

# 仮想実行環境を管理するためのライブラリ

大山 恵 弘<sup>†,††</sup> 米澤 明 憲<sup>†,††</sup>

仮想実行環境を管理するためのライブラリ VEEML (Virtual Execution Environment Management Library) を提案する。VEEML の利点は 3 つである。第 1 に、非常に多数の仮想実行環境を生成・管理する作業を容易にする。第 2 に、仮想実行環境を遠隔計算機に配置するための簡便な仕組みを提供する。第 3 に、仮想実行環境を提供する複数のシステム ( 複数の仮想化システム ) の共存を考慮しており、それらのシステムの連合を促進する。

## A Library for Managing Virtual Execution Environments

YOSHIHIRO OYAMA<sup>†,††</sup> and AKINORI YONEZAWA<sup>†,††</sup>

We propose a library for managing virtual execution environments, VEEML (Virtual Execution Environment Management Library). VEEML has three advantages. First, it supports the creation and management of numerous virtual execution environments. Second, it provides a convenient mechanism for deploying virtual execution environments on remote machines. Finally, it allows the coexistence of multiple virtualizing systems and encourages their federation.

### 1. はじめに

仮想実行環境を構築するシステム ( 以下、仮想環境システム ) が広く利用されている。仮想環境システムの例としては、VMware<sup>8)</sup>, Xen<sup>1)</sup>,  $\mu$ Denali<sup>9)</sup>, LilyVM<sup>2)</sup>, User-Mode Linux<sup>6),7)</sup>, QEMU<sup>5),10)</sup>, SoftwarePot<sup>4)</sup>, Virtual Multiprocessor<sup>3)</sup> などがある。近年の計算機能力向上と仮想環境システムの技術発展により、従来に比べ現実的な性能およびコストで仮想環境システムを利用できるようになっている。

いくつかの軽量な仮想環境システムでは、1 台の計算機上で多数 ( たとえば数十 ) の仮想実行環境を同時に運用することが可能である。数十台規模の計算機群を利用すれば、それらの上に数千個規模の仮想実行環境からなる仮想空間を構築することが現実的となる。そのような仮想空間は、分散システムの開発・デバッグ、ホスティング、ウィルスおよび侵入解析用のテストベッドなどの用途に極めて有用である。

現状では、そのような仮想空間を構築・管理する作業は容易ではない。既存の仮想環境システムと汎用スクリプト言語を組み合わせて仮想空間を構築・管理す

る場合、ユーザは煩雑な処理を記述する必要がある。たとえば、仮想実行環境を遠隔計算機に配置したり、仮想実行環境間で同期を取るための処理の記述が必要である。もしそのような処理がライブラリなどの再利用可能な形でパッケージ化されていれば、ユーザはより容易に仮想空間を利用することができる。

本研究では、仮想実行環境を管理するためのライブラリ VEEML (Virtual Execution Environment Management Library) を提案する。VEEML は、

- 多くの仮想実行環境を扱うこと
- 複数の物理計算機上に仮想実行環境を配置し、それらの仮想実行環境を同期させること
- 異なる仮想環境システムを混在させて利用すること

の単純化を狙っている。

### 2. VEEML

VEEML の基本アイデアは以下の 3 つである。

- プロセス制御 API ( exec, kill, wait 等 ) に似た、直感的にわかりやすいインタフェースによって仮想実行環境を生成/操作できるようにする
- 仮想実行環境を特定するための globally unique な ID を導入し、遠隔計算機に配置された仮想実行環境を簡便に操作できるようにする
- 様々な仮想環境システムの混在を意識したインタ

<sup>†</sup> 東京大学

The University of Tokyo

<sup>††</sup> 科学技術振興機構

Japan Science and Technology Agency

表 1 VEEML API の一部  
Table 1 A part of VEEML API.

