Workload Runtime And Placement (WRAP)

NGRM High-Level Review

March 4 2013

Dong H. Ahn

Lawrence Livermore National Laboratory

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

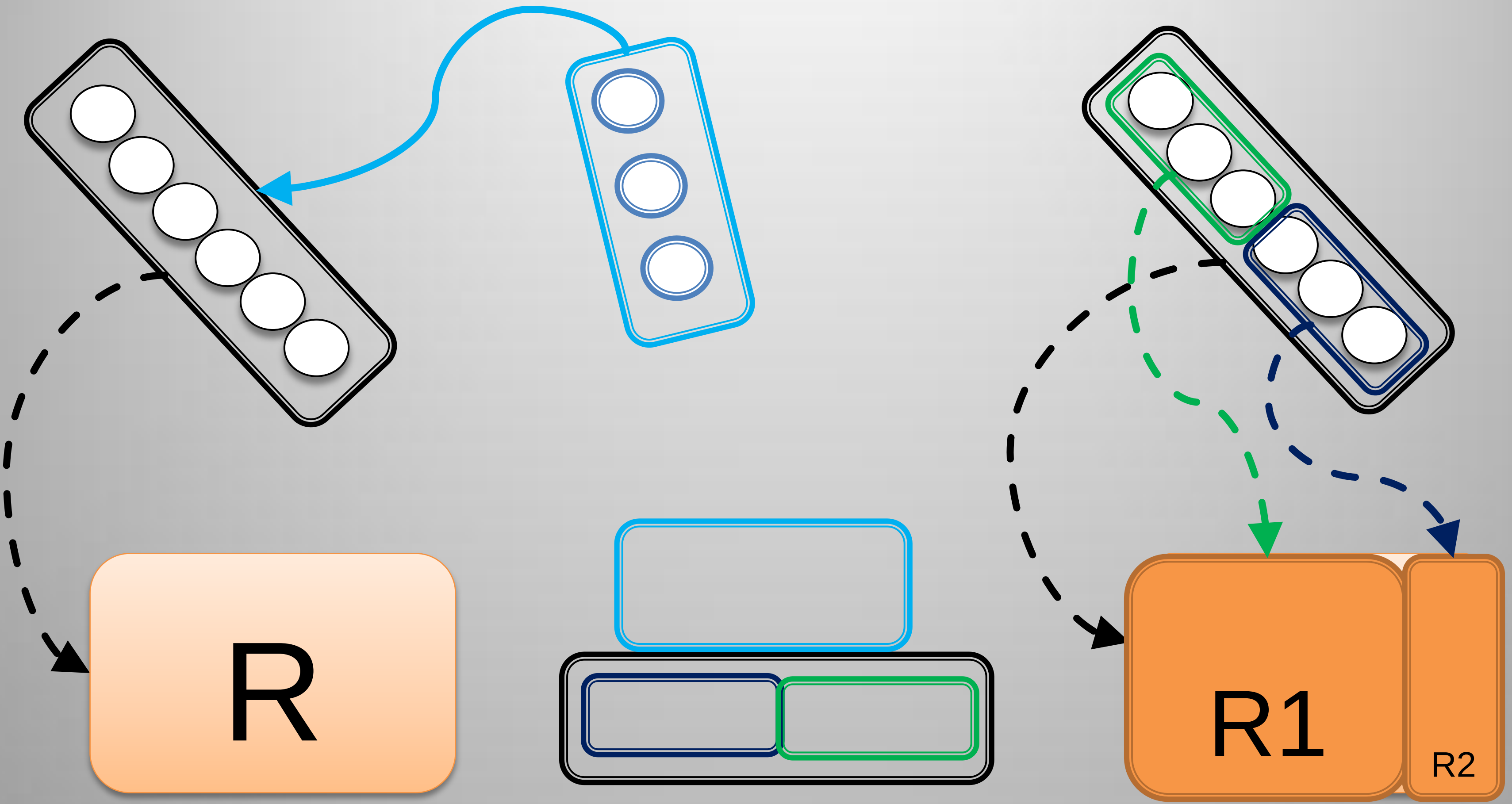
WRAP concerns all aspects of executing transactions of a job in the new paradigm.

- Scheduler sets the overall bound for resources and the duration for a job, and now what?
- Opening the job will let loose various transactions on this bound to execute.
- WRAP must provide run-time services to execute them most effectively and efficiently under the new RM paradigm.

