Lustre Static Code Analysis with Coverity

April 17th, 2013

Sebastien Buisson

Parallel File Systems
Extreme Computing R&D

Static Code Analysis with Coverity

□ Why static code analysis is useful?

□ Tool for analysis: Coverity

□ Coverity applied to Lustre

□ Defects found by Coverity

□ Benefits for the whole Lustre Community

Why static code analysis is useful?

Architect of an Open World™

Why static code analysis is useful?

From https://wiki.hpdd.intel.com/display/PUB/Project+Ideas

“Run Lustre code through static analysis tools to identify potential latent bugs in the Lustre code. These are often hard to find through testing, and easily fixed once found.”

From W. S. Humphrey, "Using a Defined and Measured Personal Software Process," IEEE Software, May, 1996

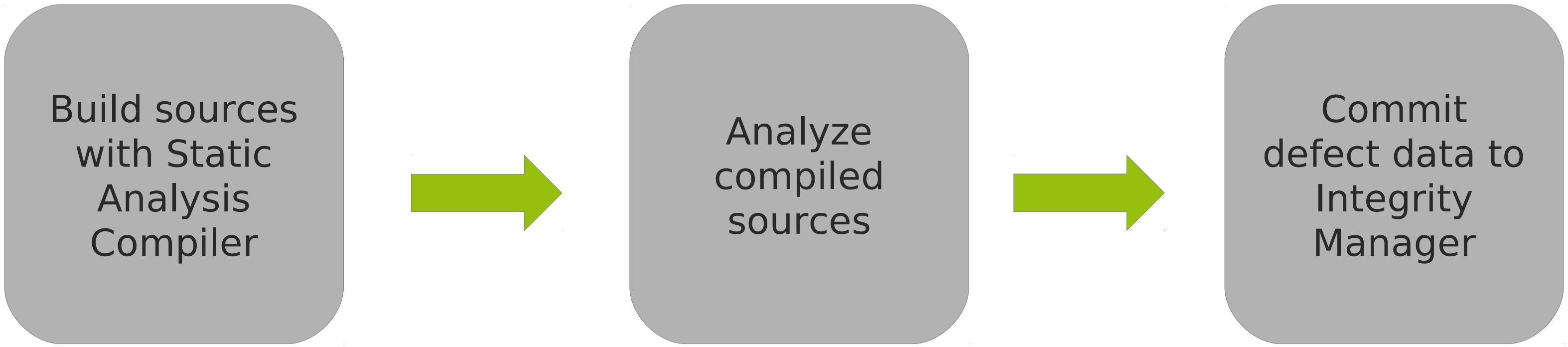
“Even experienced programmers typically make a mistake for every seven to ten lines of code they develop.”

Tool for analysis: Coverity

Architect of an Open World™

Tool for analysis: Coverity

□ How it works



Build sources with Static Analysis Compiler

Analyze compiled sources

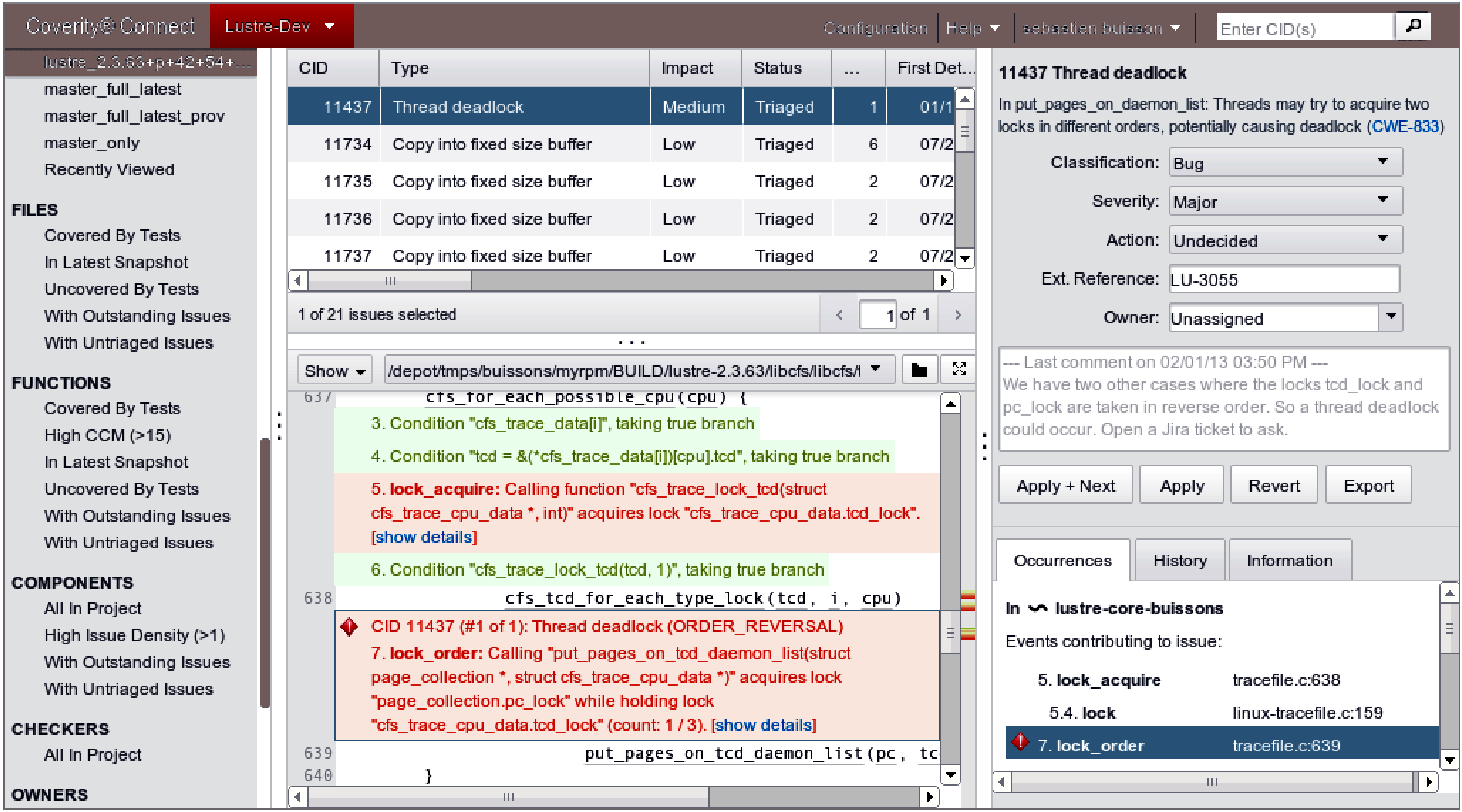
Commit defect data to Integrity Manager

□ Various checkers involved:

- STRING_SIZE
- RESOURCE_LEAK
- UNINIT
- and many others...

