# Strategien zur optimierten Speichernutzung komplexer Systeme

Prof. Dr. Heusch
Hochschule für Technik
Stuttgart

Jürgen Groß
Fujitsu Technology Solutions
München

# Motivation

FUJITSU

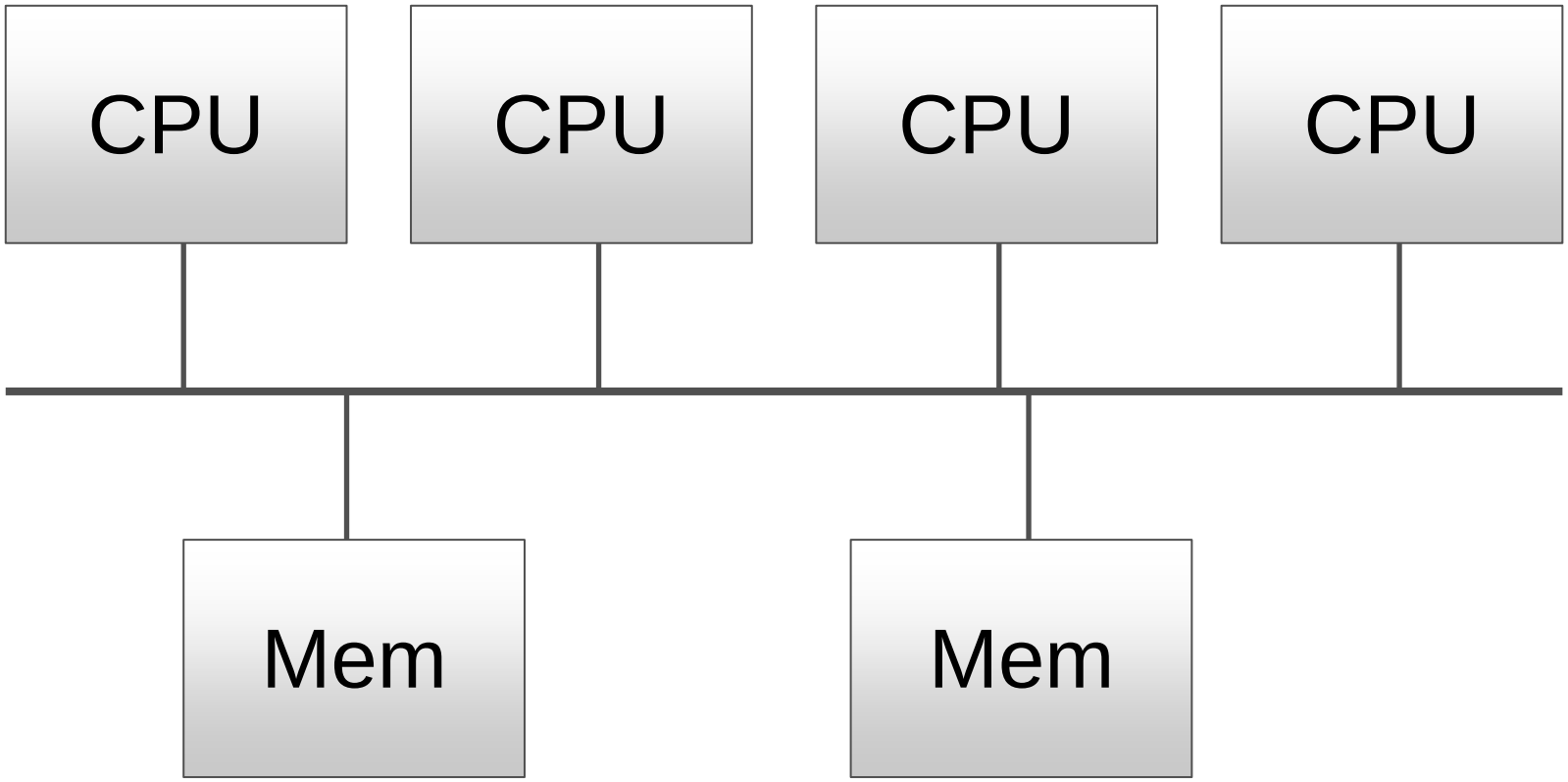
- Komplexe Systeme: Multicore NUMA Systeme
- Beispiel: SQ200 (Mainframe auf x86-Basis mit BS2000 und anderen Gästen)
- Problematik: hohe Anlagenauslastung bei gleichzeitig garantierter BS2000-Performance