The new paradigm needs new ways to organize/group processes a job executes.

- The traditional approach models transactions as a set of compute steps (e.g., job steps).
  - Limits the ability to seamlessly integrate various transactions beyond compute transactions (e.g., tools).
  - Limits the ability to relate one group of processes to another.
  - Limits the ability to enable resource elasticity.
- Designing the WRAP services after this traditional model would be an under-design.

LWJ is our model to group processes executed by a job in many meaningful way.

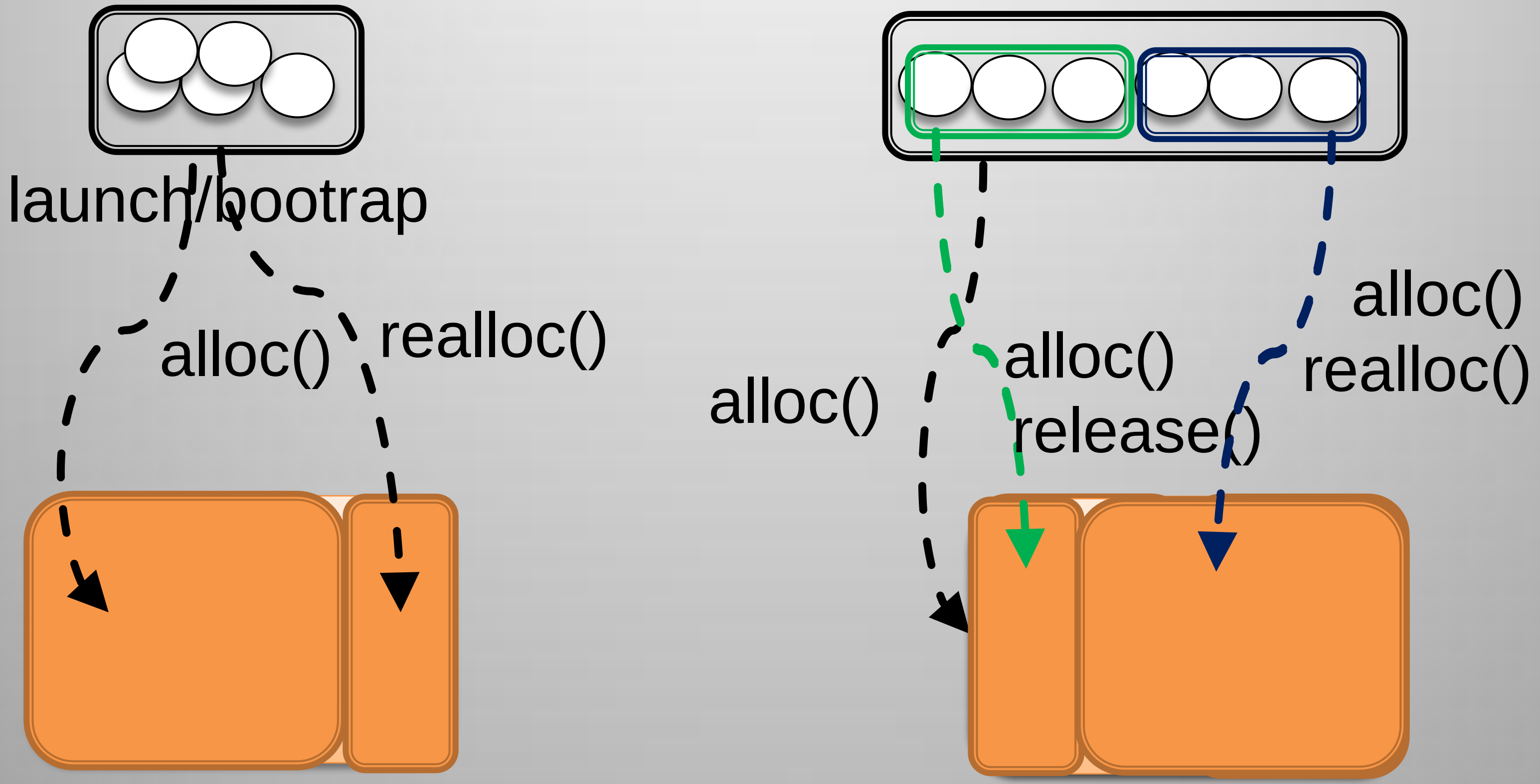


WRAP uses LWJs to express run-time services concisely.

