Item 171
git_comments:

1. pylint: disable=assignment-from-no-return
2. Positional relay function to create conv2d NCHWc operator used by frontend FFI.

git_commits:

1. **summary:** [RELAY][TOPI] `alter_op_layout` for x86 (#2602)
   message: [RELAY][TOPI] `alter_op_layout` for x86 (#2602) * alter_op_layout for x86 * cleanup * cleanup * fix lint * fix lint * fix lint * fix lint * change support level * change other support levels
   label: code-design

github_issues:

1. **title:** [Relay] x86 conv2d_alter_layout support
   body: Got the warning message "Only support alter layout for x86 in NNVM now. This pass is ignored in relay" when running codes in from_tensorflow tutorial. https://github.com/dmlc/tvm/blob/master/topi/python/topi/x86/conv2d.py#L287-L291
2. **title:** [Relay] x86 conv2d_alter_layout support
   body: Got the warning message "Only support alter layout for x86 in NNVM now. This pass is ignored in relay" when running codes in from_tensorflow tutorial. https://github.com/dmlc/tvm/blob/master/topi/python/topi/x86/conv2d.py#L287-L291
3. **title:** [Relay] x86 conv2d_alter_layout support
   body: Got the warning message "Only support alter layout for x86 in NNVM now. This pass is ignored in relay" when running codes in from_tensorflow tutorial. https://github.com/dmlc/tvm/blob/master/topi/python/topi/x86/conv2d.py#L287-L291
4. **title:** [Relay] x86 conv2d_alter_layout support
   body: Got the warning message "Only support alter layout for x86 in NNVM now. This pass is ignored in relay" when running codes in from_tensorflow tutorial. https://github.com/dmlc/tvm/blob/master/topi/python/topi/x86/conv2d.py#L287-L291
5. **title:** [Relay] x86 conv2d_alter_layout support
   body: Got the warning message "Only support alter layout for x86 in NNVM now. This pass is ignored in relay" when running codes in from_tensorflow tutorial. https://github.com/dmlc/tvm/blob/master/topi/python/topi/x86/conv2d.py#L287-L291

github_issues_comments:

1. Would you mind shoot a PR and test for this? I believe the functionality is all prepared.
2. @yzhliu Sure, I am working on another pr, will work on this fix after that, hopefully in this week.
3. Actually I need this support for some experiments with AutoTVM, so I opened a PR for this #2602. (It was a little more involved than I anticipated, because TOPI still separates x86 NCHWc from vanilla conv2d)

github_pulls:

1. **title:** [RELAY][TOPI] `alter_op_layout` for x86
   body: Adds support for `alter_op_layout` for x86 that was discussed in #2585. Additional comments: Currently we have an issue where the "vanilla" conv2d operator can be instantiated with some "NCHWc" style layouts, such as "NCHW4c," but not others (e.g., "NCHW8c" or "NCHW16c"). This seems to be due to an issue where the latter two are treated as a special case and handled by the x86 operator. This version seems to share basically all of the same properties as vanilla conv2d, but is not merged together with vanilla conv2d, and other parts of the stack (e.g., AutoTVM) treat these variants of conv2d as separate. It seems theoretically possible to merge the two, but that would likely come at the cost of breaking/changing a few other things (e.g., AutoTVM) and currently TOPI (shared between NNVM and Relay) treats them as separate. This PR mostly preserves the current "organization," but it reuses the Conv2DAttrs node for Conv2DNCHWc, as it seems very excessive to create another identical definition. However, please let me know if reusing the attribute node type is not kosher.
2. **title:** [RELAY][TOPI] `alter_op_layout` for x86

body: Adds support for `alter_op_layout` for x86 that was discussed in #2585. Additional comments: Currently we have an issue where the "vanilla" conv2d operator can be instantiated with some "NCHWc" style layouts, such as "NCHW4c," but not others (e.g., "NCHW8c" or "NCHW16c"). This seems to be due to an issue where the latter two are treated as a special case and handled by the x86 operator. This version seems to share basically all of the same properties as vanilla conv2d, but is not merged together with vanilla conv2d, and other parts of the stack (e.g., AutoTVM) treat these variants of conv2d as separate. It seems theoretically possible to merge the two, but that would likely come at the cost of breaking/changing a few other things (e.g., AutoTVM) and currently TOPI (shared between NNVM and Relay) treats them as separate. This PR mostly preserves the current "organization," but it reuses the Conv2DAttrs node for Conv2DNCHWc, as it seems very excessive to create another identical definition. However, please let me know if reusing the attribute node type is not kosher.

