K-MS-Containers

Yinjie Liu / 20211091 / Group K

The main content about this paper is introduction about the Google container management systems, author spreads this topic from different aspects, what exactly these systems are, what significant influences they have bought to data center and implementation of applications, also it includes some typical problems which are solved or not to convenient our studying.

The container-management systems (Borg, Omega and Kubernetes), firstly it roughly introduces main purposes of three different systems, Borg was built to manage both long-running services and batch jobs. Omega, which is an offspring of Borg, was driven by a desire to improve the software engineering of the Borg ecosystem. And the last one, Kubernetes was an open-source contrast to Borg and Omega as well as makes it easier to deploy and manage complex distributed systems.

And then it shortly explains what container and image are, containers can provide isolation resources to drive utilization significantly and image make up the application that runs inside the container.

Author extended the container topic in two different ways: 1. containers encapsulate the application environment and 2. the shift from machine-oriented to application oriented. Hermetic container image isolate es applications from the operating systems and give us an example of Borg to show how achieve this image. In the next paragraph, it explains functionalities of containers in detail, for instances, containers can provide convenient points for information exchange, makes information displayed in interface, enables management systems communicate and provides application-oriented monitoring. Finally, it shows the relationship between containers and rationales in Borg and Kubernetes systems.

Next part of paper is to show methods to achieve consistent approaches in order to address the complexity of different services. It can be achieved by using uniform API, decoupling in the API or common design patterns for different components.

Author then presents some of common mistakes for us to avoid these, the first one is container system shouldn’t manage port numbers since container will be assigned a new port number when it moves to a new machine and service client do not know the port number. The second one is give containers labels/ label selectors, by this way, users can identify and manage them easily. We also should be careful with ownership, because there is only one grouping mechanism handling all use cases. The last one is because of different API architecture between container management systems, they contain diverse resources, and they should be hided in detail.

Following part of this paper presents two typical problems which haven’t fully addressed, the configuration issue led to appearance of a domain-specific configuration language and another problem which is dependency management depicts that whether cluster management system can automatically instantiate dependencies. The main content is showing the current development situation about these two problems.

**References**

[1] Burns B , Grant B , Oppenheimer D , et al. Borg, Omega, and Kubernetes[J]. Communications of the ACM, 2016, 59(5):50-57.

[2] Bazel: {fast, correct}—choose two; <http://bazel.io>.

[3] Burrows, M. 2006. The Chubby lock service for loosely coupled distributed systems. Symposium on Operating System Design and Implementation (OSDI), Seattle, WA.

[4] Verma, A., Pedrosa, L., Korupolu, M. R., Oppenheimer, D., Tune, E., Wilkes, J. 2015. Large-scale cluster management at Google with Borg. European Conference on Computer Systems (EuroSys), Bordeaux, France.

[5] Kubernetes; <http://kubernetes.io/>.

[6] Schwarzkopf, M., Konwinski, A., Abd-el-Malek, M., Wilkes, J. 2013. Omega: flexible, scalable schedulers for large compute clusters. European Conference on Computer Systems (EuroSys), Prague, Czech Republic.