Tool for analysis: Coverity

- What do we get



CID	Type	Impact	Status	...	First Det...
11437	Thread deadlock	Medium	Triaged	1	01/1
11734	Copy into fixed size buffer	Low	Triaged	6	07/2
11735	Copy into fixed size buffer	Low	Triaged	2	07/2
11736	Copy into fixed size buffer	Low	Triaged	2	07/2
11737	Copy into fixed size buffer	Low	Triaged	2	07/2

1 of 21 issues selected

1 of 1

Show /depot/tmps/buissons/myrpm/BUILD/lustre-2.3.63/libcfs/libcfs/l

```
        cfs_for_each_possible_cpu(cpu) {
```

3. Condition "cfs_trace_data[i]", taking true branch

4. Condition "tcd = &(*cfs_trace_data[i])[cpu].tcd", taking true branch

5. **lock_acquire:** Calling function "cfs_trace_lock_tcd(struct cfs_trace_cpu_data *, int)" acquires lock "cfs_trace_cpu_data.tcd_lock". [show details]

6. Condition "cfs_trace_lock_tcd(tcd, 1)", taking true branch

```
                cfs_tcd_for_each_type_lock(tcd, i, cpu)
```

CID 11437 (#1 of 1): Thread deadlock (ORDER_REVERSAL)

7. **lock_order:** Calling "put_pages_on_tcd_daemon_list(struct page_collection *, struct cfs_trace_cpu_data *)" acquires lock "page_collection.pc_lock" while holding lock "cfs_trace_cpu_data.tcd_lock" (count: 1 / 3). [show details]

```
                        put_pages_on_tcd_daemon_list(pc, tc
        }
```

11437 Thread deadlock

In put_pages_on_daemon_list: Threads may try to acquire two locks in different orders, potentially causing deadlock (CWE-833)

Classification: Bug

Severity: Major

Action: Undecided

Ext. Reference: LU-3055

Owner: Unassigned

--- Last comment on 02/01/13 03:50 PM ---
We have two other cases where the locks tcd_lock and pc_lock are taken in reverse order. So a thread deadlock could occur. Open a Jira ticket to ask.

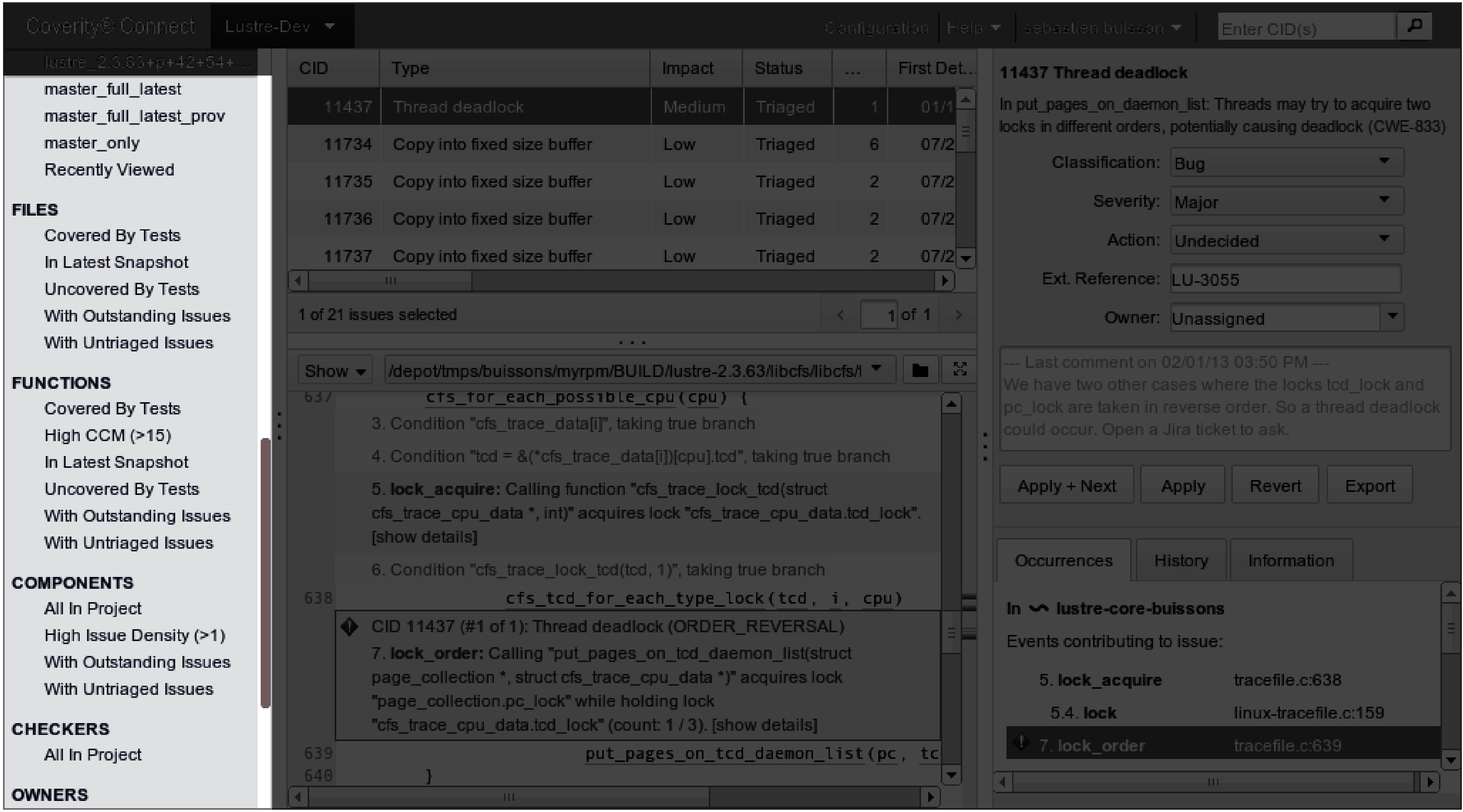
In lustre-core-buissons

Events contributing to issue:

5. **lock_acquire**	tracefile.c:638
5.4. **lock**	linux-tracefile.c:159
7. **lock_order**	tracefile.c:639

Tool for analysis: Coverity

- What do we get



lustre_2.3.63+p+42+54+...
master_full_latest
master_full_latest_prov
master_only
Recently Viewed

FILES
Covered By Tests
In Latest Snapshot
Uncovered By Tests
With Outstanding Issues
With Untriaged Issues

FUNCTIONS
Covered By Tests
High CCM (>15)
In Latest Snapshot
Uncovered By Tests
With Outstanding Issues
With Untriaged Issues

COMPONENTS
All In Project
High Issue Density (>1)
With Outstanding Issues
With Untriaged Issues

CHECKERS
All In Project

OWNERS

CID	Type	Impact	Status	...	First Det...
11437	Thread deadlock	Medium	Triaged	1	01/1
11734	Copy into fixed size buffer	Low	Triaged	6	07/2
11735	Copy into fixed size buffer	Low	Triaged	2	07/2
11736	Copy into fixed size buffer	Low	Triaged	2	07/2
11737	Copy into fixed size buffer	Low	Triaged	2	07/2

