

Mit pm-utils schlafen gehen und aufwachen

Detlef Lechner

ubuntu-berlin

26. April 2008

c-base Berlin-Mitte

pm-utils - Gliederung

- 1. Einführung
- 2. Definitionen
- 3. Gerüst
- 4. Hauptprogramm: pm-action
- 5. Programmablauf
- 6. Fehlersuche
- 7. pm-utils in der Energieverwaltung
- 8. Dialog „Beenden“

Einführung: Flash

- **pm** – Power Management = Energieverwaltung
- **utils** – utilities = Dienstprogramme
- „The next-generation suspend infrastructure“
- „power management scripts for suspend and hibernate“
- “Pm-utils provide simple shell command line tools to suspend and hibernate a computer. They can be used to run vendor, distribution, or user supplied scripts on suspend and resume.”
- „pm-utils is a small collection of scripts that handle suspend and resume on behalf of HAL.“

pm-utils - grüne Technik



Einführung: Umriß

- **Ziel:** Alle wichtigen Linux-Distributionen implementieren dieselbe Funktionalität für die Energieverwaltung.
- **Standardisierte, distributions-unabhängige Vorlage.**
- **Interface für die allen Distributionen gemeinsamen Energieverwaltungsaufgaben.**
- **Software- und Hardware- Lieferanten brauchen keine distributions-spezifische Funktionalität für Suspend und Resume zur Verfügung zu stellen.**

Stand der Einführung

- Red Hat 2006
- Suse 10.2
- **Ubuntu Gutsy > Hardy**
 - mjpg59: "Don't build s2ram. It's not sensible on Ubuntu. -- Because we use pm-utils. And, in earlier versions, acpi-support. There's no reason to provide multiple packages that do exactly the same thing. If it doesn't work out of the box, that's a bug."
- **Ubuntu und SuSE haben zum Teil noch ihre eigenen Flicker.**

2. Definitionen

- **sleep** i) ACPI-Zustände ii.) Zustandsübergang
- **suspend** > suspend to RAM ~ ACPI S3
- **resume** > resume from RAM
- **hibernate** = suspend to disk = ACPI S4
- **thaw** > resume from disk
- **hook** ein Skript, das von den pm-utils ausgeführt wird.
- **quirk** <erUSUL>: "A quirk is a routine or a variable in the kernel which you program to make hardware work that has abnormal or out-of-spec behavior."

