Devicetree

Overlay

Plumbers 2018
Vancouver, Canada

Frank Rowand, Sony

November 14, 2018
181112_0833

Overlays, one of the gating factors

```
On 10/18/17 14:46, Frank Rowand wrote:

> On Wed, 2017-10-18 at 10:44 -0500, Rob Herring wrote:

>> The issue remains that the kernel is not really setup to deal with any
>> random property or node to be changed at any point in run-time. I
>> think there needs to be some restrictions around what the overlays can
>> touch. We can't have it be wide open and then lock things down later
>> and break users.

> That paragraph is key to any discussion of accepting code to apply overlays.
> Solving that issue has been stated to be a gating factor for such code from
> the beginning of overlay development.
```

(Not the only remaining issue.)

Overlays

https://elinux.org/Frank%27s_Evolving_Overlay_Thoughts

Foundational Linux kernel code is moving in the right direction

Boot loader support is moving forward faster

dtc Compiler

commits from February 10, 2017 to October 2018 in dtc repo include:

- overlay syntactic sugar
    - * eliminate hard coding overlay metadata
- fdtoverlay
    - * standalone tool to apply overlay(s)

U-Boot

- U-Boot overlay support
  - enhancements have been added

Alternative to Linux kernel overlay loader
for some use cases

overlay validation - Linux kernel

Intent: validate kfree() of overlay related memory

- avoid memory leak
- avoid kfree() before overlay changeset remove

Code sprinkled around

- overlay apply
- overlay remove
- overlay kobject free

overlay validation - Linux kernel

Now:

- potential leak: warn, but allow overlay apply
- if error would result in invalid devicetree then fail overlay apply

    example: name collision results in renaming

Future:

- potential leak: consider whether to fail overlay apply or maybe just overlay remove

Intent: if overlay worked before validation, allow time to fix overlay

overlay validation - Linux kernel

Expected in v4.21

- [PATCH v7 00/17] of: overlay: validation checks, subsequent fixes

- https://lore.kernel.org/lkml/1541743565-23163-1-git-send-email-frowand.list@gmail.com/T/#u

Overlay loader remains out of tree, so validation messages will only be visible for unittest, FPGAs, or if you use the out of tree overlay loader

Exposed errors in core devicetree code

- Fixes are in the patch series

unittest - new messages

```
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /testcase-data/overlay-node/test-bus/test-unittest0/status
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /testcase-data/overlay-node/test-bus/test-unittest1/status
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /testcase-data/overlay-node/test-bus/test-unittest2/status
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /testcase-data/overlay-node/test-bus/test-unittest3/status
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /testcase-data/overlay-node/test-bus/test-unittest5/status
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /testcase-data/overlay-node/test-bus/test-unittest6/status
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /testcase-data/overlay-node/test-bus/test-unittest7/status
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /testcase-data/overlay-node/test-bus/test-unittest8/status
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /testcase-data/overlay-node/test-bus/test-unittest8/property-foo
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /testcase-data/overlay-node/test-bus/i2c-test-bus/test-unittest12/status
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /testcase-data/overlay-node/test-bus/i2c-test-bus/test-unittest13/status
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /testcase-data-2/substation@100/status
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /testcase-data-2/fairway-1/status
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /testcase-data-2/fairway-1/ride@100/track@30/incline-up
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /testcase-data-2/fairway-1/ride@100/track@40/incline-up
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /testcase-data-2/lights@40000/status
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /testcase-data-2/lights@40000/color
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /testcase-data-2/lights@40000/rate
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/hvac_2
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/ride_200
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/ride_200_left
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/ride_200_right
OF: overlay: ERROR: multiple fragments add and/or delete node /testcase-data-2/substation@100/motor-1/controller
OF: overlay: ERROR: multiple fragments add, update, and/or delete property /testcase-data-2/substation@100/motor-1/controller/name
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /testcase-data-2/substation@100/motor-1/rpm_avail
OF: overlay: WARNING: memory leak will occur if overlay removed, property: /testcase-data-2/substation@100/motor-1/rpm_avail
OF: overlay: ERROR: multiple fragments add, update, and/or delete property /testcase-data-2/substation@100/motor-1/rpm_avail
```

validation - ERROR, WARNING

[overlay apply]
WARNING: memory leak will occur if overlay removed, property: <prop_path>

cause: property add or modify in node not created by an overlay

[while accessing overlay nodes]
ERROR: memory leak before free overlay changeset, <node_path>

cause: too many of_node_put()