1 of 21 issues selected   1 of 1

Show ▾   /depot/tmps/buissons/myrpm/BUILD/lustre-2.3.63/libcfs/libcfs/l

```
637                 cfs_for_each_possible_cpu(cpu) {
        3. Condition "cfs_trace_data[i]", taking true branch
        4. Condition "tcd = &(*cfs_trace_data[i])[cpu].tcd", taking true branch
        5. lock_acquire: Calling function "cfs_trace_lock_tcd(struct
        cfs_trace_cpu_data *, int)" acquires lock "cfs_trace_cpu_data.tcd_lock".
        [show details]
        6. Condition "cfs_trace_lock_tcd(tcd, 1)", taking true branch
638                         cfs_tcd_for_each_type_lock(tcd, i, cpu)
        CID 11437 (#1 of 1): Thread deadlock (ORDER_REVERSAL)
        7. lock_order: Calling "put_pages_on_tcd_daemon_list(struct
        page_collection *, struct cfs_trace_cpu_data *)" acquires lock
        "page_collection.pc_lock" while holding lock
        "cfs_trace_cpu_data.tcd_lock" (count: 1 / 3). [show details]
639                                 put_pages_on_tcd_daemon_list(pc, tc
640                 }
```

11437 Thread deadlock

In put_pages_on_daemon_list: Threads may try to acquire two locks in different orders, potentially causing deadlock (CWE-833)

Classification: Bug
Severity: Major
Action: Undecided
Ext. Reference: LU-3055
Owner: Unassigned

--- Last comment on 02/01/13 03:50 PM ---
We have two other cases where the locks tcd_lock and pc_lock are taken in reverse order. So a thread deadlock could occur. Open a Jira ticket to ask.

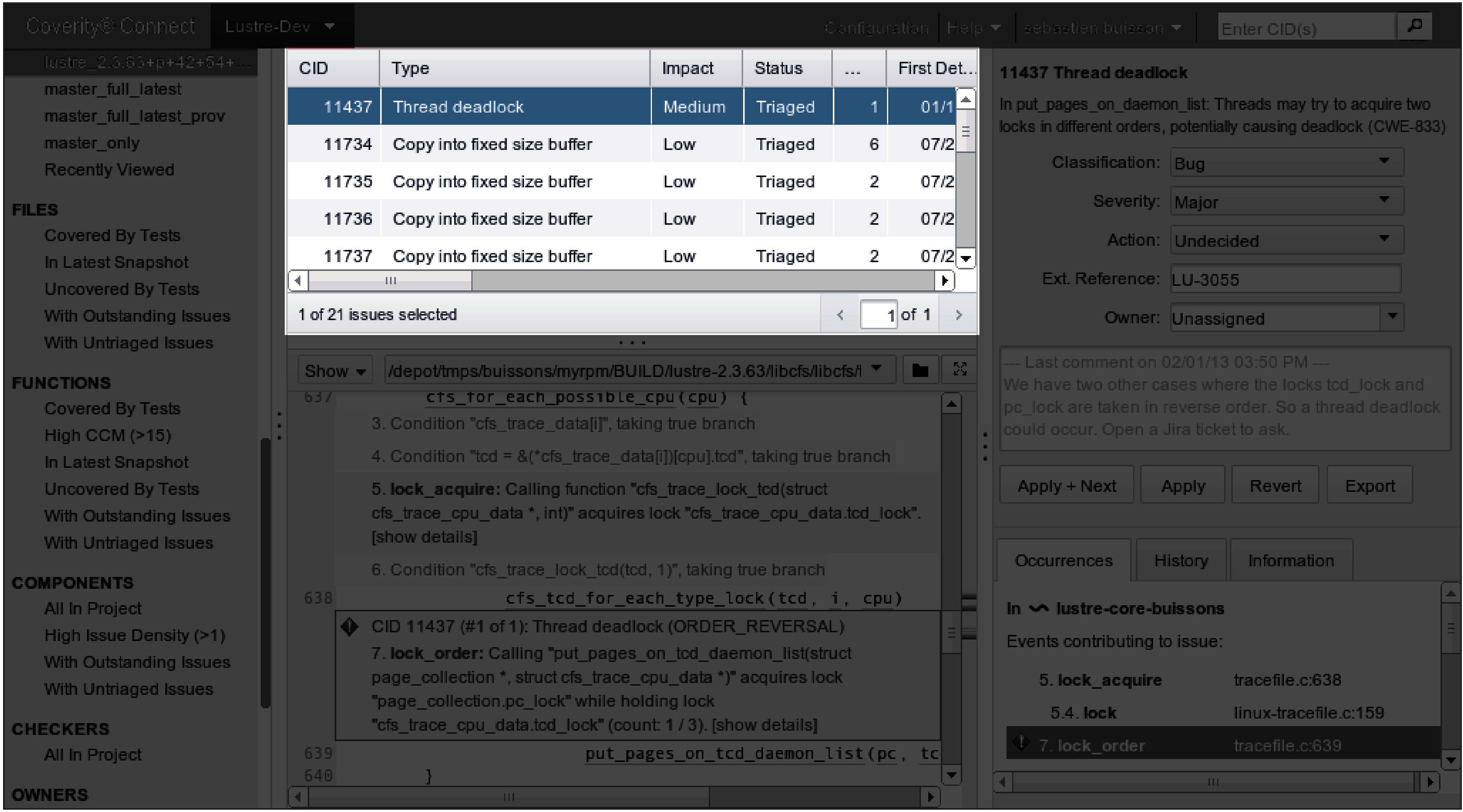
In **lustre-core-buissons**

Events contributing to issue:

5. **lock_acquire**	tracefile.c:638
5.4. **lock**	linux-tracefile.c:159
7. **lock_order**	tracefile.c:639

Tool for analysis: Coverity

- ▢ What do we get



CID	Type	Impact	Status	...	First Det...
11437	Thread deadlock	Medium	Triaged	1	01/1
11734	Copy into fixed size buffer	Low	Triaged	6	07/2
11735	Copy into fixed size buffer	Low	Triaged	2	07/2
11736	Copy into fixed size buffer	Low	Triaged	2	07/2
11737	Copy into fixed size buffer	Low	Triaged	2	07/2

1 of 21 issues selected

/depot/tmps/buissons/myrpm/BUILD/lustre-2.3.63/libcfs/libcfs/l

```
637        cfs_for_each_possible_cpu(cpu) {
```

3. Condition "cfs_trace_data[i]", taking true branch

4. Condition "tcd = &(*cfs_trace_data[i])[cpu].tcd", taking true branch

5. **lock_acquire:** Calling function "cfs_trace_lock_tcd(struct cfs_trace_cpu_data *, int)" acquires lock "cfs_trace_cpu_data.tcd_lock". [show details]

6. Condition "cfs_trace_lock_tcd(tcd, 1)", taking true branch

```
638                cfs_tcd_for_each_type_lock(tcd, i, cpu)
```

CID 11437 (#1 of 1): Thread deadlock (ORDER_REVERSAL)

7. **lock_order:** Calling "put_pages_on_tcd_daemon_list(struct page_collection *, struct cfs_trace_cpu_data *)" acquires lock "page_collection.pc_lock" while holding lock "cfs_trace_cpu_data.tcd_lock" (count: 1 / 3). [show details]

```
639                        put_pages_on_tcd_daemon_list(pc, tc
640        }
```

11437 Thread deadlock

In put_pages_on_daemon_list: Threads may try to acquire two locks in different orders, potentially causing deadlock (CWE-833)

Classification: Bug

Severity: Major

Action: Undecided

Ext. Reference: LU-3055

Owner: Unassigned

--- Last comment on 02/01/13 03:50 PM ---
We have two other cases where the locks tcd_lock and pc_lock are taken in reverse order. So a thread deadlock could occur. Open a Jira ticket to ask.