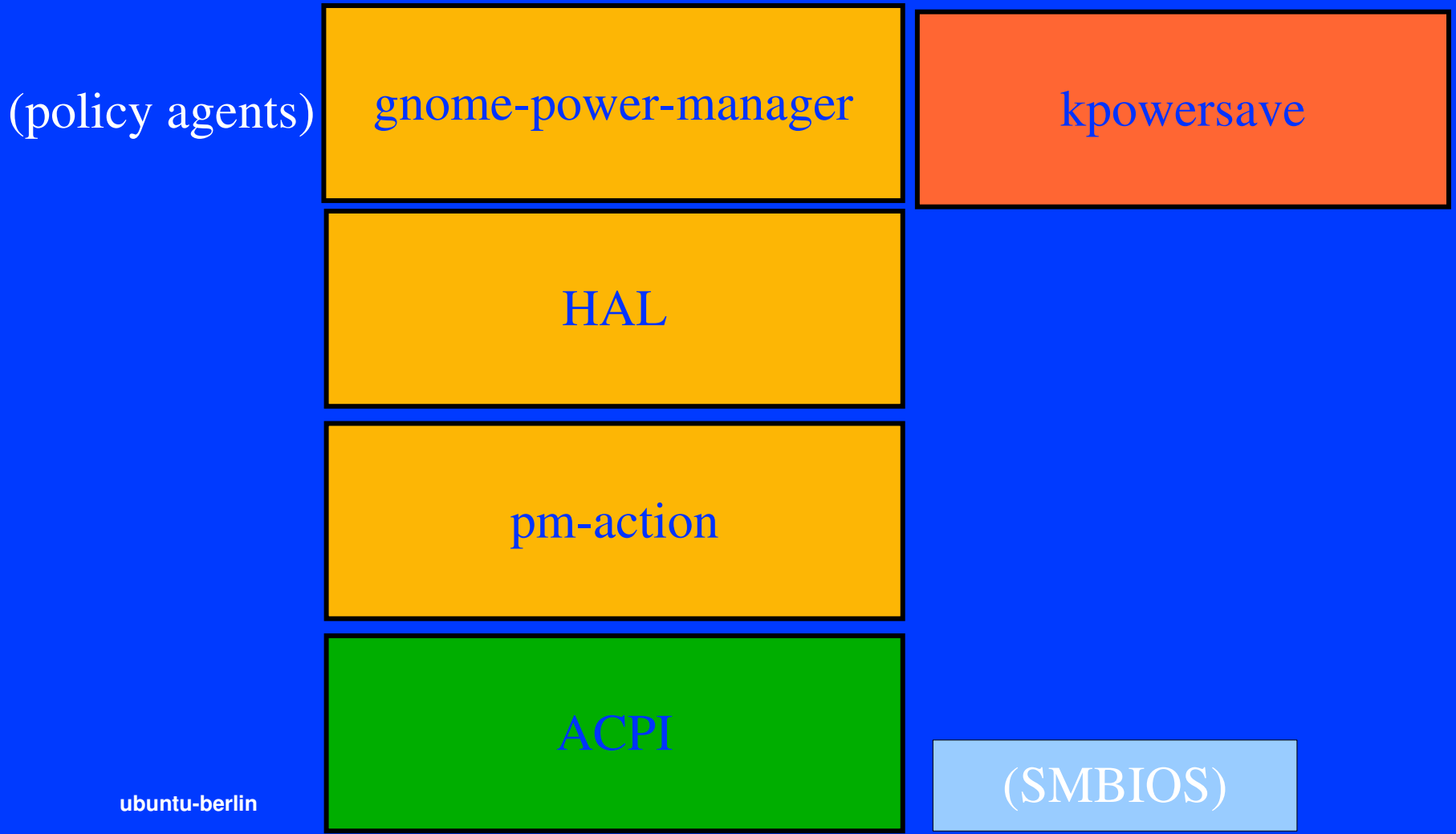
3.1. Gerüst

- **pm-utils: Gerüst, das künftig alle Linux-Distributionen benutzen werden.**
 - pm-utils = Werkzeuge. Arbeiten in Verbindung mit HAL. Schalten verschiedene Systemzustände um.
 - Die Umschaltregeln stellen die 'Policy Agents' auf. Policy Agents laufen im User-Kontext auf dem Desktop. Beispiele: gnome-power-manager, kpowersave. (Bild 1)
 - Für jeden Zustandswechsel werden Aktionen ausgeführt, ähnlich wie ein runlevel-Wechsel beim Booten. Beispiele: Suspend <-> Resume, Netzversorgung <-> Batterieversorgung.
 - Aktionen als Skripte realisiert.
 - pm-utils wurden von HAL als offiziell unterstütztes Backend auserkoren. Es findet also gewissermassen eine Konsolidierung in dem Bereich Energieverwaltung statt.

3.2. Orte

- **Orte für Konfiguration und Skripte**
 - i.) Default-Konfiguration (mitgebracht von Paketen oder pm-utils) **In /usr/lib/pm-utils/**
 - ii.) Vom Systemadministrator geändert/eingefügt **In /etc/pm/sleep.d/.** (ii.) hat Vorrang vor i.)
 - **Skripte für Aktionen In den Konfigurationsverzeichnis-en /etc/pm/power.d/ , /etc/pm/sleep.d/ .**
 - **Konfigurationsdateien In /etc/pm/config.d/.**
Gute Vorlage für diese Dateien: /usr/lib/pm-utils/default.
 - Die wichtigste Variable ist **SUSPEND MODULES.** = Liste der Module, die beim Suspend entladen werden sollen.
 - Umschalten zwischen den gewünschten Sleep-Modulen durch Setzen der Umgebungsvariablen **SLEEP_MODULE .** (Erst ab Version 1.1.0, nicht in Hardy.)

