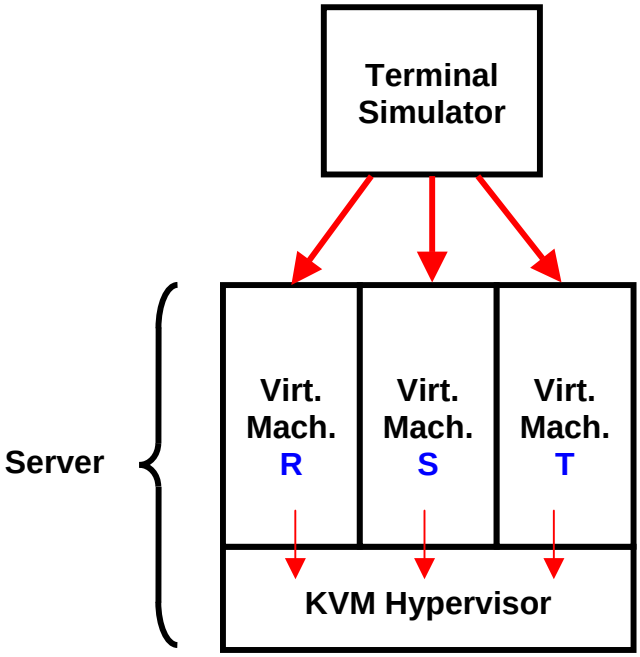
Aufgabenstellung der Masterarbeit

Für das WLM Projekt einen experimentellen Prototyp erstellen und erste Experimente damit anstellen.



Die Prototyp Konfiguration besteht aus zwei Komponenten, einem Terminal Emulator und einem Server.

Auf dem Server läuft der KVM Hypervisor. Unter dem Hypervisor laufen drei virtuelle Maschinen R, S und T. Derzeitige Annahme: alles in Linux, aber Windows ist als Alternative denkbar. KVM ist public domain, kann von z.B. http://www.linux-kvm.org/page/Downloads heruntergeladen werden, ist aber bereits Bestandteil der meisten Linux Distributionen. Der KVM Hypervisor und die drei virtuellen Maschinen R, S und T besitzen je ihre eigene IP Adresse. Für den Server steht uns bereits ein eigener Rechner zur Verfügung, der nur für das WLM Projekt benutzt wird, und über das Internet zugreifbar ist: tipc087.informatik.uni-leipzig.de.

Der Terminal Emulator ist entweder ein eigener Rechner, oder, was vermutlich vorzuziehen ist, eine weitere virtuelle Maschine X unter KVM. Er sendet laufend Nachrichten mit einer Variablen mit dem Wert 1 an die drei virtuellen Maschinen. Auf den virtuellen Maschinen R, S und T läuft je eine primitive Anwendung, die z.B. nichts anderes tut, als die Variable von 1 auf 100 000 hochzuzählen, und eine Antwort an den Terminal Emulator zurückzusenden.

Der virtuelle Speicherplatz für die Virtuellen Maschinen R, S und T ist so knapp zu bemessen, dass regelmäßig Fehlseitenunterbrechungen auftreten, z.B. in dem man in das Anwendungsprogramm `malloc` Befehle einsetzt (möglicherweise ist es am einfachsten, die Anwendung in C++ zu schreiben)

Der KVM Hypervisor ist jetzt so zu instrumentieren, dass er die Häufigkeit der Fehlseitenunterbrechungen registriert, in einer Datei sammelt und für eine Auswertung (später durch WLM) verfügbar macht.

Damit existiert ein Test Vehicle, mit dem weitere und anspruchsvollere Scenarien durchgespielt werden können.

Aufgabe der Masterarbeit

Schritt 1

A. Einen Server und Terminal Emulator wie oben beschrieben installieren und lauffähig machen.

- **KVM installieren**
- **Virtuelle Maschinen installieren**
- **Anwendung für virtuelle Maschinen schreiben (sollte nur wenige Zeilen Code enthalten)**

B. Terminal Emulator generieren

- **Code schreiben (sollte nur wenige Zeilen Code enthalten)**
- **Client/Server Operation erstellen und austesten**

C. Eine Erweiterung in KVM erstellen (Prototyp Rahmen der Mustererkennungskomponente)

- **Vertraut machen mit KVM Internals. Dokumentation unter http://www.linux-kvm.org/page/Documents.**
- **Misst die Anzahl der Fehlseitenunterbrechungen getrennt für die Virtuellen Maschinen R, S und T**
- **Erstellt hierüber eine Zusammenfassung, und macht sie von außen in Form einer File zugänglich**

Schritt 2

Je nachdem, wie viel Zeit Schritt 1 in Anspruch genommen hat:

- **das in dem Dokument „notes20111219.doc“ auf Seite 2 als „Ansatz A“ beschriebene Vorgehen teilweise oder ganz implementieren, und Prototyp der Mustererkennungskomponente erweitern.**
- **Falls immer noch Zeit vorhanden ist, das in dem Dokument „notes20111219.doc“ auf Seite 3 als „Ansatz B“ beschriebene Vorgehen untersuchen, diskutieren und evtl. teilweise implementieren. Evtl. Erweiterung der Mustererkennungskomponente.**

Schritt 3

Das Ganze als schriftliche Masterarbeit dokumentieren.

Eine wichtige Komponente der Masterarbeit ist das Kapitel „Ausblick“, was beschreibt, welches die Aufgaben der follow-on Masterarbeiten sein sollten.