

組込み機器向けコンテナオーケストレーションツールの評価

堀井 圭祐[†] 山田 竜也[†] 水口 武尚[†]

三菱電機株式会社 情報技術総合研究所[†]

1. はじめに

エンタープライズ系システムでは既に広く普及しているコンテナ系仮想化技術であるが、近年は組込み機器への適用も検討され始めている[1].

複数のサーバ向けにコンテナをデプロイするシステムでは、コンテナオーケストレーションツールを導入することで、コンテナの効率的な運用管理を実現可能である。エンタープライズ系システムでは Kubernetes がデファクトスタンダードであるが、組込み機器向けには機能過多であるうえにリソース消費量も多いことから、軽量版 Kubernetes として K3s が開発されている。そこで、K3s の実力を把握するために定性的側面、および性能面について評価する。また、Kubernetes 以外にも小規模システムに特化した MicroK8s を比較対象とする。

本稿はコンテナオーケストレーションツールの評価内容と結果をまとめたものである。

2. Kubernetes の概要

Kubernetes では、コンテナを Pod という単位で管理しており、Pod には 1 つ以上のコンテナが含まれる。また、個々のマシンを Node と呼称する。図 1 に示すように、Kubernetes のクラスタはコンテナを実行する Worker と、Worker を管理する Control Plane で構成される。

Kubernetes 適用によるメリットとしては「負荷分散」、「セルフヒーリング」、「デプロイ自動化」、「オートスケーリング」、「ストレージの自動マウント」などが挙げられる。

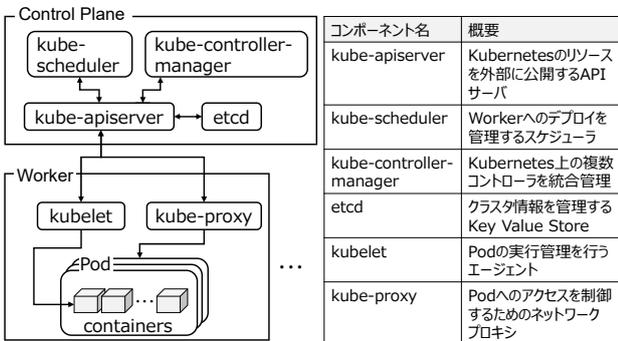


図 1 Kubernetes クラスタの構成図

3. 評価

3.1. 定性的な比較

K3s は Kubernetes から試験的な機能やストレージドライバ、クラウドプロバイダといったプラグインが削除されている。また、表 1 に示すように K3s では軽量版データストアの採用やシングルバイナリ化といった変更が適用されている。その他、図 1 に示す基本的なクラスタ構成は K3s や MicroK8s でも同様であり、2章で述べた Kubernetes の主なメリットも享受可能である。ただし、K3s は H/W 性能が低いマシンへの適用を想定しており、構築可能なクラスタ規模は CPU コア数とメモリ容量に依存する。

表 1 各ツールの特徴比較結果

	Kubernetes	K3s	MicroK8s
データストア	etcd	デフォルトはSQLite	デフォルトはDqlite
バイナリ数	複数	1	複数
セットアップ容易性	△	○	○

3.2. 性能測定

3.2.1. 測定概要

性能測定時の H/W・S/W 構成を表 2 に示す。表 2 の組込み機器を 2 台用意し、Control Plane と Worker が 1 台ずつのクラスタを構築する。また、次のテストシナリオ (Step. 1-6) に従って各ツールを操作した時の CPU 使用率とメモリ使用量を測定する。基本的に前の Step の処理完了を確認した後に次の Step の操作を実行する。ただし、定常時の性能を測定するために Step. 1 の前後と Step. 2, 3, 6 の後は 5 秒間の無操作区間とする。

- Step. 1: Control Plane 起動
- Step. 2: Worker 起動
- Step. 3: Example Pod (nginx) デプロイ
- Step. 4: Example Pod (nginx) 削除
- Step. 5: Worker 削除
- Step. 6: Control Plane 停止

表 2 性能測定時の H/W・S/W 環境

項目	仕様
組込みボード	R-Car H3 Starter Kit Premier
CPU	Arm Cortex-A57, 1.5GHz, 4 コア
メモリ	Arm Cortex-A53, 1.2GHz, 4 コア
ストレージ	LPDDR-4 4GiB
OS	eMMC 8GB
OS	Linux 5.10.41-yocto-standard, Ubuntu 20.04 LTS
Kubernetes	v1.25.4 ※コンテナランタイムは containerd v1.5.9 を使用
K3s	v1.25.4
MicroK8s	v1.25.4

An evaluation of container orchestration tools for embedded systems
 Keisuke Horii[†], Tatsuya Yamada[†], Takehisa Mizuguchi[†]
[†]Information Technology R&D Center, Mitsubishi Electric Corporation