3.3. Energieverwaltung: Software-Schichtung



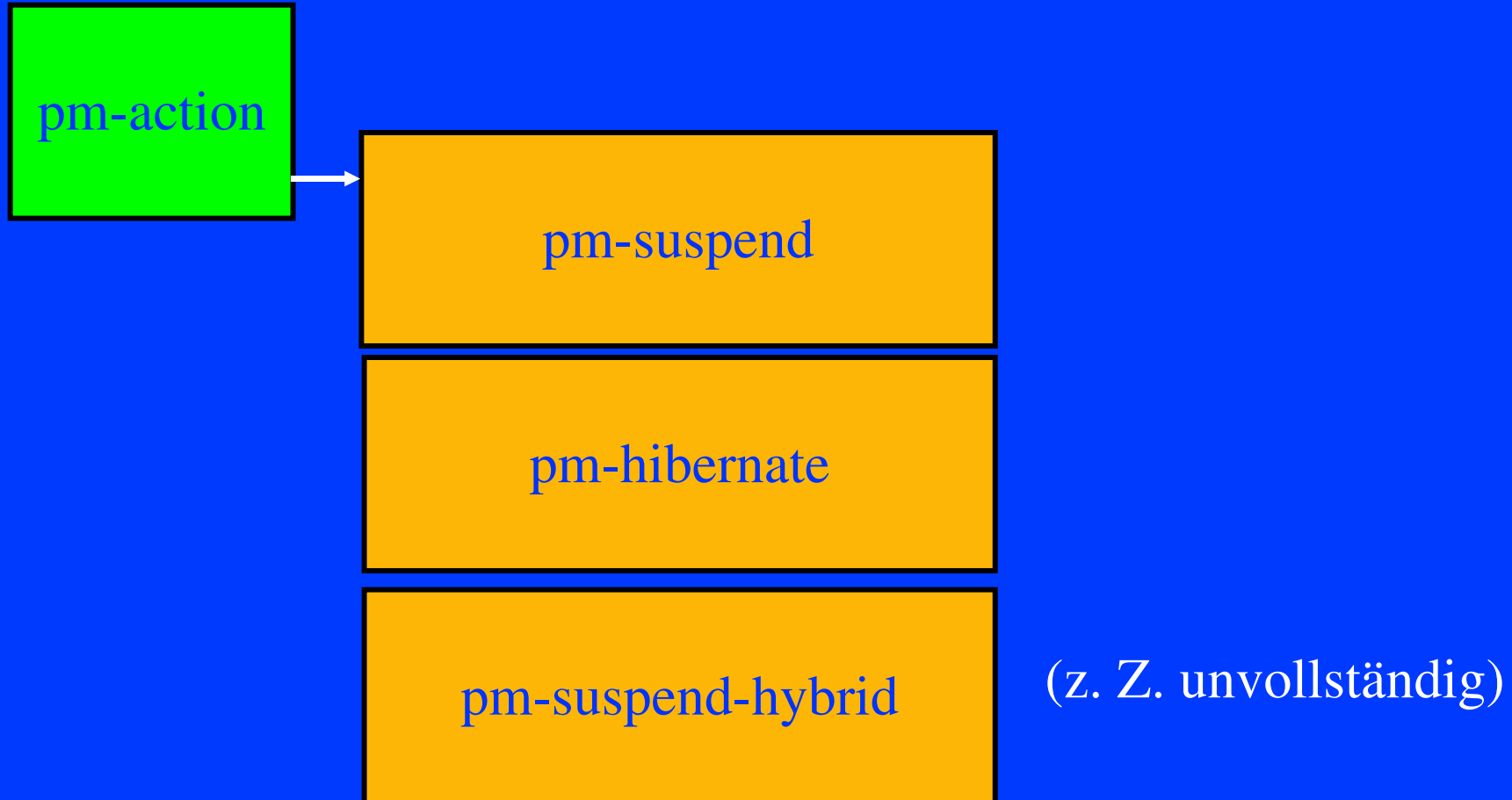
3.4. Überblick

- Die pm-utils werden von der HAL benutzt, um verschiedene Flicker (“hacks”) abzuarbeiten, um die Fehler (“bugs”) in den Treibern und Teilsystemen zu umschiffen, die (noch) nicht “Suspend” können.
- Die pm-utils-Funktionalität erweitert man durch Hinzufügen von “custom hooks” in /etc/pm/sleep.d/ :
 - Entweder: Systemadministrator.
 - Oder: Teil eines Programmpakets. Wenn besondere Aufmerksamkeit nötig ist während eines System-Suspend oder eines Stromsparübergangs (power state transition).

4. pm-action

- Das Haupt-Skript ist 'pm-action'. `/usr/lib/pm-utils/bin/pm-action`
 - pm-action wird über Symlinks aufgerufen als:
 - 'pm-suspend',
 - 'pm-hibernate' oder
 - 'pm-suspend-hybrid'. (Bild 2)
- pm-action führt die sog. "Hooks" aus.
- Wenn alle "Hooks" ausgeführt worden sind, versetzt pm-action den Computer in den Schlaf.
- Wenn der Computer aufgeweckt wird, werden alle diese "Hooks" in der umgekehrten Reihenfolge ausgeführt .
- 'pm-suspend-hybrid' ist z. Z. noch ein Platzhalter. Es ist nicht vollständig implementiert.

Aufrufsmöglichkeiten von pm-action



5.1. Ablauf

- Platzieren eine ausführbare Datei in `/etc/pm/sleep.d/` . Wenn Suspend oder Hibernate aufgerufen wird:
- Die Dateien in `/etc/pm/config.d/*` werden in alphabetischer Reihenfolge ausgewertet (Diese Dateien stammen meist von einzelnen Programmpaketen außerhalb der pm-utils und enthalten globale Konfigurationseinstellungen für pm-utils und die Hooks.) mit dem Kommandozeilen-Parameter
