Linuxの長期安定運用を目指した互換性検証技術

カーネルの互換性検証で新しいハードウェアへ移行するときのリスクを軽減

長期的に使われる製品でLinux(注1)を利用するには，運用期間中にハードウェアの変更が必要となった際に，カーネルバージョンを変更するか，又は新しいカーネルから必要機能を取り込むかなどの対応案を事前に検討しておく必要があります。なぜなら，変更前のハードウェアとソフトウェアの組合せで実現していた仕様が，変更後の組合せでは保証できないおそれがあるためです。

東芝は，長期間運用する製品でLinuxを安定して稼働させる施策の一つとして，カーネルバージョン間の互換性検証技術の研究を進め，性能面だけでなくサービス品質まで含めた互換性を検証するテスト環境を開発しました。

製品を長期間供給するうえでの制約

東芝の製品の中には，10年以上にわたり出荷が継続されているものが多数あります。しかし，ハードウェアは日進月歩で進化しており，出荷継続中であっても，100％同じ構成の製品を提供し続けることが困難な場合があります。

製品の構成とその変更の形態を**図1**に示します。ソフトウェアは，アプリケーション，ライブラリ，及びカーネルから構成されます。ハードウェアの変更が必要な場合，アプリケーションやライブラリまで含めて変更することは，保守性を大きく損ねることになります。そこで，アプリケーションを変更せずに対処するには，ハードウェアとアプリケーションの間に位置するカーネルで相違点を吸収することが有効な手段となります。

その際には，どのバージョンのカーネルを用いれば機能，性能及び信頼性の面で必要な互換性が確保できるかを検証し，カーネルを移行したときの影響を見極める必要があります。このとき，アプリケーションが単に動作可能というだけでなく，移行後の構成で“同等性”を確認することが重要です。つまり，機能面での同等性のほかに，性能や信頼性（以下，まとめてQoS（Quality of Service：サービス品質）と言う）の面での同等性についても確認が必要です。

単にLinuxカーネルと言うと同じもののように思えますが，実は，互換性に関して様々な問題を抱えています。これは，約3か月に1回というリリースサイクルの中で，新しい機能の追加を含む多くの修正が行われていることに起因しています。

当社は，カーネルバージョン間の移行作業をよりスムーズに進めるため，Linuxの互換性問題の影響を見極める互換性検証技術を研究しています。そしてその成果を，Linuxコミュニティなどを通して広く一般に公開しています。

カーネル互換性検証テストの構成

互換性を確保するには，カーネルの個々の機能が同等に動作するだけでなく，異なるハードウェアとソフトウェアの組合せでも同等に動作するQoS面での検証が必要です。そこで，検証用のテストツールを**図2**に示す構成としました。

共通部では，各テストで共通に利用できる，ソースコードを取得する処理や結果出力フォーマットを整形する処理などを行います。カーネル部では，移行対象のハードウェアに対して複数のLinuxカーネルのテストができるように，指定されたカーネルのインストールやブートローダの設定を行います。テスト部では，機能面の“API（Application Programming Interface）テスト”と，QoS面の“QoSテスト”で構成される互換性検証テストを行います。

まず，APIテストでは，アプリケーションとLinuxカーネル間のインタフェースに関するテストを行います。例えば，OSS（Open Source Software）のテストスイートのLTP（Linux Test Project）では一連のシステムコールテストを提供しているため，これを行います。

次に，QoSテストでは，APIテストでは発見できないパフォーマンス面や，ファイルシステム，スケジューラ，及びデバイスドライバなど，カーネルの各サブシステムの機能に関する動作特性面をテストします。例えば，ファイル書込み中に電源断となる状況が数千回発生した場合にデータの損失がどの程度発生するか，又は，ネットワークを通したリクエストが100 msの間に10回ある場合でも各リクエストに対して500 ms以内に応答が可能か，といったテストです。これらのテストにはしきい値が設定され，その範囲に収まっているかどうかが非常に重要な観点になります。

実際に，Linuxではカーネルバージョンの変更に伴いカーネル内部の個々の機能の実装方法が変更され，QoS面で大きな差が現れることがあるので，QoSテストをサポートしていることがこのテストツールの大きな特長です。