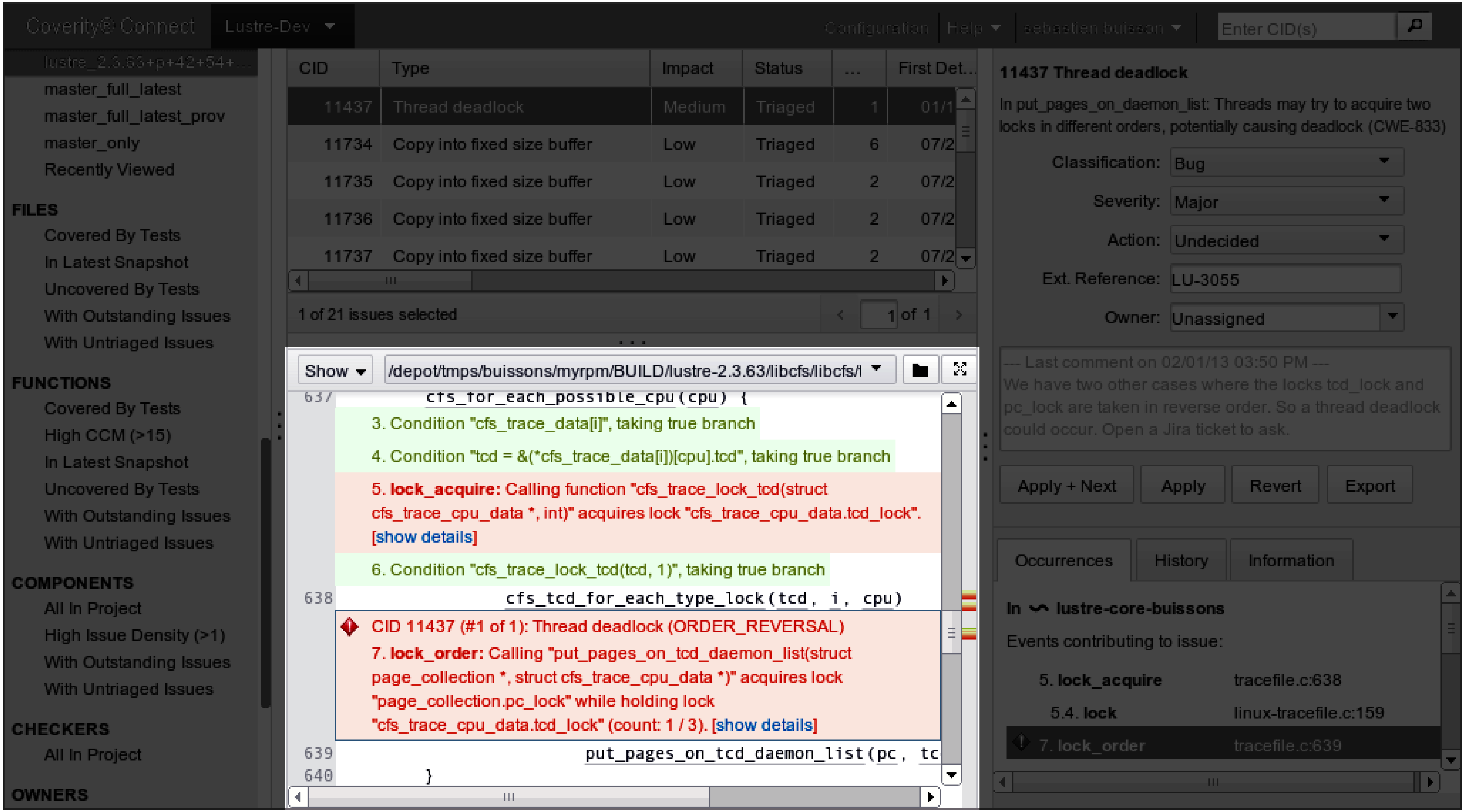
In lustre-core-buissons

Events contributing to issue:

5. **lock_acquire**	tracefile.c:638
5.4. **lock**	linux-tracefile.c:159
7. lock_order	tracefile.c:639

Tool for analysis: Coverity

- □ What do we get



lustre_2.3.63+p+42+54+...
master_full_latest
master_full_latest_prov
master_only
Recently Viewed

FILES
Covered By Tests
In Latest Snapshot
Uncovered By Tests
With Outstanding Issues
With Untriaged Issues

FUNCTIONS
Covered By Tests
High CCM (>15)
In Latest Snapshot
Uncovered By Tests
With Outstanding Issues
With Untriaged Issues

COMPONENTS
All In Project
High Issue Density (>1)
With Outstanding Issues
With Untriaged Issues

CHECKERS
All In Project

OWNERS

CID	Type	Impact	Status	...	First Det...
11437	Thread deadlock	Medium	Triaged	1	01/1
11734	Copy into fixed size buffer	Low	Triaged	6	07/2
11735	Copy into fixed size buffer	Low	Triaged	2	07/2
11736	Copy into fixed size buffer	Low	Triaged	2	07/2
11737	Copy into fixed size buffer	Low	Triaged	2	07/2

1 of 21 issues selected   1 of 1

Show ▾ /depot/tmps/buissons/myrpm/BUILD/lustre-2.3.63/libcfs/libcfs/l

```
637         cfs_for_each_possible_cpu(cpu) {
```

3. Condition "cfs_trace_data[i]", taking true branch

4. Condition "tcd = &(*cfs_trace_data[i])[cpu].tcd", taking true branch

5. **lock_acquire:** Calling function "cfs_trace_lock_tcd(struct cfs_trace_cpu_data *, int)" acquires lock "cfs_trace_cpu_data.tcd_lock". [show details]

6. Condition "cfs_trace_lock_tcd(tcd, 1)", taking true branch

```
638                 cfs_tcd_for_each_type_lock(tcd, i, cpu)
```

CID 11437 (#1 of 1): Thread deadlock (ORDER_REVERSAL)

7. **lock_order:** Calling "put_pages_on_tcd_daemon_list(struct page_collection *, struct cfs_trace_cpu_data *)" acquires lock "page_collection.pc_lock" while holding lock "cfs_trace_cpu_data.tcd_lock" (count: 1 / 3). [show details]

```
639                         put_pages_on_tcd_daemon_list(pc, tc
640         }
```

11437 Thread deadlock

In put_pages_on_daemon_list: Threads may try to acquire two locks in different orders, potentially causing deadlock (CWE-833)

Classification: Bug
Severity: Major
Action: Undecided
Ext. Reference: LU-3055
Owner: Unassigned

--- Last comment on 02/01/13 03:50 PM ---
We have two other cases where the locks tcd_lock and pc_lock are taken in reverse order. So a thread deadlock could occur. Open a Jira ticket to ask.