**Optimierung:** Speicherzugriff-Verhalten der SW an Systemarchitektur anpassen

The inherent vice of capitalism is the unequal sharing of blessings. The inherent virtue of Socialism is the equal sharing of miseries.

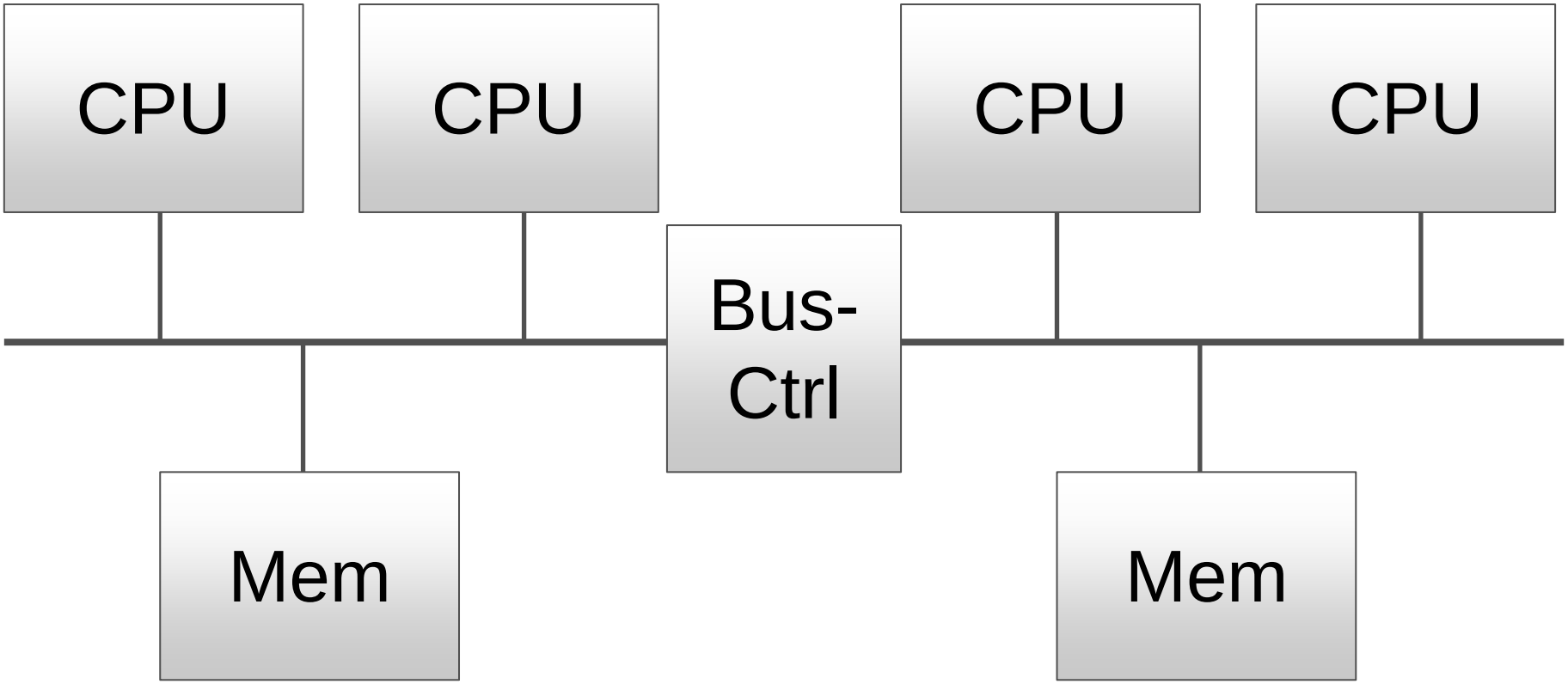
Winston Churchill, 1945

# NUMA-Architekturen: kein NUMA




- Alle Zugriffe gleich schnell
