

# Intelligent Platform Management Interface

## Systemmonitoring unter Linux (SoSe 2010)

Timme Katz

Betreuer: Timo Minartz

Fachbereich Informatik Universität Hamburg

29. Juni 2010



Universität Hamburg

# Inhaltsverzeichnis

## ① Einleitung und Motivation

## ② IPMI

- Grundlagen

- Architektur

- Monitoring

- Remote Management

## ③ Praktischer Einsatz

- Beispiele

- Live Demo

## ④ Fazit

# Warum Systemmonitoring?

- Problembehandlung
  - Probleme beheben bevor sie auftreten. (Proaktiv)
  - Probleme nach dem Auftreten analysieren.
- Sammeln von Informationen
  - Leistungsanalyse
  - Stromverbrauch
- Remotemanagement
  - Rechner Ein-/ausschalten und Neustarten

## Was ist IPMI?

- Ist eine Spezifikation
  - Erkennung und Verwaltung von Sensoren
  - Powercontrol
  - Watchdogs (Automatischer Systemreset im Fehlerfall)
- Funktioniert unabhängig von BIOS und Betriebssystem.
- Umfasst verschiedene Bereiche der Systemverwaltung.
- Benötigt spezielle Hardwareunterstützung.

- Ist eine Spezifikation
  - Erkennung und Verwaltung von Sensoren
  - Powercontrol
  - Watchdogs (Automatischer Systemreset im Fehlerfall)
- Funktioniert unabhängig von BIOS und Betriebssystem.
- Umfasst verschiedene Bereiche der Systemverwaltung.
- Benötigt spezielle Hardwareunterstützung.

1. Selbst wenn der Rechner ausgeschaltet ist und alle CPUs defekt sind, ist ein Zugriff per IPMI möglich. (Mainboard darf nicht defekt sein, Rechner muss an Strom angeschlossen sein.)

# Gründung

- 1998 durch (Intel, HP, NEC, Dell)
- Versionen:
  - v1.0 (16.09.1998)
  - v1.5 (01.03.2001)
  - v2.0 (14.02.2004)

- 1998 durch (Intel, HP, NEC, Dell)
- Versionen:
  - v1.0 (16.09.1998)
  - v1.5 (01.03.2001)
  - v2.0 (14.02.2004)

1. Neu in v1.5: Serial/LAN Messaging and Alerting, Zusätzliche Sensor und Event Typen, Platform Event Filtering
2. Neu in v2.0: Verbesserte Authentifizierung, VLAN, SOL, Payloads, Encryption, Firmware Firewall

# IPMI Features

I P M I	Monitoring
	Recovery
	Logging
	Alerting
	Inventory



- ① Management Controller
  - BMC
  - Satellite Controller
- ② Kommunikation
  - Lokales System Interface
  - Seriell
  - LAN
- ③ Informationsspeicher
  - Sensor Data Record (SDR)
  - Field Replaceable Unit (FRU)
  - System Event Log (SEL)

# Schema

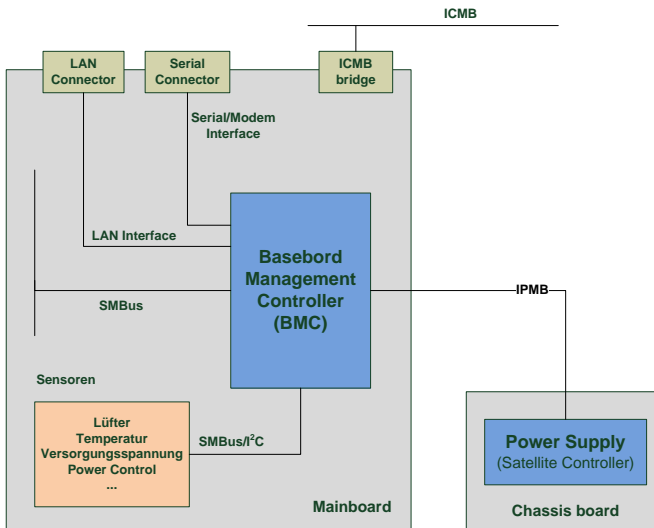


Abbildung: Eigene Zeichnung nach [IHND09]