[overlay remove]
ERROR: memory leak, expected refcount 1 instead of <refcount>,
of_node_get()/of_node_put() unbalanced - destroy cset entry: attach overlay node <node_path>

cause: too many of_node_get() or not enough of_node_put()

validation - ERROR

Malformed FDT will not cleanly apply - apply now fails

[overlay apply]

ERROR: changing value of #address-cells is not allowed in <node_path>

ERROR: changing value of #size-cells is not allowed in <node_path>

ERROR: multiple fragments add and/or delete node <node_path>

ERROR: multiple fragments add, update, and/or delete property <prop_path>

ERROR: multiple fragments add and/or delete node <node_path>

```
# drivers/of/unittest-data/overlay_bad_add_dup_node.dts:
```

```
// SPDX-License-Identifier: GPL-2.0
/dts-v1/;
/plugin/;

/*
 * &electric_1/motor-1 and &spin_ctrl_1 are the same node:
 *   /testcase-data-2/substation@100/motor-1
 *
 * Thus the new node "controller" in each fragment will
 * result in an attempt to add the same node twice.
 * This will result in an error and the overlay apply
 * will fail.
 */

&electric_1 {

        motor-1 {
                controller {
                        power_bus = < 0x1 0x2 >;
                };
        };
};

&spin_ctrl_1 {
                controller {
                        power_bus_emergency = < 0x101 0x102 >;
                };
};
```

ERROR: multiple fragments add, update, and/or delete property <prop_path>

```
# drivers/of/unittest-data/overlay_bad_add_dup_prop.dts:
```

```
// SPDX-License-Identifier: GPL-2.0
/dts-v1/;
/plugin/;

/*
 * &electric_1/motor-1 and &spin_ctrl_1 are the same node:
 *   /testcase-data-2/substation@100/motor-1
 *
 * Thus the property "rpm_avail" in each fragment will
 * result in an attempt to update the same property twice.
 * This will result in an error and the overlay apply
 * will fail.
 */

&electric_1 {

        motor-1 {
                rpm_avail = < 100 >;
        };
};

&spin_ctrl_1 {
                rpm_avail = < 100 200 >;
};
```

Linux Internal ERROR

[overlay remove]

ERROR: of_node_release(), unexpected properties in <node_path>

Metadata - see FDT format slides

How should the metadata required by overlays be encoded in the FDT?

Discussion was in progress on devicetree-compiler list

Subject: [RFC] devicetree: new FDT format version
Message-ID: <b96829f9-2e8b-fdc5-5090-58591e2260cf@gmail.com>
Date: Mon, 22 Jan 2018 00:09:18 -0800

side-effect: update of FDT format required

Metadata - see Size slides

Motivation:

