Hardware Platform Requirements

See also: Hardware Platform Enablement In ONAP

Can we use the TOSCA Specs Normatives with the HPA requirements?

Overview of the existing TOSCA Specs Normatives

```
capability_types:

  tosca.capabilities.Compute:
    derived_from: tosca.capabilities.Root
    properties:
      name:
        type: string
        required: false
      num_cpus:
        type: integer
        required: false
        constraints:
          - greater_or_equal: 1
      cpu_frequency:
        type: scalar-unit.frequency
        required: false
        constraints:
          - greater_or_equal: 0.1 GHz
      disk_size:
        type: scalar-unit.size
        required: false
        constraints:
          - greater_or_equal: 0 MB
      mem_size:
        type: scalar-unit.size
        required: false
        constraints:
          - greater_or_equal: 0 MB

  tosca.capabilities.Container:
    derived_from: tosca.capabilities.Compute

  tosca.capabilities.Storage:
    derived_from: tosca.capabilities.Root
    properties:
      name:
        type: string
        required: false




tosca.relationships.HostedOn:
  derived_from: tosca.relationships.Root
  valid_target_types: [ tosca.capabilities.Container ]
# tosca.relationships.HostedOn:
# - impacts declarative workflow
# - based on tosca.capabilities.Container
```

Critics of TOSCA Spec Normatives:

- tosca.capabilities.Container inherits from Compute!!!

- tosca.capabilities.Compute, property 'name' - we don't actually need it?
- tosca.capabilities.Storage is not similar to the tasca Compute - it only includes the 'name' property, does nor specify any storage quantifiers like size etc.
- in TOSCA Specs, Storage is not an infrastructure-level requirement; it is rather a kind of application-level (attachable, with an initial image), required by (attached to) other application nodes
- Tosca Compute capability is a mix of CPU+Mem. Don't we want to keep them apart for greater flexibility?
- TOSCA Specs does not have requirements for I/O
- TOSCA Specs normatives do not provide all information items required by the ETSI and IM
- tosca.capabilities.Compute and *.Storage bring with them unnecessary inheritance, coupling with relationship types, may affect the TOSCA declarative workflows

Proposed solution

1. Model the basic HPA requirements with TOSCA capability types. VDUs will have requirements of these capability types. These HPA requirements will never be satisfied within the VNF and Service models, the orchestrator will do that in run-time
2. Abstain from using tosca.capability.Compute as it does not fully match the requirements
3. Define special ONAP capability types, separate for each of these categories: CPU, Memory, Storage, I/O, networking. These capability types express the most basic hardware characteristics. They also should be simple, flat bags of strictly named properties, all properties of primitive data types with clear restrictions.
4. Derive from the basic capability types an additional level of capabilities, with advanced (HPA) details. As detailed as they seem, these capabilities are still generic, with strict definitions of properties that are shared by the major hardware vendors. In addition to the strictly-typed properties, the HPA-level capabilities will also have a json-formatted property in order to allow for even greater customization flexibility.
5. Allow for further customization of the HPA-level capabilities into vendor-specific capabilities. These vendor-specific customized capability types may extend their HPA-level generic base by adding new properties and providing new constraints for the existing properties. In addition to the refinement of the strictly defined properties, the vendor-specific capabilities may provide their own validation schema for the json-formatted "flexible" property.

Basic requirements

```
####### Basic specifications of hadware capabilities ####
capability_types:
  onap.capabilities.infrastructure.CPU:
    derived_from: tosca.capabilities.Root
    description: basic processor capabilities
    properties:
      num_cpus:
        type: integer
        required: false
        constraints:
          - greater_or_equal: 1
      cpu_frequency:
        type: scalar-unit.frequency
        required: false
        constraints:
          - greater_or_equal: 0.1 GHz

  onap.capabilities.infrastructure.Memory:
    derived_from: tosca.capabilities.Root
    description: basic memory capabilities
    properties:
      mem_size:
        type: scalar-unit.size
        required: false
        constraints:
          - greater_or_equal: 0 MB

  onap.capabilities.infrastructure.Storage:
    derived_from: tosca.capabilities.Root
    description: basic storage specifications
    properties:
      storage_size:
        type: scalar-unit.size
        required: false
        constraints:
          - greater_or_equal: 0 MB

  onap.capabilities.infrastructure.IO:
    derived_from: tosca.capabilities.Root
    description: basic IO specifications

  onap.capabilities.infrastructure.NIC:
    derived_from: tosca.capabilities.Root
    description: basic networking interface characteristics
```