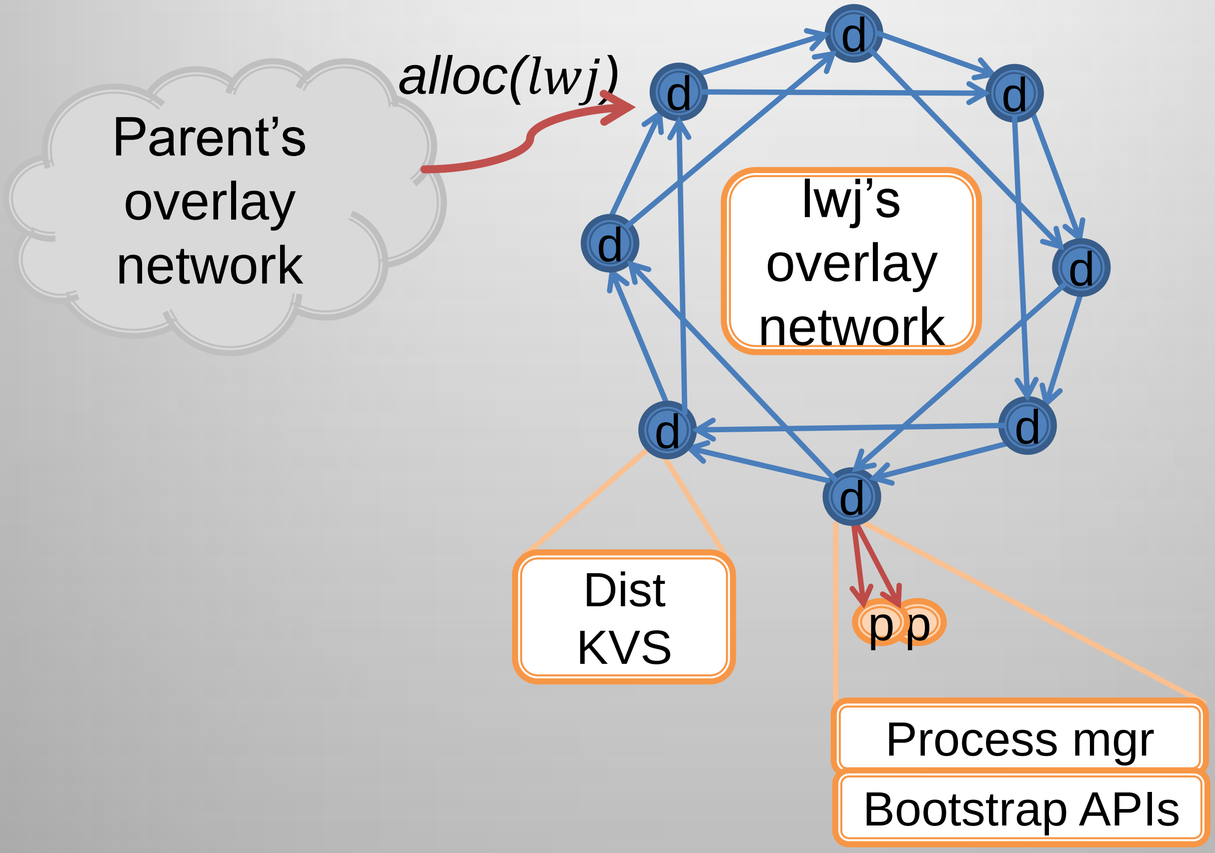
- LWJ serves as group identifiers for WRAP services to relate each LWJ to diverse resources and other LWJs.
- Resource allocation and elasticity: ***alloc(lwj, c)***, ***realloc(lwj,c)***, and ***release(lwj, r)***.
- Process management/confinement: ***launch(lwj)***, ***destroy(ljw)***, ***etc.***
- Synchronization: ***sync(lwj(i), lwj(k))***.
- Resource discovery and provenance: ***query(lwj)*** and ***record(lwj)***.

We can compose the primitives to enable high-level services of the new paradigm.