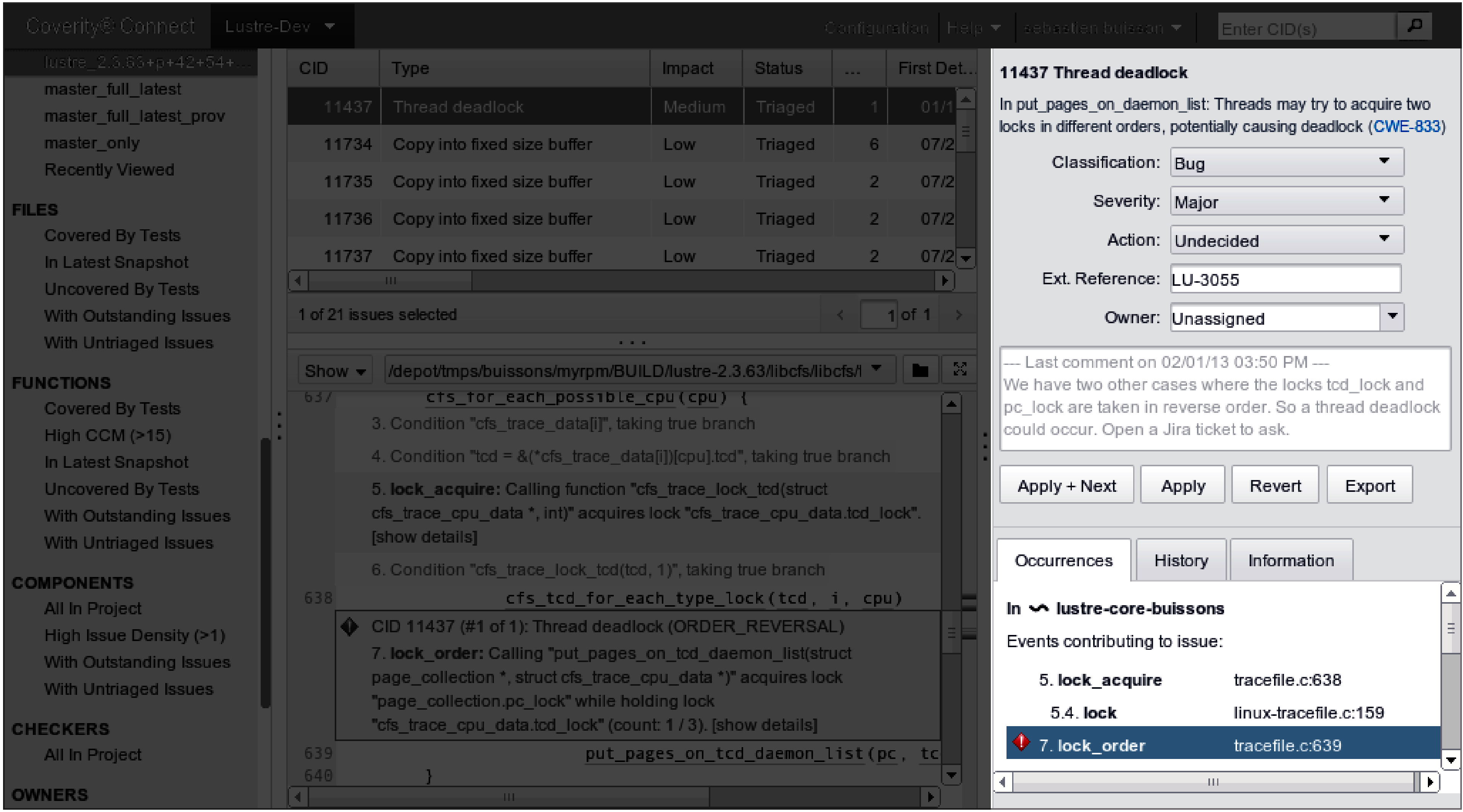
In lustre-core-buissons

Events contributing to issue:

5. **lock_acquire**	tracefile.c:638
5.4. **lock**	linux-tracefile.c:159
7. **lock_order**	tracefile.c:639

Tool for analysis: Coverity

- What do we get



CID	Type	Impact	Status	...	First Det...
11437	Thread deadlock	Medium	Triaged	1	01/1
11734	Copy into fixed size buffer	Low	Triaged	6	07/2
11735	Copy into fixed size buffer	Low	Triaged	2	07/2
11736	Copy into fixed size buffer	Low	Triaged	2	07/2
11737	Copy into fixed size buffer	Low	Triaged	2	07/2

1 of 21 issues selected 1 of 1

/depot/tmps/buissons/myrpm/BUILD/lustre-2.3.63/libcfs/libcfs/l

```
        cfs_for_each_possible_cpu(cpu) {
```

3. Condition "cfs_trace_data[i]", taking true branch

4. Condition "tcd = &(*cfs_trace_data[i])[cpu].tcd", taking true branch

5. **lock_acquire:** Calling function "cfs_trace_lock_tcd(struct cfs_trace_cpu_data *, int)" acquires lock "cfs_trace_cpu_data.tcd_lock". [show details]

6. Condition "cfs_trace_lock_tcd(tcd, 1)", taking true branch

```
                cfs_tcd_for_each_type_lock(tcd, i, cpu)
```

CID 11437 (#1 of 1): Thread deadlock (ORDER_REVERSAL)

7. **lock_order:** Calling "put_pages_on_tcd_daemon_list(struct page_collection *, struct cfs_trace_cpu_data *)" acquires lock "page_collection.pc_lock" while holding lock "cfs_trace_cpu_data.tcd_lock" (count: 1 / 3). [show details]

```
                        put_pages_on_tcd_daemon_list(pc, tc
        }
```

11437 Thread deadlock

In put_pages_on_daemon_list: Threads may try to acquire two locks in different orders, potentially causing deadlock (CWE-833)

Classification: Bug

Severity: Major

Action: Undecided

Ext. Reference: LU-3055

Owner: Unassigned

--- Last comment on 02/01/13 03:50 PM ---
We have two other cases where the locks tcd_lock and pc_lock are taken in reverse order. So a thread deadlock could occur. Open a Jira ticket to ask.