Advanced HPA specifications

```
####### Advanced hardware capabilities, with more details ####
capability_types:
  onap.capabilities.infrastructure.hpa.CPU:
    derived_from: onap.capabilities.infrastructure.CPU
    description: detailed processor capabilities for hardware-aware VNF vendors
    properties:
      schema_selector:
        description: vendor+architecture, for example, Intel64
        type: string
        required: true
      schema_version:
        type: version
        required: false
      custom_features:
        description: additional features, formatted as JSON, validated against a schema
        type: json
        constraints:
          - schema: http://schema.url
        required: false

      simultaneousMultiThreading:
        type: boolean
        description: |
          The use of Simultaneous Multi-Threading HW is an efficient way to
          increase the compute capacity of a platform. SMT HW threads share
          some CPU core resources. In some VDU implementations, it may be
          necessary to very explicitly control the HW thread allocation on
          a platform. This could be to help ensure locality in data caches
          or as a mechanism to enhance determinism
        required: false
      logicalCpuPinningPolicy:
        type: string
        constraints:
          - valid_values: [Dedicated, Shared]
        required: false
      logicalCpuThreadPinningPolicy:
        type: string
        constraints:
          - valid_values:
              - Isolate   # Allocate on different execution units.
              - Prefer    # co-location of vCPUs to physical execution units
              - Require   # co-location of vCPUs to physical execution units
        required: false
      instructionSetExtensions:
        type: string
        constraints:
          - valid_values:
              - Isolate   # Allocate on different execution units.
              - Prefer    # co-location of vCPUs to physical execution units
              - Require   # co-location of vCPUs to physical execution units
        required: false
      hypervisorConfiguration:
        type: string
        required: false
      computeRas:
        description: Reliability, Availability, Serviceability (RAS)
        type: string
        required: false

  onap.capabilities.infrastructure.hpa.Memory:
    derived_from: onap.capabilities.infrastructure.Memory
    description: HPA-level memory capabilities
    properties:
      schema_selector:
        description: vendor+architecture, for example, Intel64
        type: string
        required: true
      schema_version:
        type: version
        required: false
      custom_features:
        description: additional features, formatted as JSON, validated against a schema
        type: json
        required: false

      memoryPageSize:
        type: scalar-unit.size
        required: false
      memoryAllocationPolicy:
        type: string
        constraints:
          - valid_values:
              - StrictLocal
              - PreferredLocal
        required: false
      memoryBandwidth:
        description: Agreed unit of memory bandwidth
        type: scalar-unit.size
        required: false
      processorCacheAllocation:
        description: Agreed unit of processor cache
        type: string
        required: false
      memoryType:
        description: Type of memory
        type: string
        required: false
      memorySpeed:
        description: Agreed unit of memory speed
        type: string
        required: false
      memoryRas:
        description:
        type: string
        required: false
      localNumaMemory:
        type: boolean
        required: false

  onap.capabilities.infrastructure.hpa.Storage:
    derived_from: onap.capabilities.infrastructure.Storage
    description: HPA-level storage specifications
    properties:
      schema_selector:
        description: vendor+architecture, for example, Intel64
        type: string
        required: true
      schema_version:
        type: version
        required: false
      custom_features:
        description: additional features, formatted as JSON, validated against a schema
        type: json
        required: false

      storageIops:
        type: integer
        required: false
        constraints:
          - greater_or_equal: 0
      storageResilencyMechanism:
        type: string
        required: false
        description: Erasure code based back-end, triple replication

  onap.capabilities.infrastructure.hpa.IO:
    derived_from: onap.capabilities.infrastructure.IO
    description: HPA-level IO characteristics
    properties:
      schema_selector:
        description: vendor+architecture, for example, Intel64
        type: string
        required: true
      schema_version:
        type: version
        required: false
      custom_features:
        description: additional features, formatted as JSON, validated against a schema
        type: json
        required: false

      pciVendorId:
        description: PCI-SIG vendor ID for the device
        type: string
        required: false
      pciDeviceId:
        description: PCI-SIG device ID for the device
        type: string
        required: false
      pciNumDevices:
        description:         Number of PCI devices required.
        type: string
        required: false
      pciAddress:
        description: Geographic location of the PCI device via the standard PCI-SIG addressing model of Domain:
Bus:device:function
        type: string
        required: false
      pciDeviceLocalToNumaNode        Boolean
        type: string
        required: false

  onap.capabilities.infrastructure.hpa.NIC:
    derived_from: onap.capabilities.infrastructure.NIC
    description: HPA-level networking interface characteristics
    properties:
      schema_selector:
        description: vendor+architecture, for example, Intel64
        type: string
        required: true
      schema_version:
        type: version
        required: false
      custom_features:
        description: additional features, formatted as JSON, validated against a schema
        type: json
        required: false

      nicFeature:
        description: Long list of NIC related items such as LSO, LRO, RSS, RDMA, etc.
        type: map
        entry_schema: string
        required: false
      dataProcessingAccelerationLibray:
        description: Name and version of the data processing acceleration library required. Orchestration can
match any NIC that is known to be compatible with the specified library.
        type: string
        required: false
      interfaceType:
        description: Virtio, PCI-Passthrough, SR-IOV, E1000, RTL8139, PCNET, etc.
        type: string
        required: false
```