テストの実行環境

カーネルの動作特性は，実機で検証することが不可欠で，**図3**に示すようなテスト実行環境を構築しました。ターゲットシステムとして，移行前のカーネル及びハードウェアの組合せと移行後のカーネル及びハードウェアの組合せを用意し，ホストシステムと協調して動作可能な設定を行います。ターゲットシステムそれぞれで必要なテストを行って結果を比較します。

テストの自動実行

APIテストとQoSテストを様々な動作特性に対して全て行うには，非常に長い時間が必要です。例えば，電源断の際のデータ信頼性テストには，1日から1週間程度の時間が掛かります。

こういったテストを効率的に行うためには自動化が不可欠です。しかし，OSSとして公開されているテストスイートや，社内で開発したテストスイートは，個々に違う設定方法や解析方法になっており統一性がありません。そこで図2のテスト環境では，各テストのシナリオをセットアップ，実行，結果取得，及び解析という四つのフェーズで定義し，それぞれのテンプレートを用意することで流れの自動化を容易にしました（**図4**）。

テストを追加する際，このテンプレートをベースにテストごとに異なる設定方法や解析方法の差を吸収するようにすることで，テストの効率化を進めています。

今後の展望

今後は，QoS面でのテスト内容を拡充し，Linuxカーネルの移行に対して，より精度の高い見積りを行えるようにするとともに，これらのテストを通して，Linuxを利用する製品の品質向上に取り組みたいと考えています。

小林　良岳
ソフトウェア技術センター
先端ソフトウェア開発担当主務

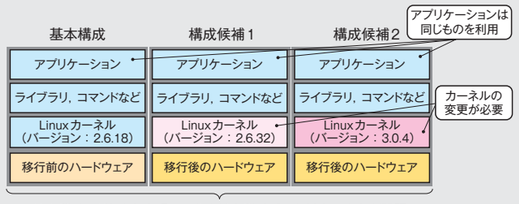

図1．カーネルの移行を想定したときの製品構成 ― 製品のハードウェア部分を変更してもアプリケーションの改変が難しい場合は，カーネルバージョンとハードウェアの複数の組合せでテストを実施します。

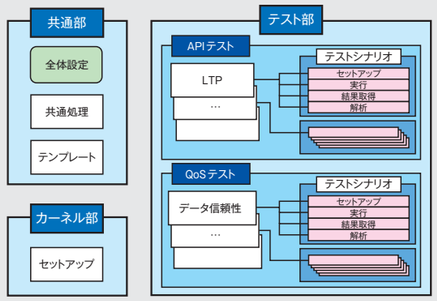

図2．テストツールの構成 ― 様々なテストを行うためのテンプレートの位置づけで作成された，テスト実行のためのスクリプトやテンプレートの集まりで，カーネル部とテスト部，及びこの両者から利用される共通部で構成されます。

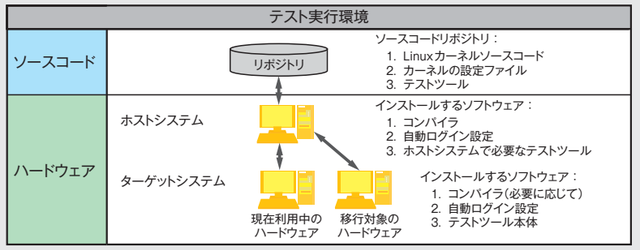

図3．テスト実行環境の構成 ― テスト実行環境はテスト実施の指示を出すホストシステムと実際にテストを実行するターゲットシステムで構成され，それぞれがネットワークで接続されています。

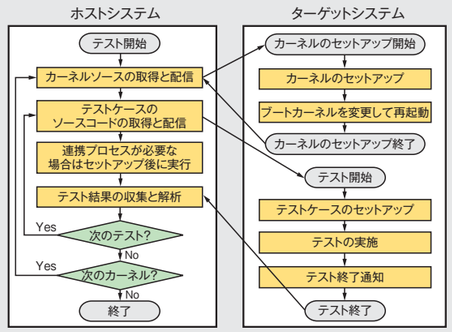

図4．自動化されたテストの流れ ― テスト実行環境はホストシステムとターゲットシステムが協調して動作することを想定しており，ホストシステムの指示により解析まで含めて自動化することができます。

(注1)　Linuxは，Linus Torvalds氏の米国及びその他の国における登録商標。