仮想実行環境の開始	<code>vid_t vee_exec(vee_type_t vt, char *snapshot_file, place_t place, bool_t start_now);</code>
仮想実行環境の保存	<code>int vee_snapshot(vid_t vid, char *snapshot_file, bool_t continue);</code>
仮想実行環境の停止	<code>int vee_suspend(vid_t vid);</code>
仮想実行環境の再開	<code>int vee_resume(vid_t vid);</code>
仮想実行環境へのイベントの待機	<code>int vee_wait(vid_t vid, int *status);</code>
仮想実行環境へのイベントの送信	<code>int vee_kill(vid_t vid, int sig);</code>
仮想実行環境の差分ファイルの作成	<code>int vee_makediff(char *difffile, char *snapshot1, char *snapshot2);</code>
仮想実行環境の差分ファイルの融合	<code>int vee_mergediff(vid_t vid, char *difffile);</code>
仮想実行環境の優先度の取得	<code>int vee_getprio(vid_t vid);</code>
仮想実行環境の優先度の更新	<code>int vee_setprio(vid_t vid, int priority);</code>

### フェースにする

VEEML は仮想実行環境の生成・配置・操作・同期などの機能を提供するライブラリである。VEEML の API の一部を表 1 に示す。アプリケーションプログラマは自分のプログラムに VEEML をリンクして利用する。VEEML は、仮想実行環境を管理するためのコマンドプログラムの構成要素としても利用される。

VEEML がリンクされたプログラムを分散環境における複数の計算機上で実行するために、サーバデーモン `veemd` を用いる。`veemd` は仮想実行環境のスナップショットをネットワークから受け取り、その仮想実行環境を実行する処理を繰り返す。仮想実行環境のマイグレーション処理も行う。

各仮想実行環境には作成時に仮想実行環境 ID (VID) が割り当てられる。VEEML が提供する関数の引数に VID を与えることにより、その VID の仮想実行環境に関するイベント発生を待つことや、その仮想実行環境にイベントを送ることができる。たとえば、ある仮想実行環境内のプログラムが、別の仮想実行環境の終了を待ったり、別の仮想実行環境のマイグレーションを命令することができる。VID は globally unique である。仮想実行環境が別の計算機に移動しても、その仮想実行環境を特定するための VID は変化しない。

### 3. 関連研究

SBUML<sup>6)</sup> の研究において、仮想実行環境を操作するためのコマンド集が提案されている。本研究は SBUML から大きな影響を受けている。本研究では、分散環境上に多くの仮想実行環境を配置して管理するという応用に注目し、SBUML などの汎用仮想環境システムをその用途に適用するために追加すべき機構について考察した。

仮想機械モニタ Xen<sup>1)</sup> のパッケージには、仮想計算機の生成や制御を司るデーモン `Xend`、仮想計算機をコマンドラインから制御するためのコマンド `Xm` が

含まれている。それらは Xen が作る仮想計算機のみを対象としているが、VEEML および `veemd` は複数の仮想環境システムが連合することを意識して設計されている。

### 参考文献

- 1) P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the Art of Virtualization. In *Proceedings of SOSP '03*, pp. 164–177, 2003.
- 2) H. Eiraku and Y. Shinjo. Running BSD Kernels as User Processes by Partial Emulation and Rewriting of Machine Instructions. In *Proceedings of BSDCon '03*, pp. 91–102, 2003.
- 3) K. Kaneda. Virtualizing a Multi-processor Machine on a Network of Computers. <http://www.yl.is.s.u-tokyo.ac.jp/~kaneda/dvm/>.
- 4) K. Kato and Y. Oyama. SoftwarePot: An Encapsulated Transferable File System for Secure Software Circulation. In *Software Security – Theories and Systems*, Vol. 2609 of LNCS, pp. 112–132, 2003.
- 5) QEMU CPU Emulator. <http://fabrice.bellard.free.fr/qemu/>.
- 6) O. Sato, R. Potter, M. Yamamoto, and M. Hagiya. UML Scrapbook and Realization of Snapshot Programming Environment. In *Software Security – Theories and Systems*, Vol. 3233 of LNCS, pp. 281–295, 2004.
- 7) User-Mode Linux. <http://user-mode-linux.sourceforge.net/>.
- 8) VMware. <http://www.vmware.com/>.
- 9) A. Whitaker, R. S. Cox, M. Shaw, and S. D. Gribble. Constructing Services with Interposable Virtual Hardware. In *Proceedings of NSDI '04*, pp. 169–182, 2004.
- 10) 尾上, 大山, 米澤. Quasar: CPU エミュレータ QEMU を利用した移動計算システム. 第 8 回プログラミングおよび応用のシステムに関するワークショップ (SPA 2005), 2005.