Examples of a vendor-specific refinment of an HPA capability

```
capability_types:
  com.intel.capabilities.hpa.CPU:
    derived_from: onap.capabilities.infrastructure.hpa.CPU
      schema_selector:
        type: string
        required: true
        constraints:  # fixed value for this vendor
          - equal_to: Intel64
      schema_version:
        type: version
        required: false
        constraints:
          - equal_to: 2.0
      instructionSetExtensions:
        type: string
        constraints:   # changes property definitions
          - valid_values: [aes, sse, avx, cat, cmt, mbm]
        required: false
      a_well_known_property_of_this_vendor: # adds new strictly typed properties
        type: integer
        required: false
```

VDU type - requires hardware

```
node_types:
  onap.nodes.VDU:
    derived_from: onap.nodes.Resource
    capabilities:
      host:
        type: onap.capabilities.Container
    requirements:
      - cpu:
          capability: onap.capabilities.infrastructure.CPU:
          occurrences: [0..UNBOUNDED]
      - memory:
          capability: onap.capabilities.infrastructure.Memory:
          occurrences: [0..UNBOUNDED]
      - storage:
          capability: onap.capabilities.infrastructure.Storage:
          occurrences: [0..UNBOUNDED]
      - io:
          capability: onap.capabilities.infrastructure.IO:
          occurrences: [0..UNBOUNDED]
      - nic:
          capability: onap.capabilities.infrastructure.NIC:
          occurrences: [0..UNBOUNDED]
```

Example of VDU node

```
node_templates:
  vdu_123:
        type: onap.nodes.VDU
    capabilities:
      host:
    requirements:
      - memory:
          node_filter:
            capabilities:
              - onap.capabilities.infrastructure.Memory:
                  properties:
                    - mem_size: {greater_or_equal: 2MB}

      - cpu:
          node_filter:
            capabilities:
              - onap.capabilities.infrastructure.hpa.CPU:
                  properties:
                    - schema_selector:
                        constraints:  # fixed value for this vendor
                          - equal_to: Intel64
                    - schema_version:
                        constraints:
                          - greater_or_equal: 2.0
                    - custom_features:
                        constraints:
                          - equals_to: {,,.sigjigigirgigmirig}
                          - schema: http://json.schema.url
```

Implications

- Compliance with TOSCA 1.2 for using JSON properties
- Support for the 'node_filter' construct

- New languages to understand: JSON Schema, XML Schema, etc