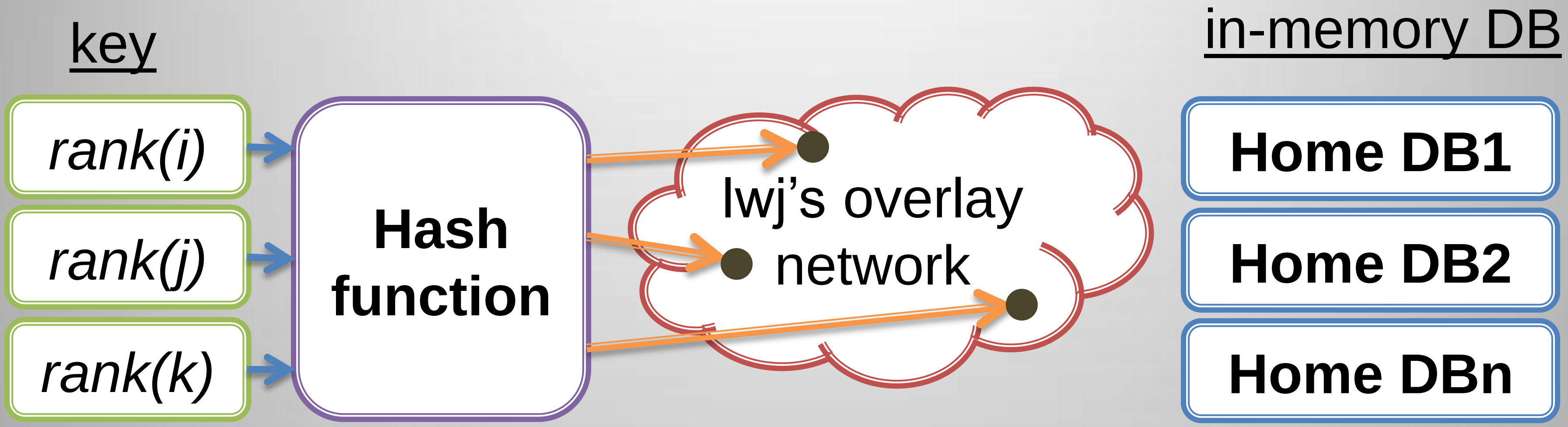
- grow and shrink for elasticity on any resource



The base architecture builds on comms. framework and distributed key-value store.



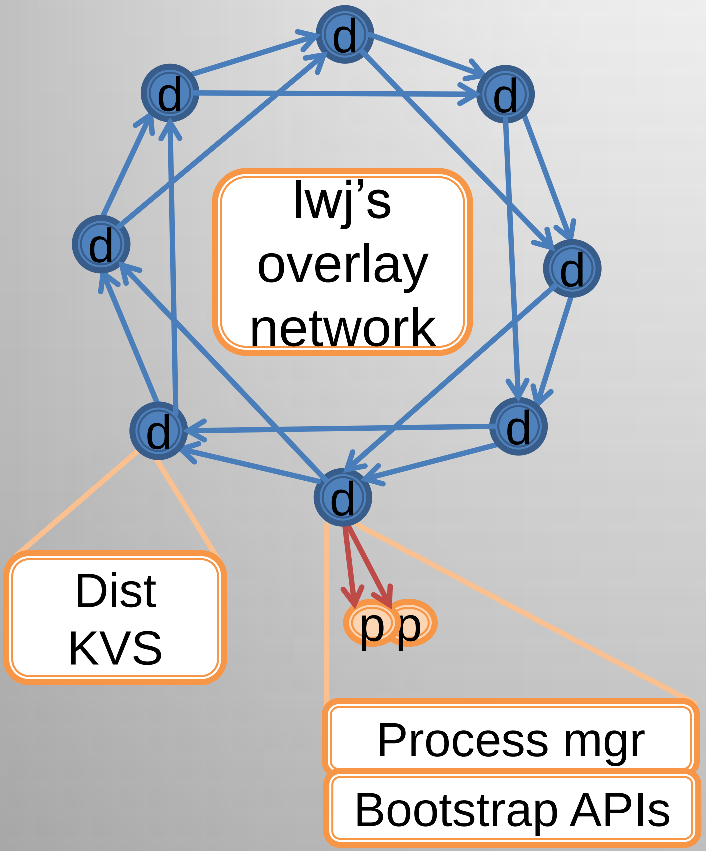
DKVS is scalable distributed shared memory for an LWJ and its descendants.



- Get/put for data access
- Collective Fence for memory consistency

lwj(1)::resource	cores (128)	power(10KW)	lic (10 tokens)	…
lwj(1)::rank(10)	host(1)	pid(345)	port(445)	
lwj(1)::record	info1	info2	…	…
lwj(1)::lwj(2)::resource	cores(64)	power(4KW)	lic (2 tokens)	

All process management operations must be scalable.