In lustre-core-buissons

Events contributing to issue:

5. **lock_acquire**	tracefile.c:638
5.4. **lock**	linux-tracefile.c:159
7. **lock_order**	tracefile.c:639

Tool for analysis: Coverity

Defect categories

- API usage errors
- Code maintainability issues
- Concurrent data access violations
- Control flow issues
- Error handling issues
- Incorrect expression
- Integer handling issues
- Memory - corruptions
- Memory - illegal accesses
- Null pointer dereferences
- Program hangs
- Resource leaks
- Security best practices violations
- Uninitialized variables

Coverity applied to Lustre

Architect of an Open World™

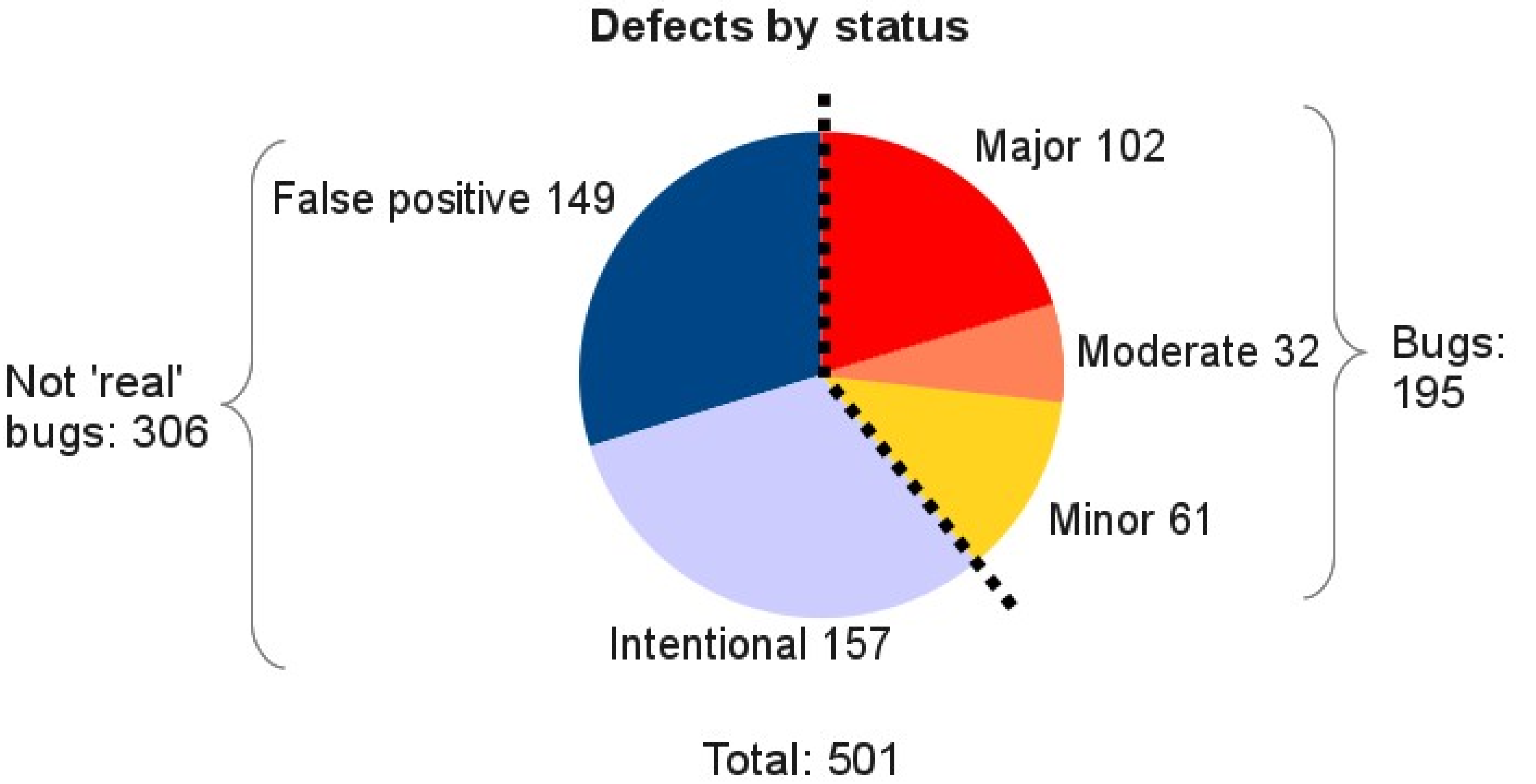
Coverity applied to Lustre

□ How we proceed

- Main work on Lustre Master branch + 2.1 in parallel
- Current status for master:
  - Latest tag analyzed: v2_3_63 (March 22nd)
  - Next steps: diff with new tags until 2.4 GA