- **'suspend'** (= suspend to RAM) bzw.
- **'hibernate'** (= suspend to disk) .
- Das Betriebssystem geht in den Schlafzustand.
- Wenn ein Ereignis den Computer aufweckt:
- Die Dateien in `/etc/pm/config.d/*` werden beim Aufwecken ausgeführt mit dem Parameter
- **'resume'** (= resume from RAM) bzw.

5.1. Hooks

- Die Hooks bügeln die "quirks" der betreffenden Distribution aus. -- **Beispiele:**
- Den Urlader vorbereiten.
- Das Bluetooth-Subsystem anhalten.
- Kritische Module entladen.
- Kontroll-LEDs im Laptopcomputer ein- und ausschalten.
- Suspend-GUIs freigeben (wie früher suspend2).
- Video wieder einschalten.
- Dienste stoppen und starten, die Suspend nicht vertragen.
- Die Uhr mittels ntp neu synchronisieren.
- Module bei Bedarf modproben.
- Grub zum Default-Urlader nach dem Aufwecken aus dem Hibernate-Zustand machen.
- weitere Flickschustereien auf spezifischen Systemen.

5.2. Eigene Hooks

- **Hooks selbst schreiben:**
- Anpassungslösung für den eigenen Computer beim Suspend oder Hibernate: Eigene Hooks in das Verzeichnis `/etc/pm/sleep.d/` stellen.
- Die Hooks in `/etc/pm/sleep.d/` werden in alphabetischer Reihenfolge während Suspend und in umgekehrter Reihenfolge während Resumes ausgeführt.
- (Deshalb beginnen alle Hook-Namen mit einer 2stelligen Zahl.)
- **Beispiel:** Ein Hook, der nur je eine Meldung in die Logdatei bei Suspend bzw. Resume schreibt:
`/etc/pm/sleep.d/66dummy`.
- `chmod +x /etc/pm/sleep.d/66dummy`