- size reduction of FDT and kernel data
- remove metadata from tree name space

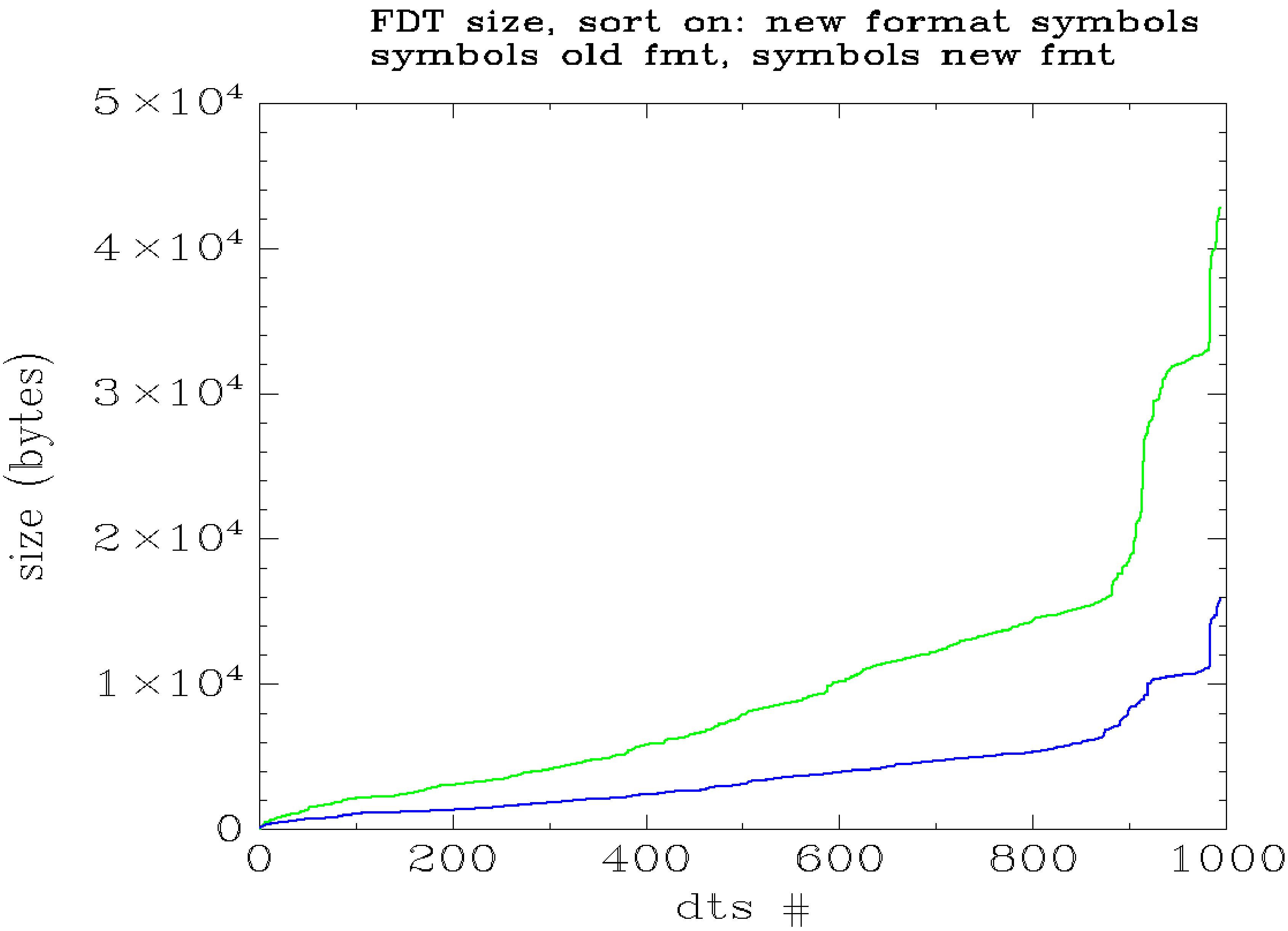
side-effects:

- update of FDT format required
- additional features possible, eg
  * phandle as property value decompile
  * validation features

Metadata - base FDT overhead

Takeaway:

base metadata to enable overlay apply can be large -- this is a concern



FDT size, sort on: new format symbols
symbols old fmt, symbols new fmt

dtc - overlays - Linux v4.15

dtc creates the .dtb **OVERLAY INTERNAL DATA (“metadata”)**

Do not hand code overlay internal data nodes in DTS source:

fragment@

__overlay__

__fixup__

__local_fixup__

__symbols__

dtc - overlays - example - old.dts

```
/dts-v1/;
/plugin/;

/ {
        fragment@0 {
                target-path = "/soc/base_fpga_region";
                #address-cells = <1>;
                #size-cells = <1>;

                __overlay__ {
                        ranges = <0x00000000 0x00000000 0xc0000000 0x00040000>,
                                 <0x00000001 0x00000000 0xff200000 0x00001000>;

                        external-fpga-config;

                        #address-cells = <2>;
                        #size-cells = <1>;

                        fpga_pr_region0 {
                                compatible = "fpga-region";
                                fpga-bridges = <&freeze_controller_0>;
                                ranges;
                        };

                        freeze_controller_0: freeze_controller@100000450 {
                                compatible = "altr,freeze-bridge-controller";
                                reg = <0x00000001 0x00000450 0x00000010>;
                                interrupt-parent = <&intc>;
                                interrupts = <0 21 4>;
                        };
                };
        };
};
```