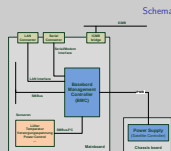
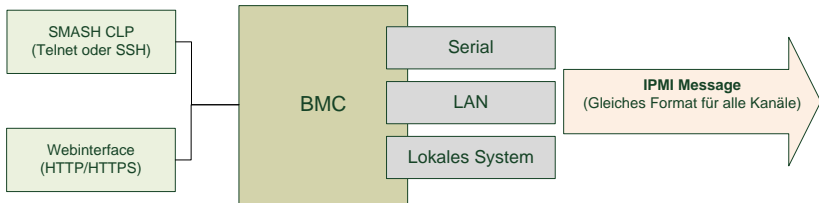
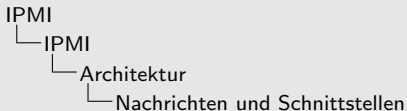


Abbildung: Eigene Zeichnung nach [HN00]

- BMC
  - Herzstück und Intelligenz von IPMI
  - Brücke zwischen System und Platform Management
- MC
  - Weitere Management Controller können über IPMB angebunden werden.
  - MCs sind meistens nicht auf dem Basebord integriert (z.B. Power Supply)
  - Anbindung über IPMB → Messaging

# Nachrichten und Schnittstellen





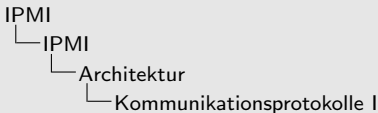
1. SMASH = Systems Management Architecture for Server Hardware.
2. SMASH wird von einer DMTF TaskForce entwickelt um server management einheitlicher zu gestalten.
3. Zusätzlich auch tw. noch ein Webinterface zur Konfiguration des BMC vorhanden.

# Lokales System Interface

- Lokales System Interface
  - Unter Linux über einen Kernelmodul realisiert.
  - Zugriff über das Device */dev/ipmi0*
- Vorteile
  - Per *ipmitool* kann der BMC konfiguriert werden.
  - z.B. Netzwerkeinstellungen und Benutzer.

# Kommunikationsprotokolle I

- $I^2C$ 
  - Einfacher, weit verbreiteter Bus.
  - Nur 2-Drähte werden benötigt.
  - Geringe Bandbreite.
- SMBus
  - Untermenge von  $I^2C$ .
  - Ziel: Robustheit und Interoperabilität.
  - Striktere Anforderungen an das Protokoll und die elektrischen Parameter.
  - SMBus fordert das senden von ACK/NACK Antworten.
  - SMBus unterstützt Packet Error Checking.



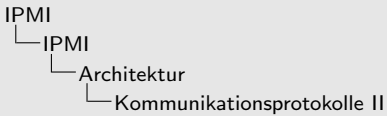
- I<sup>2</sup>C
  - Einfacher, weit verbreiteter Bus.
  - Nur 2-Dröhte werden benötigt.
  - Geringe Bandbreite.
- SMBus
  - Untermenge von I<sup>2</sup>C.
  - Ziel: Robustheit und Interoperabilität.
  - Stärkere Anforderungen an das Protokoll und die elektrischen Parameter.
  - SMBus fordert das senden von ACK/NACK Antworten.
  - SMBus unterstützt Packet Error Checking.

1. I<sup>2</sup>C ist gut geeignet für Fälle in denen es nicht auf Geschwindigkeit ankommt und die Produktionskosten niedrig sein müssen.
2. Beispiele: Konfiguration aus SPD EEPROMS auslesen (s.a. FRU), Monitor Einstellungen (DDC), Ein-/Ausschalten des Netzteils.
3. SMBus oft auf Mainboards anzutreffen. Kommunikation mit Netzteil.
4. Unterschiede ACK/NACK handling. Bei I<sup>2</sup>C können Geräte das ACK/NACK auslassen, wenn sie gerade Echtzeit Operationen ausführen. Bei SMBus muss geantwortet werden.
5. SMBus hat ein Timeout Feature, wenn eine Operation länger als 35ms dauert, wird der Bus zurückgesetzt.
6. SMBus unterstützt Packet Error Checking



## Kommunikationsprotokolle II

- Intelligent Platform Management Bus (IPMB)
  - $I^2C$  basierter serieller Bus
  - Verbindung von MCs und BMC
  - Abfrage von Sensoren
  - Kommunikation innerhalb eines Chassis
- Intelligent Chassis Management Bus (ICMB)
  - BMC kann als ICMB Bridge dienen.
  - Kommunikation zwischen verschiedenen Chassis.



