# Sparse Warnings

Daniel Axtens

linux.conf.au 2017 Kernel Miniconf

## Using sparse

```
make C=2 ...
```

## Where we're going

## Simple static analysis

- Undefined/unwise behaviour:

```
warning: expression using sizeof bool
warning: odd constant _Bool cast (ffffffffffffff becomes 1)
```

- Odd accesses:

```
warning: invalid access past the end of 's32' (12 8)
```

- Static suggestions:

```
warning: symbol 'ppc_fadvise64_64' was not declared. \
Should it be static?
```

## What does sparse understand?

sparse + annotations ⇒ understanding more than the C compiler alone:

- Endinaness of variables
- Address space of pointers
- Pointers that should not be dereferenced
- Types needing explicit conversion
- and (probably) more...

## Base types

```
unsigned int instr;
instr = cpu_to_le32(instr);
```

## Base types

```
unsigned int instr;
instr = cpu_to_le32(instr);
```

---

```
warning: incorrect type in assignment (different base types)
   expected unsigned int [unsigned] [assigned] instr
   got restricted __le32 [usertype] <noident>
```

## Address spaces

```
unsigned long pc;
int instr;

probe_kernel_address((unsigned int __user *)pc, instr);
```

## Address spaces

```
unsigned long pc;
int instr;

probe_kernel_address((unsigned int __user *)pc, instr);
```

---

```
 warning: incorrect type in argument 2 (different address spaces)
    expected void const *src
    got unsigned int [noderef] <asn:1>*<noident>
```

## Address spaces

```
struct rt_sigframe __user *frame;

printk_ratelimited(regs->msr & MSR_64BIT ? fmt64 : fmt32,
                   tsk->comm, tsk->pid, "setup_rt_frame",
                   (long)frame, regs->nip, regs->link);
```

## Address spaces

```
struct rt_sigframe __user *frame;

printk_ratelimited(regs->msr & MSR_64BIT ? fmt64 : fmt32,
                   tsk->comm, tsk->pid, "setup_rt_frame",
                   (long)frame, regs->nip, regs->link);
```

---

```
warning: cast removes address space of expression
```

## But wait, there's more!

- no cast types

```
arch/powerpc/kernel/time.c:361:37: warning: implicit cast to nocast type
arch/powerpc/kernel/time.c:362:29: warning: implicit cast to nocast type
```

- no dereference pointers (e.g. IO)

```
arch/powerpc/kernel/io.c:40:24: warning: dereference of noderef expression
arch/powerpc/kernel/io.c:56:18: warning: dereference of noderef expression
```

- restricted types

```
arch/powerpc/sysdev/mpic.c:356:18: warning: cast to restricted __le32
```