5.3. Hook '66dummy'

- `#!/bin/bash`
- `case $1 in`
- `hibernate)`
- `echo "Hey honey, we are going to suspend to disk!"`
- `;;`
- `suspend)`
- `echo "Oh, this time we're doing a suspend to RAM. Cool!"`
- `;;`
- `thaw)`
- `echo "oh, suspend to disk is over, we are resuming..."`
- `;;`
- `resume)`
- `echo "hey, the suspend to RAM seems to be over..."`
- `;;`
- `*) echo "somebody is calling me totally wrong."`
- `;;`
- `esac`

6. Fehlersuche (1)

- `~$ lshal | grep power_management`
- **Logdatei**
 - Wenn Suspend oder Hibernate nicht korrekt mit den pm-utils arbeiten: Logdatei `/var/log/pm-suspend.log`
 - Welche „Hooks“ wurden abgearbeitet?
Welche Fehlermeldungen haben sie erzeugt?

6. Fehlersuche (2)

- **Einen Hook abschalten**
 - (Fehlerbericht erbeten, falls Hook mäßig brauchbar, unbrauchbar oder gar schädlich.)
 - **Hook selbst abschalten:** Eine Datei in `/etc/pm/config.d` erzeugen. In sie die Zeile `HOOK_BLACKLIST="hookname 99another-hook"` schreiben.
 - (HOOK_BLACKLIST ist eine Environment-Variable.)

6. Fehlersuche (3)

- **pm-suspend, pm-hibernate:**
 - Manuell ausführen in einem Terminal: 'sudo pm-suspend' bzw. 'sudo pm-hibernate' .
 - Eventuell erhält man im Terminal Ausgaben, die auf die Fehlerursache führen.
 - Die Fehlermeldungen der Suspend-Skripte in `/var/log/pm-suspend.log` analysieren.

6. Fehlersuche (4)

- **Umgebungsvariablen**

ADD_PARAMETERS, DROP_PARAMETERS

- a.) Fehler aufspüren, die durch die (fehlende) Übergabe eines Parameters an pm-utils auftreten, oder
- b.) Debuggen von Suspend/Resume allgemein:
 - Eine Datei in /etc/pm/config.d erzeugen mit den Zeilen:
 - **ADD_PARAMETERS="--parameter-to-add --<another-parameter>"**
 - **DROP_PARAMETERS="--parameter-to-drop --<another-parameter-to-drop>"**

6. Fehlersuche (5)

- **Weitere Umgebungsvariablen nutzen:**
 - **Alle Parameter ausschalten** (Testzwecke oder HAL-Fehler flicken):
 - **DROP_PARAMETERS="all"**
 - Ermittlen, ob ein **Kernelmodul** Suspend oder Resume verhindert: Die Umgebungsvariable **SUSPEND_MODULES** entfernt dieses Modul vor dem Suspend und lädt es wieder, wenn das System aufwacht.

6. Fehlersuche (6)

- **Weitere Parameterübergaben:**
 - Welche Parameter kann man an pm-suspend und Verwandte übergeben?
 - Ausführen: 'sudo pm-suspend **–help** –<weitere Parameter>'
 - Die Ausgabe enthält die Optionen, die das Kommando unterstützt, und zusätzlich, welche Hooks oder Module diese Optionen akzeptieren.

7. Energieverwaltung

- (Das Applet rechts im Benachrichtigungsfeld.)

—

—



- Standby (S3) engl. Suspend
- Ruhezustand (S4) engl. Hibernate
- Gutsy: Netzwerk deaktivieren und aktivieren, Irssi

24

7. Energieverwaltung

- (Das Applet im rechts im Benachrichtigungsfeld.)



8. Dialog „Beenden“

- **GNOME-Menü** System > Beenden...
 - Sommerschule (?)
- `~/Linux/GNOME/gnome-session-2.20.1/gnome-session/logout.c`
- **Bereitschaft** - **Sleep**
- **Ruhezustand** - **Hibernate**

Exit



Log Out



Restart



Shut Down

Pause



Lock Screen



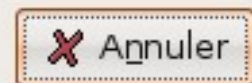
Switch User



Sleep



Hibernate



bye-bye

- **Ubuntu-Berlin findet man im IRC im Netzwerk „Freenode“ im Kanal #ubuntu-berlin.**