- Intelligent Platform Management Bus (IPMB)
  - I<sup>2</sup>C basierter serieller Bus
  - Verbindung von BMCs und BMC
  - Abfrage von Sensoren
  - Kommunikation innerhalb eines Chassis
- Intelligent Chassis Management Bus (ICMB)
  - BMC kann als ICMB Bridge dienen.
  - Kommunikation zwischen verschiedenen Chassis.

1. ICMB: Beispiel: externes Bandlaufwerk.
2. ICMB Bus Länge max 300feet (100m)

# Informationen über Sensoren

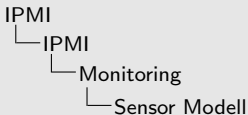
- Capabilities Command
  - Standard IPMI Kommandos.
  - Abfrage der Fähigkeiten des BMC.
- Sensor Data Records
  - Typ und Anzahl von Sensoren
  - Sensor Schwellwerte
  - Event Erzeugung (s.a. Schwellwerte)
  - Typ der Sensor Daten (Linear/Non-Linear)

# Abfrage der Sensordaten

- Abfrage der Sensoren erfolgt über:
  - IPMB
  - SMBus
  - $I^2C$  (z.B. Abfrage der FRU Infos aus dem EEPROM)

# Sensor Modell

- Kein direkter Zugriff auf die Sensor Hardware.
- Beschreibung der Sensoren über Sensor Data Records
- Kategorisierung von Sensoren
  - Daten: Linear / Non-Linear
  - Events: Schwellwert / diskret



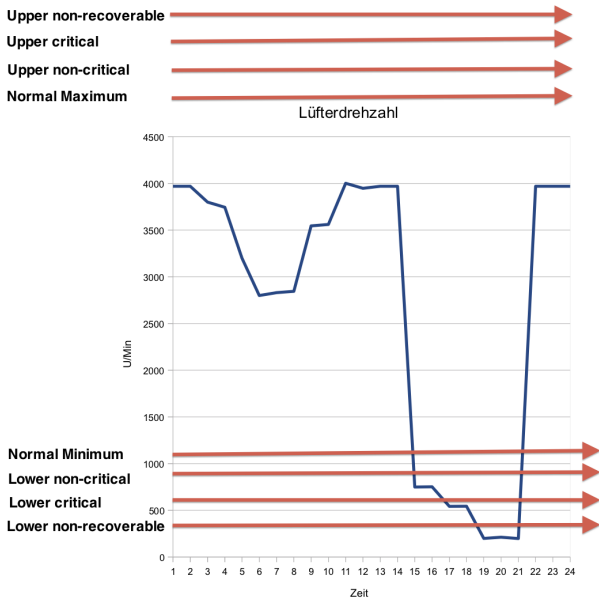
- Kein direkter Zugriff auf die Sensor Hardware.
- Beschreibung der Sensoren über Sensor Data Records
- Kategorisierung von Sensoren
  - Daten: Linear / Non-Linear
  - Events: Schwellwert / diskret

1. Linear: konstanter Umrechnungsfaktor für die Sensor RAW Werte vorhanden.
2. Non-Linear: der Umrechnungsfaktor hängt vom Zeitpunkt der Messung ab.

## Welche Sensoren gibt es?

- Versorgungsspannung (vom Netzteil)
- Netzteilstatus
- CPU (Temperatur, Spannung)
- Temperatur
- Lüfter (Anwesenheit, Umdrehung)
- Speicher (Versorgungsspannung)
- Festplatten (Anwesenheit)
- ...

# Sensor Grenzwerte Beispiel Lüfter





## Was Sensoren noch können...

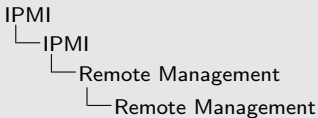
- Entity Association Records
  - Verbindung und Gruppierung einzelner Sensoren.
  - Mehrere Netzteile können zu einer redundanten Einheit verbunden werden.
- Field Replaceable Unit (Ersatzteilnummer)
- Verbindung von Events und FRU Informationen
  - Verbindung von SDR und FRU über Codes im System Event Log.
  - Vorteil: Servicetechniker kann sofort das richtige Ersatzteil mitbringen.