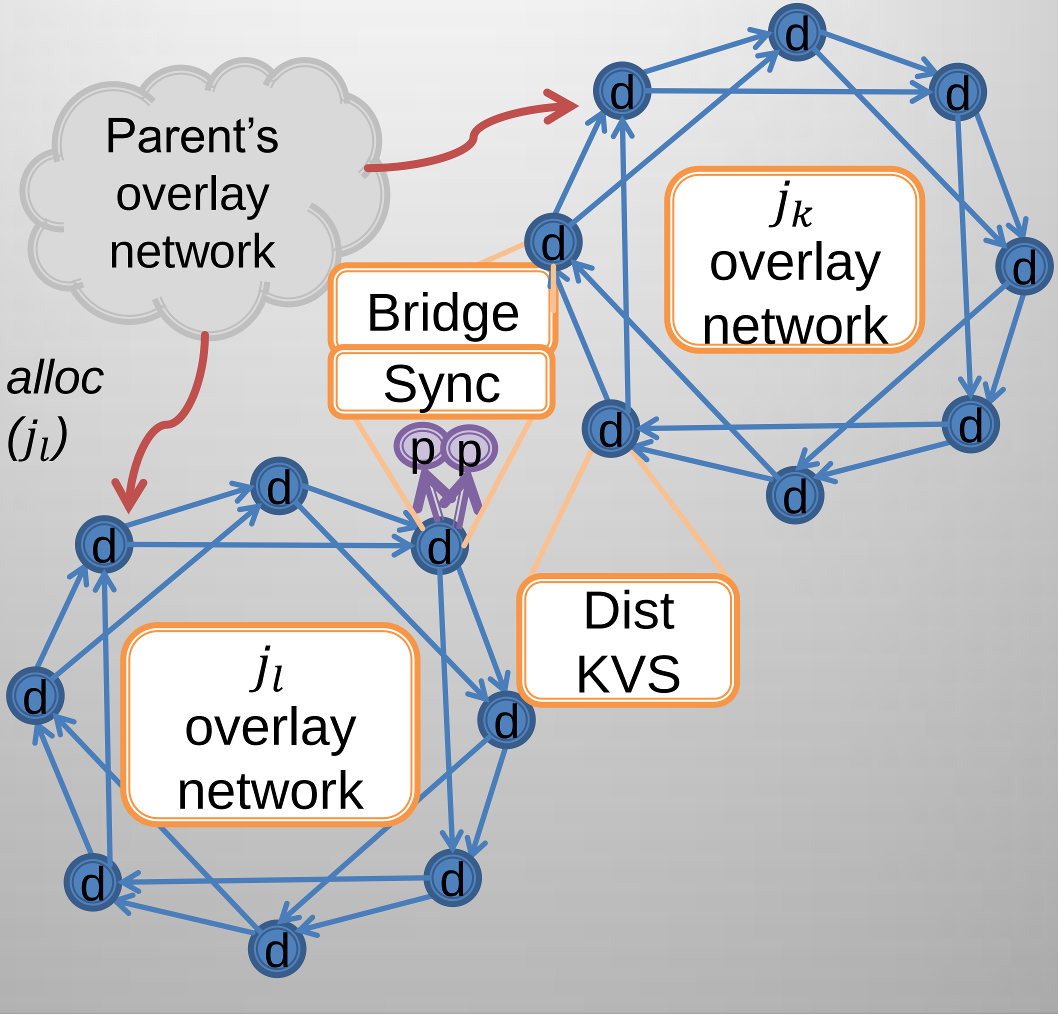


- Environment propagation: Bcast problem *O(log(N))*
- stdin/signal: *Bcast problem, O(log(N))*
- stderr/stdout handling: Reduction problem, *O(log(N)) with an ability to reduce the outputs on the fly*
- Process termination detection/analysis: Reduction problem: *O(log(N)) with an ability to reduce the outputs on the fly*