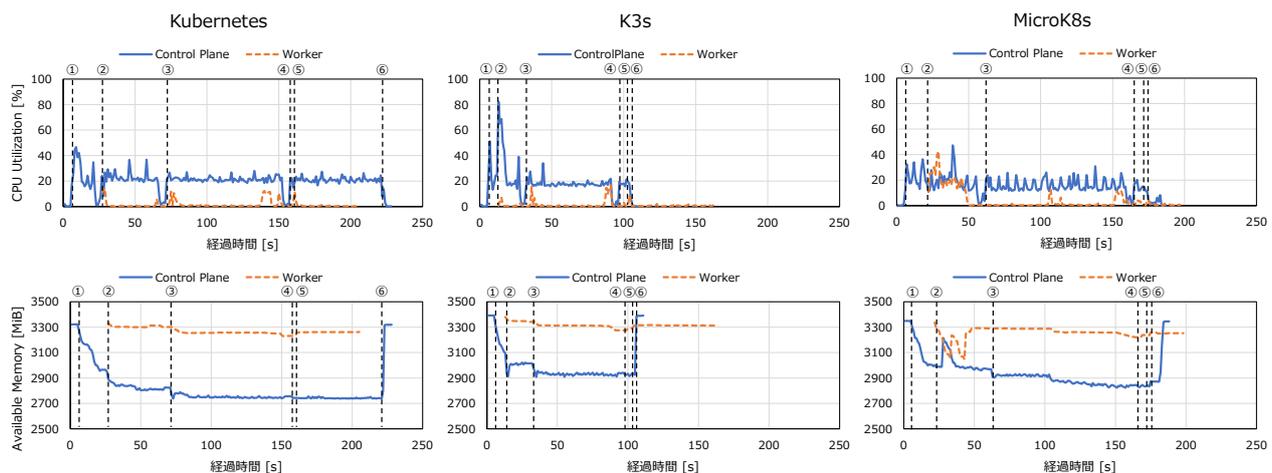


図2 性能評価結果（上段：CPU使用率，下段：使用可能なメモリ容量）

3.2.2. 測定結果

測定結果を図2に示す。図2の①～⑥は前節における性能測定シナリオの各Stepに対応しており、各Stepの実行時間はツール毎に異なる。また、CPU使用率の結果は8コアの平均値を示す。

まず、各シナリオの実行に要する時間を比較すると、“Step.4: Example Pod 削除”を除く全ての項目でK3sが最も高速であった。Kubernetesは他ツールと比較して“Step.2: Worker 起動”と“Step.5: Worker 削除”が特に遅く、MicroK8sは“Step.3: Example Pod デプロイ”が遅いという結果であった。

CPU使用率についてはControl Planeのピーク値はK3sが最も高く約82%であった。また、いずれのツールにおいても“Step.1: Control Plane 起動”と“Step.2: Worker 起動”での負荷が高くなる傾向であった。PodやNodeを操作していない定常区間のCPU使用率はいずれも2～4%程度であり大きな差はみられなかった。WorkerについてはMicroK8sの負荷が最も高く、ピーク値は約43%であった。また、Workerにおいてkubeletやkube-proxyなどのランタイムのみが動作している状態のCPU使用率はいずれも1%未満と非常に低負荷であった。

メモリについては最大値との差分を使用量として比較する。Control Planeにおけるピーク値、平均値ともにKubernetes > MicroK8s > K3sの関係にあり、Kubernetesのピーク値が約590MiB、平均値が約500MiB、K3sのピーク値が約480MiB、平均値が約385MiBであった。また、いずれのツールにおいてもクラスタの規模拡大に伴い徐々に増加していく傾向であった。

WorkerにおいてはMicroK8sの使用量が最も多く、ピーク値が約290MiB、平均値が約83MiBであった。また、KubernetesとK3sはピーク値が約106MiB、平均値が約65MiBで同程度であった。

4. 考察

定性的側面の比較結果より、K3sはKubernetesから組込み機器向けには不要な機能を取り除くことによって、軽量化を実現している。また、K3sではシングルバイナリ化によってインストール作業が簡易化されるだけでなく、Kubernetesにおけるプロセス間通信が最適化され、高速化に寄与していると考えられる。

性能面の測定結果より、K3sは今回の測定シナリオ完了までの時間が他のツールよりも2倍近く高速であり、クラスタの操作性能が他のツールよりも優れていることを確認できた。ただし、短時間に処理が集中して実行される分、CPU使用率のピークはK3sが最も高いという結果であった。他のアプリケーションと同時に実行する際には、処理を阻害しないよう優先度を設計するとともに、cgroupsなどによるリソース制限が必要になる可能性もある。また、K3sのメモリ使用量は3種類のツール内では最小であるものの、最小規模のクラスタで400MiB近く必要である。そのため、実運用時はクラスタの最大規模を考慮してK3sの適用可否を判断すべきである。

5. おわりに

組込み機器でのコンテナオーケストレーション実現に向けて、Kubernetes, K3s, MicroK8sの3種類のツールを定性的側面、および性能面で比較したところ、K3sはKubernetesの主な特徴は維持しつつ、クラスタ操作性の向上や省リソース化を実現できていることを確認した。

今後は、ネットワークやディスクアクセスなど他観点での性能評価を実施する予定である。

参考文献

- [1] 堀井圭祐ら，“組込み機器向けコンテナランタイムの評価”，第84回情報処理学会全国大会講演論文集，Vol. 1, pp. 11-12, Feb. 2022