# Remote Management

## Power Control (Betriebssystem unabhängig)

Power Up	Einschalten
Power Down	soft off S4/S5 power state
Power Cycle*	Neustarten wenn Rechner läuft.
Hard Reset	Neustarten in jedem Fall.
Soft Shutdown*	Herunterfahren über ACPI

\*: optional



Power Control (Betriebssystem unabhängig)

Power Up	Einschalten
Power Down	soft off S4/S5 power state
Power Cycle*	Neustarten wenn Rechner lauff.
Hard Reset	Neustarten in jedem Fall
Soft Shutdown*	Herunterfahren über ACPI

\* optional

1. Power Down: S4=Hibernate, S5= Soft-Off-Modus. Kein kontrolliertes Herunterfahren des OS!
2. Power Cycle startet den Rechner Neu. Wenn der Rechner aus ist, sollte nichts passieren.
3. Soft Shutdown: Führt das System über Simulation eines fatal overtemperature herunter.

# Netzwerkzugriff

- IPMI over LAN (seit v1.5)
  - Versand der Daten über RMCP
  - RMCP basiert auf UDP
- Neuerungen in v 2.0
  - RMCP+
  - Übertragung zusätzlicher Payload mit IPMI Messages.
  - Authentizität und Vertraulichkeit der Nachrichten gewährleistet.

# Serial over LAN

## Bootloader anpassen

```
serial -unit=1 --speed=19200 -word=8 -parity=no -stop  
terminal -timeout=5 serial console
```

[...]

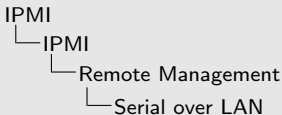
```
kernel /boot/vmlinuz root=/dev/sda1 ro rhgb  
console=tty0 console=ttyS1,19200n8r
```

## Linux Konsole auf serieller Schnittstelle

```
s0:2345:respawn:/sbin/agetty 19200 ttyS0 vt100-nav  
s1:2345:respawn:/sbin/agetty 19200 ttyS1 vt100-nav
```

## ipmitool

```
ipmitool -I lanplus -H intel-ipmi -U admin sol activate
```

[Bootloader anpassen](#)

```
serial --unit=1 --speed=19200 --word=8 --parity=no --stop
terminal --simout=5 serial console
[...]
```

[Linux Konsole auf serieller Schnittstelle](#)

```
u0:2345:respawn:/sbin/agetty 19200 ttyS0 vt100-nov
s1:2345:respawn:/sbin/agetty 19200 ttyS1 vt100-nov
```

[ipmitool](#)

```
ipmitool -I lanplus -H intel-ipmi -U admin sasl activate
```

1. Grundsätzlich müssen drei Dinge angepasst werden: BIOS, Bootloader und Betriebssystem
2. GRUB: `--unit` ist die Nummer der seriellen Schnittstelle. `0 = ttyS0 = COM1`
3. GRUB: serial console: Ausgabe auf Terminal und serieller Schnittstelle

# Nagios IPMI Plugin

- Nagios Netzwerküberwachung
  - Host-checks
  - Service-checks
  - Über Plugins beliebig erweiterbar.
- IPMI Plugin
  - Lokales System Interface oder LAN
  - Beschränkung auf einzelne Sensortypen möglich. (z.B. FAN)

## Proactive Fault Tolerance for HPC

- Parallelrechner Umgebung.
- Virtualisierung mit XEN.
- IPMI zum Systemmonitoring.
- Migration der VM von potentiell fehlerhaften auf gesunde Nodes.

Quelle: [NMES07]



Live Demo...

- Spezifikation für einheitliches Systemmonitoring.
- Unterstützt Monitoring, Recovery, Logging, Alerting und Inventory
- Vorteile:
  - Herstellerübergreifend.
  - Betriebssystem unabhängig.
  - Remotemanagement fähig.
- Nachteile:
  - Benötigt extra Hardware unterstützung.
  - Ist meist nur im Serverbereich anzutreffen.

## Literaturverzeichnis I



FISCHER, Walter:

*Thomas Krenn IPMI Wiki.*

<http://www.thomas-krenn.com/de/wiki/Kategorie:IPMI>, 06  
2010



INTEL ; HEWLETT-PACKARD ; NEC ; DELL:

*Intelligent Platform Management Interface Specification  
Second Generation.*

06 2009



NAGARAJAN, Arun B. ; MUELLER, Frank ; ENGELMANN,  
Christian ; SCOTT, Stephen L.:

*Proactive Fault Tolerance for HPC with Xen Virtualization.*  
ISC 2007, 06 2007

## Literaturverzeichnis II



NXP:

*I2C-bus specification and user manual (Rev. 03).*

06 2007



WIKIPEDIA:

*System Management Bus.*

06 2010