

Linux Containers/cgroups BoF: サーバから組み込みまで “Container actually is all around”

高橋 雅彦

日本電気株式会社 システムプラットフォーム研究所

概要: 仮想化技術の一つであるコンテナ技術は、OS レベルで仮想化することによってオーバヘッドが非常に小さい仮想化環境を提供できる技術である。Linux のコンテナ技術に関する議論と実装は、過去数年間にわたって主に Containers メーリングリストで行なわれており、昨年からその一部がメインラインに入るに至った。しかしながらその機能はまだ本来のコンテナ技術のほんの一部に過ぎず、これからも議論と機能拡充を続ける必要がある。

このメーリングリストでは、主に資源管理機能と名前空間機能とチェックポイント/リスタート機能の 3 つの機能が議論されている。どの機能をとっても従来の Linux には不足していた機能であり、Linux が単なるデスクトップ OS に留まらずにサーバ・スーパーコンピュータから組み込み機器までをカバーする OS として利用されるために必要不可欠な機能であると考えられる。

本 BoF では、今後、これらの機能を実装しメインライン化するにあたって、参加者の各領域（組み込みからサーバまで）の視点から「コンテナ技術のあり方、欲しい機能」などの議論をして欲しいと考えている。また、会場で議論をするだけに留まらず、発言者や聴衆の人々の将来的な ML への意見投稿やパッチ投稿へとつながる、実のある BoF になることを望んでいる。

I. コンテナ技術

コンテナ技術は仮想化技術の一つであり、ハードウェアレベルの仮想化や命令セットレベルのエミュレーションを行なう他の仮想化とは異なり、OS レベルの仮想化である。OS レベル仮想化なので極めて低オーバヘッドな仮想化環境が提供でき、実際に同様の機能を提供している Solaris 10 におけるオーバヘッドは 1.6% (4 ゾーンの場合) である [7]。

現在の Linux コンテナでは資源管理と名前空間の機能が実装され始めている。また、実行状態を保存するチェックポイント・リスタート (C/R) 機能の検討も進められている [1]。

一般に OS 上では複数のアプリケーションが動作しているが、特定のアプリケーションがハードウェア資源 (CPU、メモリ、I/O 帯域など) を占有するのを防ぐ機能が資源管理である。資源によってはアプリケーションの使用帯域を保証することも可能である。また、C/R 機能を用いると、障害発生時に過去の保存状態から計算を継続できたり、負荷分散のために実行環境 (プロセス群) を他の計算機環境に移動することが可能となる。この C/R 機能の実現のためには名前空間の独立が必要なため、現在は名前空間の実装が先行して行なわれている。勿論、セキュリティの観点からも名前空間の独立は重要視されている。

II. LINUX コンテナメーリングリスト (CONTAINERS ML)

Linux をベースにしているコンテナの実装は数年前からいくつかが存在した。例えば、IBM の MetaCluster[3]、Parallels (旧 SWSoft) の OpenVZ[6]、Linux-VServer プロジェクト [5]、コロムビア大学の Zap[4] などである。このうちのいくつかはオープンソースとして公開されているが、いずれもメインライン化はされていなかった。コンテナ ML[2] の存在意義はそこにある。最近では、上記の開発者を始めとした多くのエンジニアが ML 上で議論を繰り広げている。

このコンテナ ML での議論の成果は、昨年からその一部がメインラインに取り込まれるに至った。例えば、cgroups フレームワークや、CPU やメモリの資源管理である。また、

プロセス ID を始めとする名前空間の実装も一部が取り込まれた。しかしながらこれらの機能はまだ本来のコンテナのほんの一部に過ぎない。

また、2007 年からはミニサミットが非定期的に開催されている。OLS などのコンファレンスで発表または聴講する人が多い機会を利用して、今後のコンテナ技術の方向性の議論や確認をする face-to-face の場という位置付けである。特に 2008 年のミニサミットでは C/R の今後の活動方針に関しての議論がなされ、その結論に沿った行動が、ミニサミット後に早速、行なわれた。オープンソースの活動は ML でなされることが多いが、その水面下では face-to-face で決められていることも少なからず存在するようである。

III. BoF で議論したいこと

先述した 3 つの機能のどれをとっても従来の Linux に不足していたものであり、Linux が単なるデスクトップ OS にとどまらずサーバから組み込み機器までをカバーする OS として利用されるためには必要不可欠な機能である。例えば最近のサーバではハードウェアの急速な性能向上に伴って複数のシステムを 1 筐体に統合する仮想化技術が多用されており、そのためには資源の仮想化はもちろんのこと、1 筐体の資源をシステム毎に割り振り、それを制御する機能が必要となる。また、負荷変動に追従してシステム環境を移動させられれば、運用の柔軟性も上がる。

他方、組み込み分野でも Linux が広く使われている。特に組み込みは資源が限られているので、資源管理は重要である。例えば、マルチメディアアプリケーションでは I/O 使用帯域を確保する必要がある。また、i アプリに代表されるように、あとからシステムに追加する外部アプリケーションが今後は増える予想されており、その場合に、外部アプリケーションの使用資源を制御し、同時に資源の独立性を保つことも重要である。

そう、コンテナ技術はいたるところで利用可能であり、将来的にはそこかしこで使われるだろう。

本 BoF では、参加者の専門分野の視点からコンテナ技術に対する意見や要望を述べて頂きたい。それはコンテナ技術のメインライン化にとって非常に有益なものである。そして、単なる傍観者に留まらず、一人でも多くの日本人がコンテナ ML で活躍されることを望む次第である。

REFERENCES

- [1] Sukadev Bhattiprolu, et al., Virtual Servers and Checkpoint/Restart in Mainstream Linux, ACM SIGOPS Operating Systems Review, Vol. 42, Issue 5, 2008
- [2] <https://lists.linux-foundation.org/mailman/listinfo/containers>
- [3] Cédric Le Goater, et al., Making Applications Mobile Under Linux, Ottawa Linux Symposium, 2006
- [4] Oren Laadan, et al., Transparent Checkpoint-Restart of Multiple Processes on Commodity Operating Systems, USENIX Technical Conference, 2007
- [5] Linux-VServer, <http://linux-vserver.org/>
- [6] Andrey Mirkin, et al., Containers Checkpointing and Live Migration, Ottawa Linux Symposium, 2008
- [7] <http://jp.sun.com/company/partners/sap/report/feature01.html>