Devicetree

Linux kernel memory size
FDT size

Plumbers 2018
Vancouver, Canada

Frank Rowand, Sony

November 14, 2018
181113_1529
Made some of the fields optional (based on CONFIG_…) in
- struct property
- struct device_node

CONFIG_DYNAMIC requires most of the fields

Nicolas Pitre reported:
  My test case went from 118072 bytes down to 21548 bytes
  with this series.
dtc - ignore disabled nodes

New node property:
/omit-if-no-ref/

Linux kernel v4.18
commit 50aafd60898a
scripts/dtc: Update to upstream version v1.4.6-21-g84e414b0b5bc
dtc repository
commit 4038fd90056e8
dtc: add ability to make nodes conditional on them being referenced
dtc - tests/omit-no-ref.dts

/dts-v1/;

/{
    test-phandle = <&node3>;
    test-path = &node4;

    /omit-if-no-ref/ node1: node1 {
        bar = <0xdeadbeef>;
    };

    node2: node2 {
        foo = <0x42>;
    };

    node3: node3 {
        test = "test";
    };

    node4: node4 {
        test;
    };
}

/omit-if-no-ref/ &node2;
/omit-if-no-ref/ &node3;
/omit-if-no-ref/ &node4;
Linux decreased memory use

Remove full path from np->full_name
   - add %pOF to generate full path at run-time

Attempt to remove phandle properties
   - https://lore.kernel.org/lkml/87mv92szsw.fsf@concordia.ellerman.id.au/T/#u
   - phandle values are also a field in the node
   - issue with DLPAR systems needs to be resolved before implementing
Linux decreased memory use

Nicolas Pitre reported:

My test case went from 118072 bytes down to 21548 bytes with this series.

After adding dtc skipping disabled nodes and no longer storing the full path of every node, Nicolas reported:

... it is down to 11732 bytes
Linux increased memory use

nodes as kobjects

phandle cache
- reduced overhead of phandle access
- size: $4 \times \text{roundup}\_\text{pow}\_\text{of}\_\text{two}(\# \text{ of phandles})$
Linux memory opportunities

Option to not load overlay metadata from FDT
- If bootloader has applied overlay(s) and no more overlays will be applied

Place FDT overlay metadata somewhere other than nodes, and in a more compact format
- may be able to access in-place in FDT image
- more compact format is a win

Option to not load inactive nodes from FDT
FDT size

Overlay Metadata format and encoding

Motivation:
- size reduction of FDT and kernel data
- remove metadata from tree name space
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 - base FDT overhead

Metadata overhead measured for arch/arm/boot/dts/*

“symbols old fmt” is added size from 'dtc -@'
  for the current FDT format

“symbols new fmt” is added size from 'dtc -@'
  for first proposed format in the email thread
## Metadata - base FDT overhead

Metadata overhead measured for arch/arm/boot/dts/*

<table>
<thead>
<tr>
<th>row</th>
<th>dtb no symbols</th>
<th>delta symbols</th>
<th>delta new fmt</th>
<th>bytes saved</th>
</tr>
</thead>
<tbody>
<tr>
<td>99%</td>
<td>90531</td>
<td>42721</td>
<td>15766</td>
<td>26955</td>
</tr>
<tr>
<td>83%</td>
<td>44302</td>
<td>14582</td>
<td>5163</td>
<td>9419</td>
</tr>
<tr>
<td>66%</td>
<td>26277</td>
<td>11662</td>
<td>4628</td>
<td>7034</td>
</tr>
<tr>
<td>49%</td>
<td>21047</td>
<td>7328</td>
<td>2754</td>
<td>4574</td>
</tr>
<tr>
<td>33%</td>
<td>12864</td>
<td>4305</td>
<td>1705</td>
<td>2600</td>
</tr>
<tr>
<td>16%</td>
<td>12009</td>
<td>2929</td>
<td>1520</td>
<td>1409</td>
</tr>
<tr>
<td>0%</td>
<td>1220</td>
<td>68</td>
<td>149</td>
<td>-81</td>
</tr>
</tbody>
</table>

- “delta symbols” is added size from `dtc -@`
- “new fmt” is added size from `dtc -@` for first proposed in the email thread
FDT size, sort on: new format symbols
symbols old fmt, symbols new fmt
FDT size, sort on: saved
old fmt, new fmt, no symbols, saved
FDT size

Thought for the future:

Tool to strip overlay metadata from FDT

Maybe “easy” to implement with proposed new FDT format.

I discourage implementing with current FDT format (more legacy to obsolete)