VVC-cond-U    823635 0.000000
```

388 instructions

```
ZAM-prof % awk 'NF == 3' prof.out | sort -nr -k3 | head -28
```

log-wrotec-V	146159	8.068651
construct-known-record-V	952094	4.663268
field-RVi-R	7967908	3.868006
load-val-VV-R	6210445	3.304763
next-table-iter-VV	765332	2.946889
val-is-not-in-table-cond-VV	820597	2.930559
init-record-V	457917	2.861222
table-index1-VV-R	853300	2.522969
table-elem-assign-VV	374831	2.354646
val-is-in-table-cond-VV	1284116	2.231540
has-field-cond-VV	3964927	2.110147
call2-X 72030	1.720522	
field-RVi-U	3562537	1.637321
load-global-VV-T	3059054	1.618656
call3-V-I	145065	1.448343
field-RVi-T	2188778	1.395426
call3-V-S	340925	1.392862
addstmt1-VV	293642	1.185722
call1-V 249115	1.150504	
if-VV 2228240	1.130058	
val-is-in-table-VV	909478	1.117178
field-RVi-S	2239168	1.113405
call1-VV-I	261490	1.074082
deltabale-V0	164205	1.044427
call2-V-U	125223	0.950687
field-RVi-A	2057921	0.925958
load-global-VV-I	1127531	0.843917
if-not-VV _	2204416	0.805226