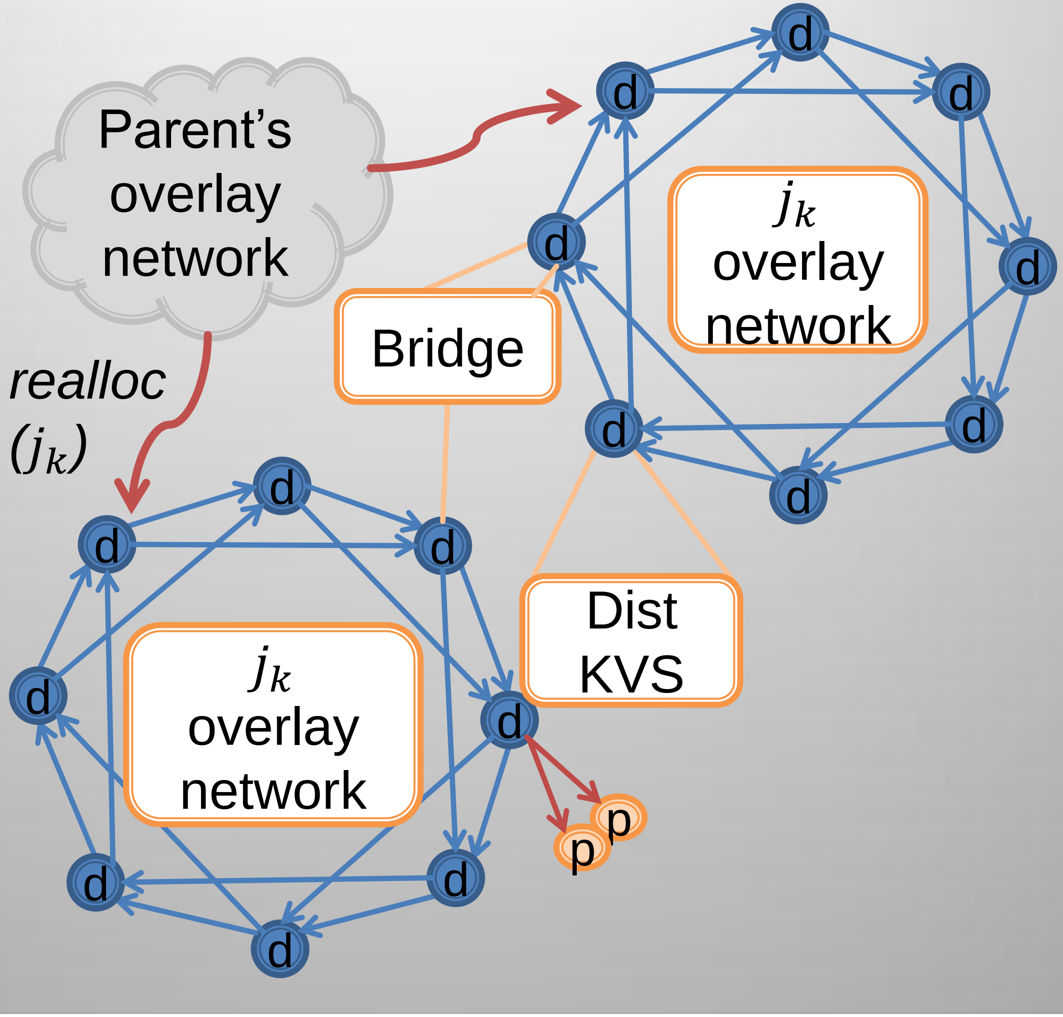
DKVS allows ease integration with various types of LWJs beyond MPI

- PMI 1, 2 will be a very thin layer on top of DKVS.
- PMGR, PMGR Collective, COBO, LaunchMON, and LIBI use essentially the same bootstrapping technique: launched processes discover open TCP ports of their children processes or parent process in the binomial tree.
- Our plug-ins for these well-known bootstrappers will serve as the reference implementation.
- Other types of LWJs can write their own plug-ins for ease integration into WRAP.

We can easily extend the base architecture to implement *sync*.



We can easily extend the base architecture to implement *realloc* w/ respect to CNs.



We devised a WRAP architecture that will allow us to create an effective development plan.

- WRAP has analyzed the requirements and use cases to conceive novel models as well as run-time services.
- WRAP devised a high-level architecture that can the services scalably.
- WRAP's architecture allowed us to identify the major components and their subsystems to develop (Unsolicited section).
- WRAP plans to use a phased approach to mitigate the risk of developing these components (Unsolicited section).
- WRAP work items will need to be considered in the context of the global Work Breakdown Structure (WBS).

Lawrence Livermore
National Laboratory