3. **title:** [RELAY][TOPI] `alter_op_layout` for x86
   body: Adds support for `alter_op_layout` for x86 that was discussed in #2585. Additional comments: Currently we have an issue where the "vanilla" conv2d operator can be instantiated with some "NCHWc" style layouts, such as "NCHW4c," but not others (e.g., "NCHW8c" or "NCHW16c"). This seems to be due to an issue where the latter two are treated as a special case and handled by the x86 operator. This version seems to share basically all of the same properties as vanilla conv2d, but is not merged together with vanilla conv2d, and other parts of the stack (e.g., AutoTVM) treat these variants of conv2d as separate. It seems theoretically possible to merge the two, but that would likely come at the cost of breaking/changing a few other things (e.g., AutoTVM) and currently TOPI (shared between NNVM and Relay) treats them as separate. This PR mostly preserves the current "organization," but it reuses the Conv2DAttrs node for Conv2DNCHWc, as it seems very excessive to create another identical definition. However, please let me know if reusing the attribute node type is not kosher.
   label: code-design
4. **title:** [RELAY][TOPI] `alter_op_layout` for x86
   body: Adds support for `alter_op_layout` for x86 that was discussed in #2585. Additional comments: Currently we have an issue where the "vanilla" conv2d operator can be instantiated with some "NCHWc" style layouts, such as "NCHW4c," but not others (e.g., "NCHW8c" or "NCHW16c"). This seems to be due to an issue where the latter two are treated as a special case and handled by the x86 operator. This version seems to share basically all of the same properties as vanilla conv2d, but is not merged together with vanilla conv2d, and other parts of the stack (e.g., AutoTVM) treat these variants of conv2d as separate. It seems theoretically possible to merge the two, but that would likely come at the cost of breaking/changing a few other things (e.g., AutoTVM) and currently TOPI (shared between NNVM and Relay) treats them as separate. This PR mostly preserves the current "organization," but it reuses the Conv2DAttrs node for Conv2DNCHWc, as it seems very excessive to create another identical definition. However, please let me know if reusing the attribute node type is not kosher.
5. **title:** [RELAY][TOPI] `alter_op_layout` for x86
   body: Adds support for `alter_op_layout` for x86 that was discussed in #2585. Additional comments: Currently we have an issue where the "vanilla" conv2d operator can be instantiated with some "NCHWc" style layouts, such as "NCHW4c," but not others (e.g., "NCHW8c" or "NCHW16c"). This seems to be due to an issue where the latter two are treated as a special case and handled by the x86 operator. This version seems to share basically all of the same properties as vanilla conv2d, but is not merged together with vanilla conv2d, and other parts of the stack (e.g., AutoTVM) treat these variants of conv2d as separate. It seems theoretically possible to merge the two, but that would likely come at the cost of breaking/changing a few other things (e.g., AutoTVM) and currently TOPI (shared between NNVM and Relay) treats them as separate. This PR mostly preserves the current "organization," but it reuses the Conv2DAttrs node for Conv2DNCHWc, as it seems very excessive to create another identical definition. However, please let me know if reusing the attribute node type is not kosher.
6. **title:** [RELAY][TOPI] `alter_op_layout` for x86
   body: Adds support for `alter_op_layout` for x86 that was discussed in #2585. Additional comments: Currently we have an issue where the "vanilla" conv2d operator can be instantiated with some "NCHWc" style layouts, such as "NCHW4c," but not others (e.g., "NCHW8c" or "NCHW16c"). This seems to be due to an issue where the latter two are treated as a special case and handled by the x86 operator. This version seems to share basically all of the same properties as vanilla conv2d, but is not merged together with vanilla conv2d, and other parts of the stack (e.g., AutoTVM) treat these variants of conv2d as separate. It seems theoretically possible to merge the two, but that would likely come at the cost of breaking/changing a few other things (e.g., AutoTVM) and currently TOPI (shared between NNVM and Relay) treats them as separate. This PR mostly preserves the current "organization," but it reuses the

Conv2DAttrs node for Conv2DNCHWc, as it seems very excessive to create another identical definition. However, please let me know if reusing the attribute node type is not kosher.

github_pulls_comments:

1. @eqy Could you also help to change the other contrib oplevels in this file?
2. Thanks @eqy @tqchen This is now merged

github_pulls_reviews:

1. maybe use level 5 instead for contrib op?
2. contrib level is 10

jira_issues:

jira_issues_comments: