

Dynamic Resources Terminology

The intent of this document: Define the terminology around Dynamic Resources

Terminology:

- **Dynamic Resource Management:**
 - **Goal:** Optimize the usage of *HPC resources* according to given optimization objective(s).
 - **Method:** Dynamically change the association between *HPC execution concepts* and *HPC resources*
- **HPC Execution Concepts:** HPC execution concepts are concepts that describe the execution of computational work using *HPC resources*.
 - **Process:** A process is a fine-grained execution concept. Processes execute instances of a single executable with its own address space and execution context and possibly multiple execution threads. Processes are executed on *HPC resources* (e.g. a CPU core) and have access to other *HPC resources* (e.g. to multiple CPU cores using threads or to GPUs using offloading).
 - **Application:** In this work, we assume an application instance to be a single executable executed in parallel by one or more processes. Applications have access to the union of the resources associated with the processes executing the single instance of the application. Applications and application libraries usually are expressed using parallel programming models, such as MPI, to exploit and manage the parallelism present when used with multiple processes. Note, that these application processes could provide a runtime environment for higher-level programming models such as task-based approaches.
 - **Job:** A job consists of one or more application instances executed as a single invocation. It has access to the union of the resources accessed by the application instances it consists of. The applications composed in a single job often share a common communication context, e.g. a global MPI communicator.
 - **Workflow:** A workflow consists of one or more interdependent jobs running simultaneously and/or sequentially. The jobs in a workflow typically use a separate communication context.
 - **Process Set (PSet):** A PSet is a set of processes where both, the set and its members are uniquely identifiable.

- **HPC Resources:** HPC resources are the essential components and capabilities that contribute to the execution of *execution concepts* on HPC systems.
 - **Compute Resources:** Compute resources are used by execution concepts to carry out computations. This includes CPU cores, GPU execution engines, cores on SmartNICs, as well as processing elements on other accelerators.
 - **Storage Resources:** Storage resources are used by execution concepts to store and load data for the computations. This includes, for instance, memory (typically RAM), persistent storage, and more structured storage like databases.
 - **Indirect Resources:** Indirect resources are resources that are not directly used by the execution concepts, but are associated with compute and storage resources or system services. For instance, network resources, including node-local and internode networks, connect different compute and/or storage resources. Energy resources, such as energy budget, and cooling systems are associated with the operation of compute and storage resources.

- **HPC Resource Management Software (RMS):** The task of HPC RMS is the assignment of HPC resources to execution concepts to facilitate their efficient execution. According to the different types of HPC resources and execution concepts, the term HPC RMS can refer to a wide range of different software components embedded in the HPC software stack.
 - **System-level Resource Manager:** The system-level resource manager manages the submission and scheduling of workflows and assigns resources to workflows through resource allocations. It is usually configured directly by system administrators to align with specific optimization objectives and policies desired by the hosting entity.
 - **Process Manager:** A process manager is usually running inside of an allocation provided by the system-level resource manager and is responsible for the job execution. It manages the lifecycle of processes, including launch, termination, and failure handling. The process manager also often provides a runtime environment for the employed parallel programming models, such as MPI or task-based programming models.

- **Application-specific (resource management) libraries:** Many HPC applications use libraries for the management of application resources and load-balancing, which are specialized for the particular type of application (e.g. adaptive mesh refinement or task execution frameworks). These libraries often serve as a shim between the application and some parallel programming models to ease the complexity of resource management for application developers in this particular domain.

- Workload != Workflow
- Tasks
- Step
- Parallel Runtime
- Malleability != Elasticity
 - Active
 - Passive
- Dynamic Environment: give and remove resources dynamically to/from somebody or something

- Why should we care about dynamic environments?
 - User perspective
 - Site perspective
- What would a dynamic environment consist of?
 - Resource Manager
 - Parallel Runtime
 - Applications
 - Storage
 - Policies (QoS)
 - Monitoring
 - API