# #fail

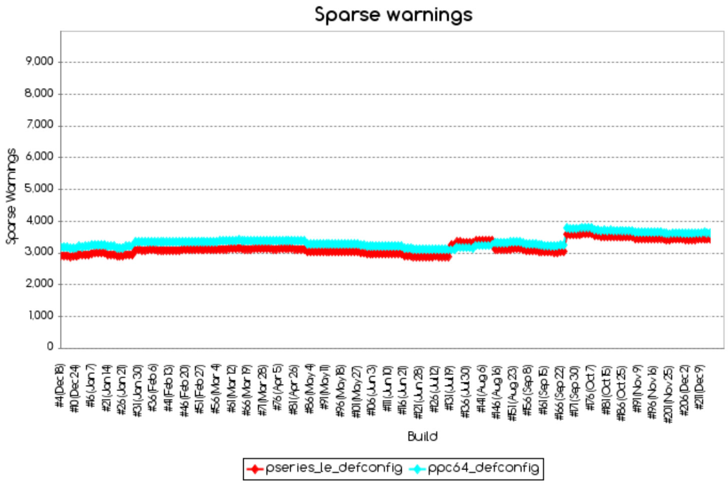

Figure: Sparse warnings, 2 ppc defconfigs, 2016

## In the long term...

Fix the warnings

## PowerPC warnings

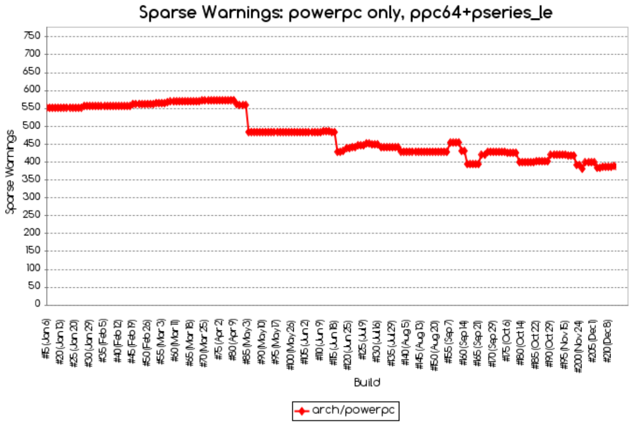

Figure: `arch/powerpc` sparse warnings, combination of 2 defconfigs, 2016

# PowerPC warnings

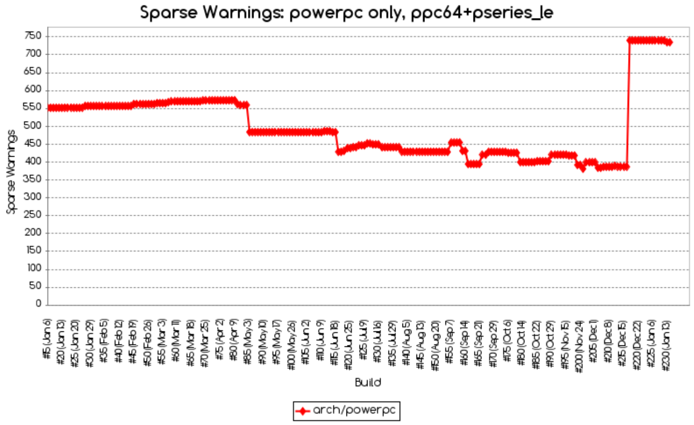

Figure: arch/powerpc sparse warnings, combination of 2 defconfigs, 2016-present

## Total warnings

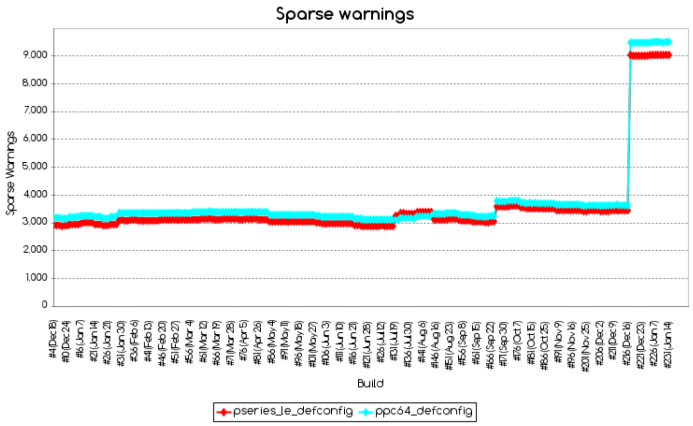

Figure: Sparse warnings, 2 ppc defconfigs, 2016-present

## What went wrong? https://patchwork.kernel.org/patch/9467371/

**linux/types.h: enable endian checks for all sparse builds**

```
By now, linux is mostly endian-clean. Enabling endian-ness
checks for everyone produces about 200 new sparse warnings for me -
less than 10% over the 2000 sparse warnings already there.

Not a big deal, OTOH enabling this helps people notice
they are introducing new bugs.

So let's just drop __CHECK_ENDIAN__. Follow-up patches
can drop distinction between __bitwise and __bitwise__.

Cc: Linus Torvalds <torvalds@linux-foundation.org>
Suggested-by: Christoph Hellwig <hch@infradead.org>
Signed-off-by: Michael S. Tsirkin <mst@redhat.com>
```

## diff <before> <after>

```
arch/powerpc/kernel/nvram_64.c:1177:32: warning: cast to restricted __be16

arch/powerpc/kernel/nvram_64.c:893:22: warning: incorrect type in assignment \
    (different base types)
    expected unsigned short [unsigned] [addressable] length
    got restricted __be16 [usertype] <noident>

arch/powerpc/kvm/book3s_64_vio_hv.c:282:37: warning: cast to restricted __be64

arch/powerpc/kvm/book3s_hv_builtin.c:421:22: warning: incorrect type in assignment \
    (different base types)
    expected restricted __be32 [addressable] [usertype] xirr
    got unsigned int

arch/powerpc/perf/hv-24x7.c:1166:18: warning: cast to restricted __be64
```

## In the long term...

Fix the warnings

## In the short term...

**How do you diff compiler warnings/sparse output?**

- `grep 'arch/powerpc' sparse-output | wc -l`
  - Simple, low fidelity.
- `diff sparse-output-1 sparse-output-2`
  - Reordering due to parallel builds
  - Changing line numbers
- Write our own.

## Introducing `smart-sparse-diff`

`https://github.com/daxtens/smart-sparse-diff`

## Where to from here

- **Fix sparse warnings in your code**
- When they have reached an acceptably low quantity
  - Contributors: don't add warnings
  - Maintainers: require it of contributors
  - ML reviewers: evaluate it in your reviews

## Sparse as a gateway to kernel development

- Pick a file with sparse warnings
- How do patches go into that file?
    - What mailing list?
    - What subject line format?
- When doing patches:
    - Don't fix just one of many, fix all of one type
    - Compile (and if appropriate, run) test!
- Consult docs on submitting patches:
    - Commit messages
    - How to send a non-broken email
- Expect bike-shedding.

## Conclusion

## Legal