Defects found by Coverity

- Statistics on master

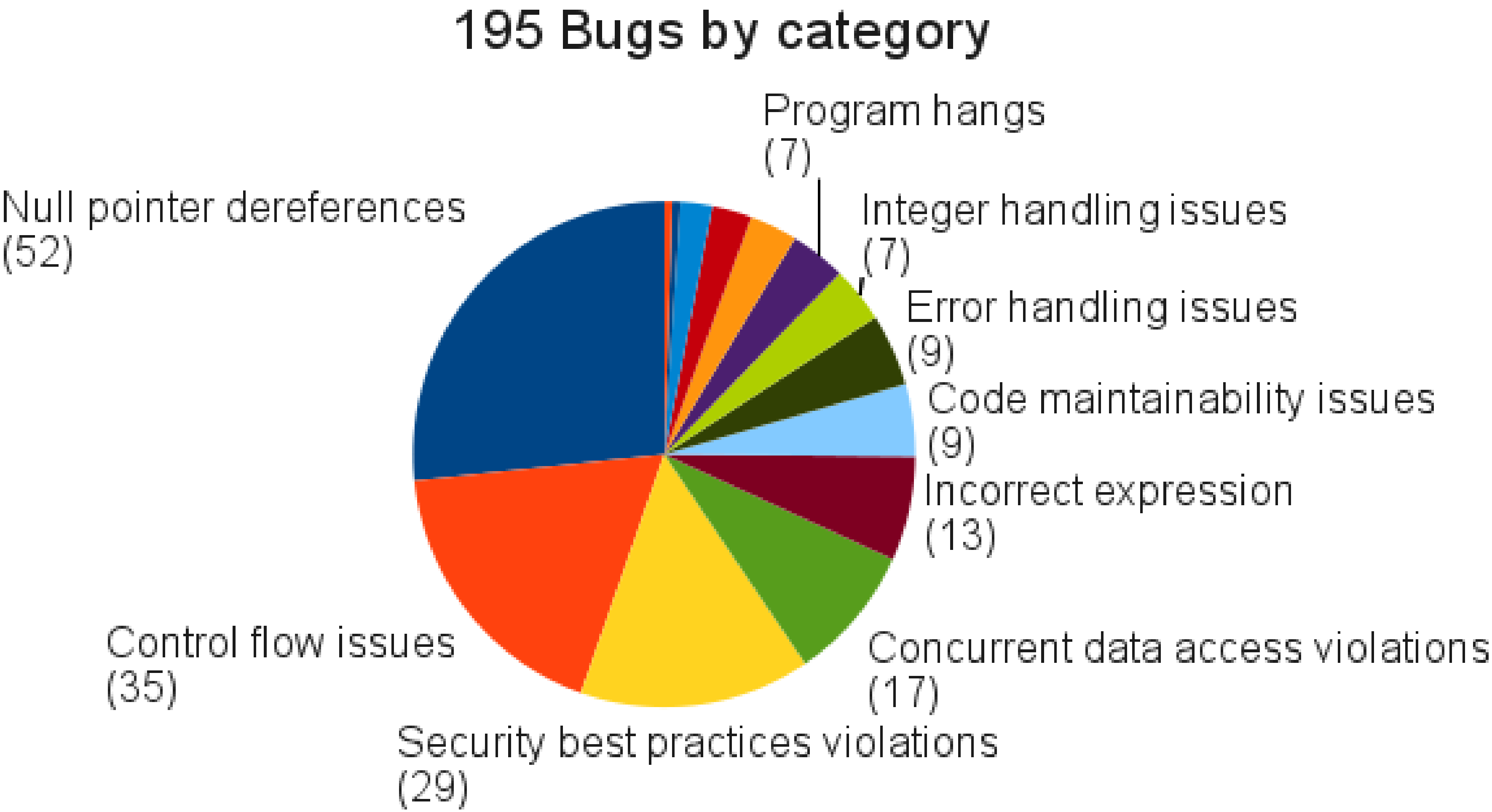


Defects found by Coverity

A few words on false positives and intentionals

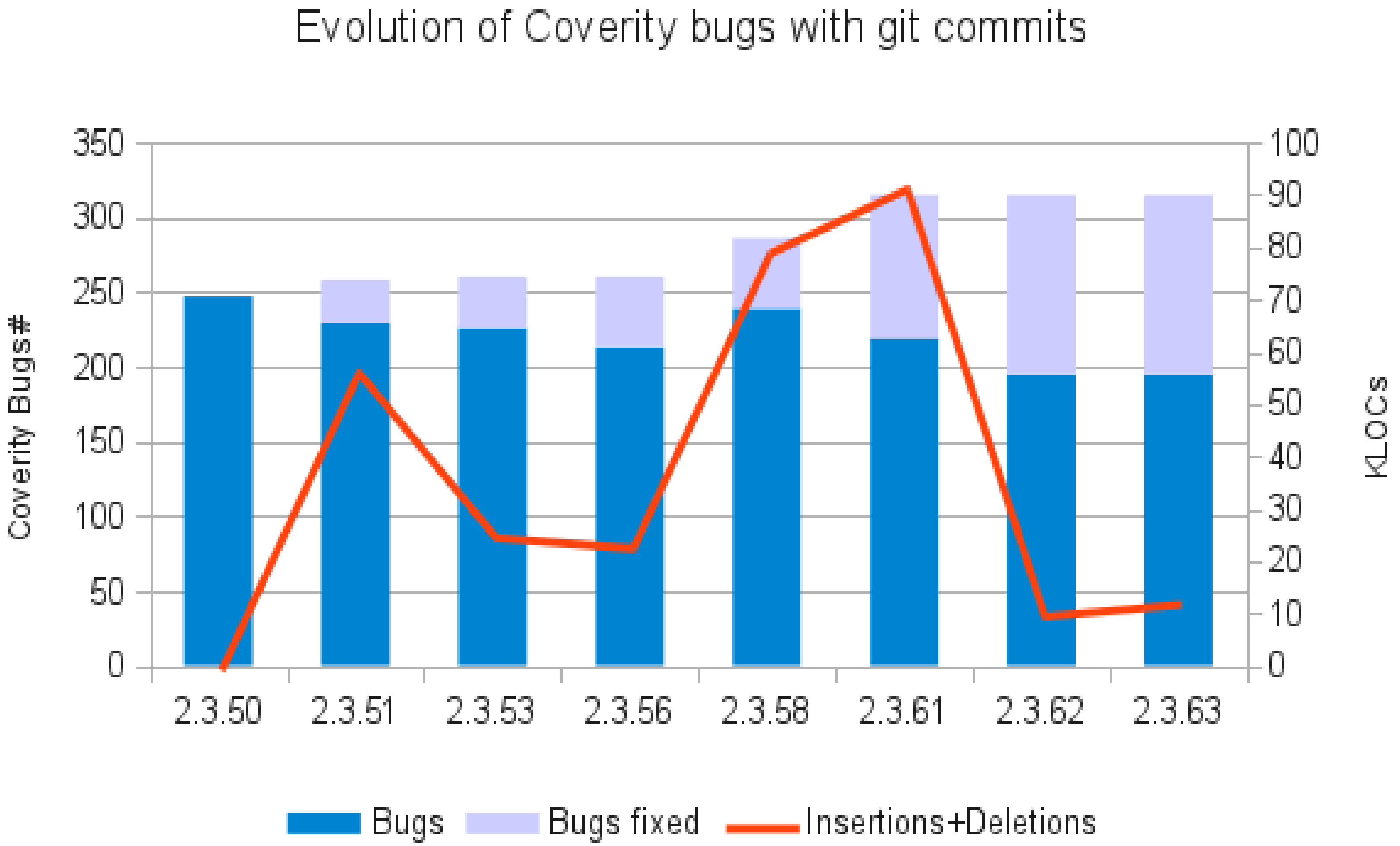
- How can they be avoided?
  - 'fall through' in switch cases: please comment
  - function pointers, like the ones set in cfs_hash_create()
    - Too complex path to follow for Coverity
    - Redesign code?
    - Specific Coverity comments for future analysis