dtc - overlays - example - new.dts

```
/dts-v1/;
/plugin/;

&fpga_region {
        ranges = <0x00000000 0x00000000 0xc0000000 0x00040000>,
                 <0x00000001 0x00000000 0xff200000 0x00001000>;

        external-fpga-config;

        #address-cells = <2>;
        #size-cells = <1>;

        fpga_pr_region0 {
                compatible = "fpga-region";
                fpga-bridges = <&freeze_controller_0>;
                ranges;
        };

        freeze_controller_0: freeze_controller@100000450 {
                compatible = "altr,freeze-bridge-controller";
                reg = <0x00000001 0x00000450 0x00000010>;
                interrupt-parent = <&intc>;
                interrupts = <0 21 4>;
        };
};
```

dtc - overlays - example

```
$ diff -b -u old.dts new.dts
--- old.dts
+++ new.dts
@@ -1,13 +1,7 @@
 /dts-v1/;
 /plugin/;

-/ {
-        fragment@0 {
-                target-path = "/soc/base_fpga_region";
-                #address-cells = <1>;
-                #size-cells = <1>;
-
-                __overlay__ {
+&fpga_region {
                         ranges = <0x00000000 0x00000000 0xc0000000 0x00040000>,
                                  <0x00000001 0x00000000 0xff200000 0x00001000>;

@@ -28,6 +22,4 @@
                                 interrupt-parent = <&intc>;
                                 interrupts = <0 21 4>;
                         };
-                };
-        };
 };
```

dtc - overlays - new.dts - no label?

```
/dts-v1/;
/plugin/;

&fpga_region {
        ........
};
```

What if there is no label for the overlay target in the base devicetree?

What if the overlay target is the root node (dtc does not allow a label on the root node)?

dtc - overlays - example - new.dts

```
/dts-v1/;
/plugin/;

&{/soc/base_fpga_region} {
        ranges = <0x00000000 0x00000000 0xc0000000 0x00040000>,
                 <0x00000001 0x00000000 0xff200000 0x00001000>;

        external-fpga-config;

        #address-cells = <2>;
        #size-cells = <1>;

        fpga_pr_region0 {
                compatible = "fpga-region";
                fpga-bridges = <&freeze_controller_0>;
                ranges;
        };

        freeze_controller_0: freeze_controller@100000450 {
                compatible = "altr,freeze-bridge-controller";
                reg = <0x00000001 0x00000450 0x00000010>;
                interrupt-parent = <&intc>;
                interrupts = <0 21 4>;
        };
};
```

.dtsi source vs overlay .dtsi

With the new dtc --

Overlay .dts file contains directives:

```
/dts-v1/;
/plugin/;
```

.dtsi include file does not

Use include as .dtsi or overlay

With sugar syntax, the syntax used by an overlay is now compatible with the syntax used by an include file, if the include file uses labels as paths instead of using explicit paths.

- This may be convenient for development workflows

- Do not become dependent on this for overlays that will be long lived -- current thinking is that we want many / most overlays to use the connector model

Use include as .dtsi or overlay

---------- base tree -------------------------------

```
$ expand fpga_tree.dts
/dts-v1/;

/* labels used by overlay are in the base tree */

/ {
        soc {
                intc: interrupt_ctrl {
                };
                fpga_region: base_fpga_region {
                };
        };
};

/include/ "fpga_plugin_or_dtsi.dts"
```

---------- overlay ----------------------------------

```
$ expand fpga_overlay.dts
/dts-v1/;
/plugin/;

/include/ "fpga_plugin_or_dtsi.dts"
```

The .dtsi

```
$ expand fpga_plugin_or_dtsi.dts
&fpga_region {
        ranges = <0x00000000 0x00000000 0xc0000000 0x00040000>,
                 <0x00000001 0x00000000 0xff200000 0x00001000>;

        external-fpga-config;

        #address-cells = <2>;
        #size-cells = <1>;

        fpga_pr_region0 {
                compatible = "fpga-region";
                fpga-bridges = <&freeze_controller_0>;
                ranges;
        };

        freeze_controller_0: freeze_controller@100000450 {
                compatible = "altr,freeze-bridge-controller";
                reg = <0x00000001 0x00000450 0x00000010>;
                interrupt-parent = <&intc>;
                interrupts = <0 21 4>;
        };
};
```

How to get a copy of the slides

1) frank.rowand@sony.com

2) https://elinux.org/Device_Tree_presentations_papers_articles