Statistics on master



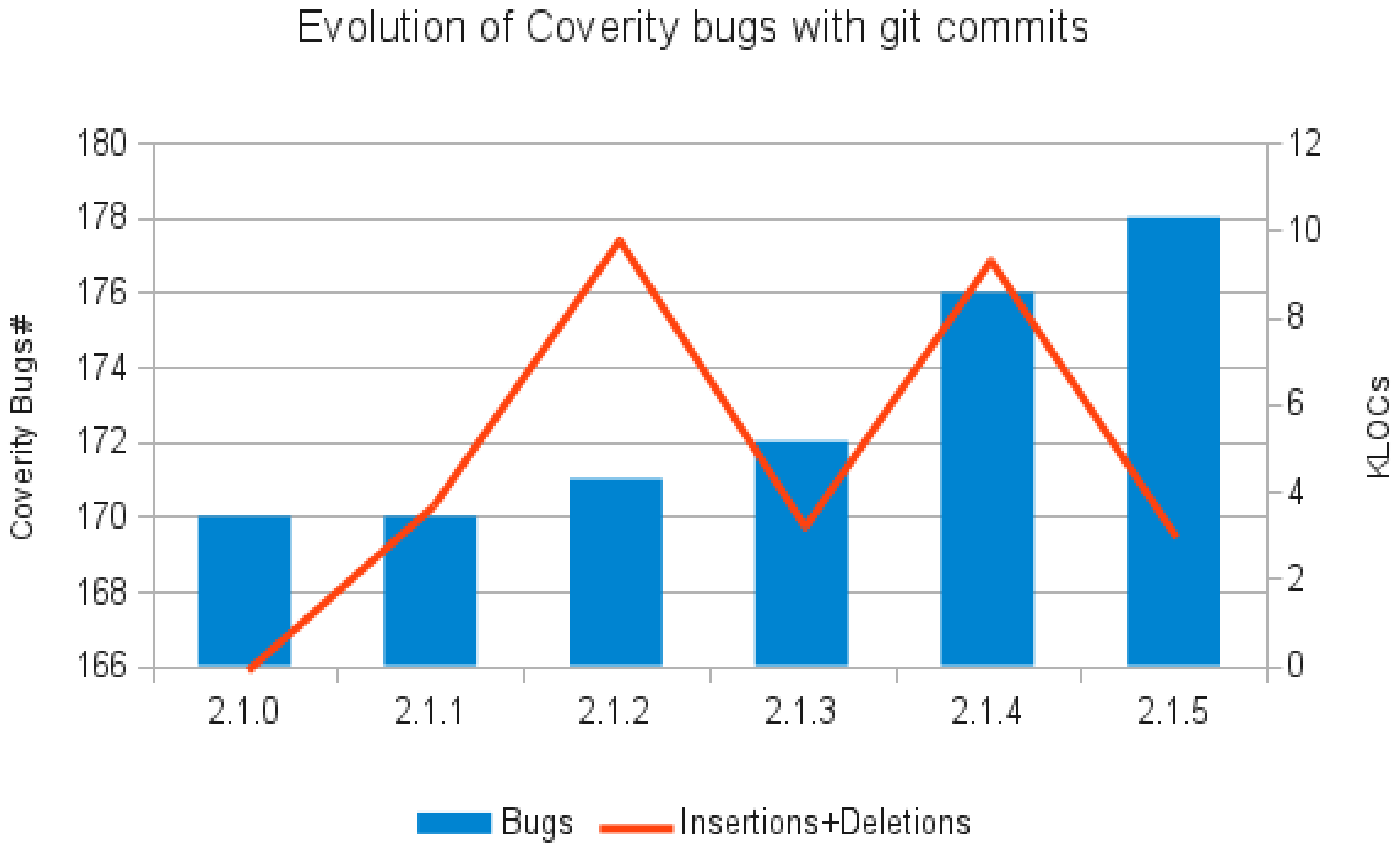
Statistics on master

- Historical view



Statistics on b2_1

- Historical view



Statistics on b2_1

- Removed from 2.1.0 to 2.1.5:
  - "Dereference after null check" in `lustre/obdfilter/filter.c`
    - LU-1042 "1.8 clients show wrong dates with 2.1 servers"
  - "Dereference before null check" in `lustre/mdd/mdd_dir.c`
    - LU-1331 "Allow changelog to extend record"
  - "Logically dead code" in `lustre/obdclass/lustre_peer.c`
    - LU-570 "Add function to find connect uuid by nid"

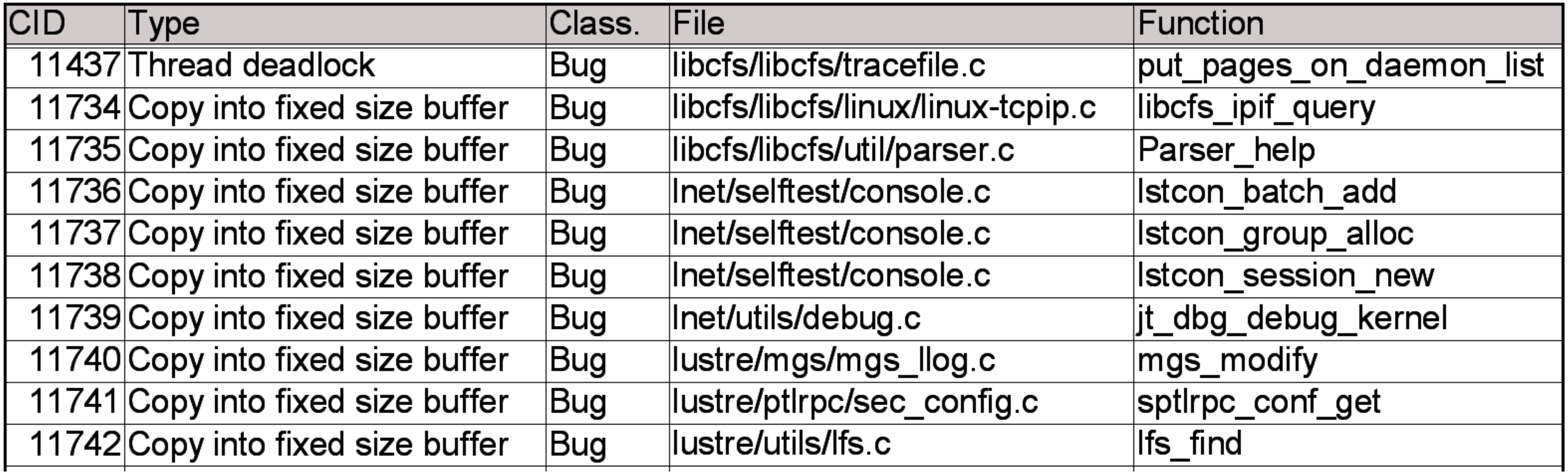
- Introduced between 2.1.0 and 2.1.5:
  - 11 new Coverity bugs

Benefits for the whole Lustre Community

Architect of an Open World™

Access to Coverity defects list

- Export only possible in CSV or XML formats



CID	Type	Class.	File	Function
11437	Thread deadlock	Bug	libcfs/libcfs/tracefile.c	put_pages_on_daemon_list
11734	Copy into fixed size buffer	Bug	libcfs/libcfs/linux/linux-tcpip.c	libcfs_ipif_query
11735	Copy into fixed size buffer	Bug	libcfs/libcfs/util/parser.c	Parser_help
11736	Copy into fixed size buffer	Bug	lnet/selftest/console.c	lstcon_batch_add
11737	Copy into fixed size buffer	Bug	lnet/selftest/console.c	lstcon_group_alloc
11738	Copy into fixed size buffer	Bug	lnet/selftest/console.c	lstcon_session_new
11739	Copy into fixed size buffer	Bug	lnet/utils/debug.c	jt_dbg_debug_kernel
11740	Copy into fixed size buffer	Bug	lustre/mgs/mgs_llog.c	mgs_modify
11741	Copy into fixed size buffer	Bug	lustre/ptlrpc/sec_config.c	sptlrpc_conf_get
11742	Copy into fixed size buffer	Bug	lustre/utils/lfs.c	lfs_find

- No remote access to web GUI
  - License considerations

Benefits for the whole Lustre Community

- Jira tickets opened, patches proposed
  - 30 tickets opened, covering all Coverity bugs
  - Identified with 'coverity' label
  - 13 already merged
  - Thanks for Intel's responsiveness

- Ongoing effort
  - We hope as many patches as possible will be landed before 2.4 GA
  - Our goal is to continue to watch Lustre code with Coverity

Bull

Architect of an Open World™