- nur 1 Zugriff auf einmal
- Gesamtdurchsatz durch Bus bestimmt

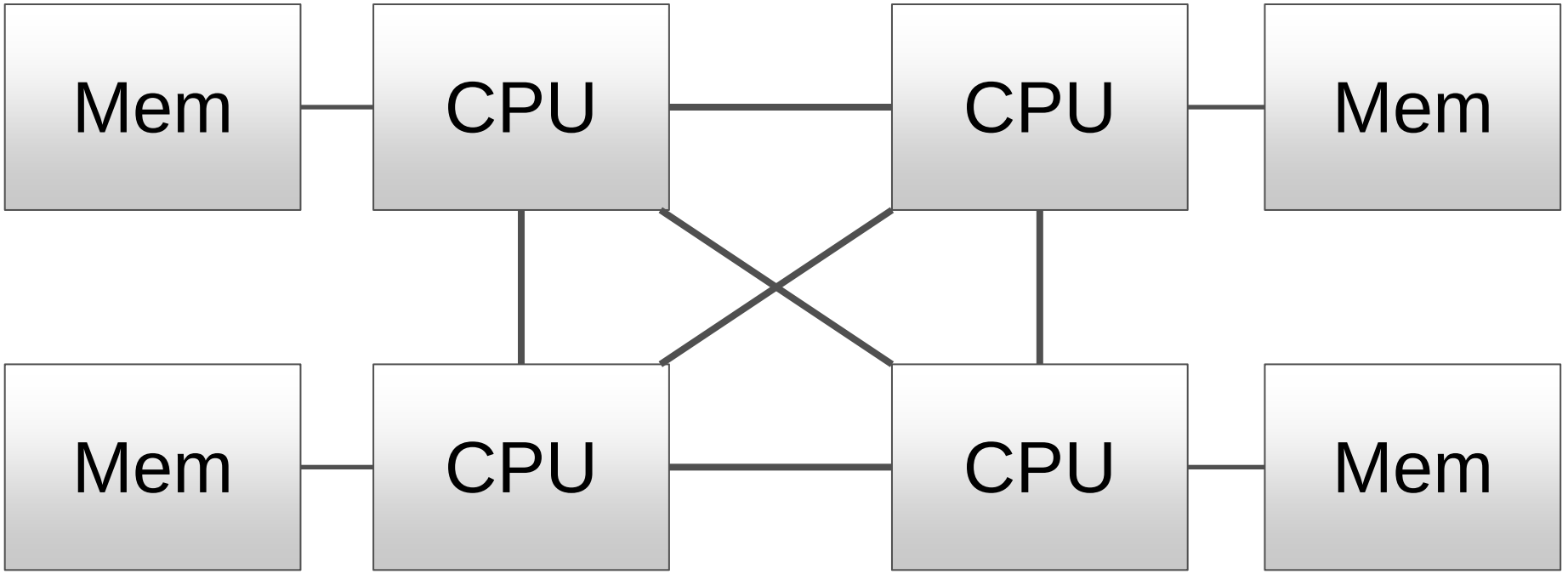
# NUMA-Architekturen: Bus-basiert




- lokale Zugriffe schneller als remote
- 1 Zugriff pro Bus auf einmal
- Gesamtdurchsatz durch Lokalität und Anzahl der Busse und Auslastung der Busse bestimmt

# NUMA-Architekturen: Link-basiert

FUJITSU



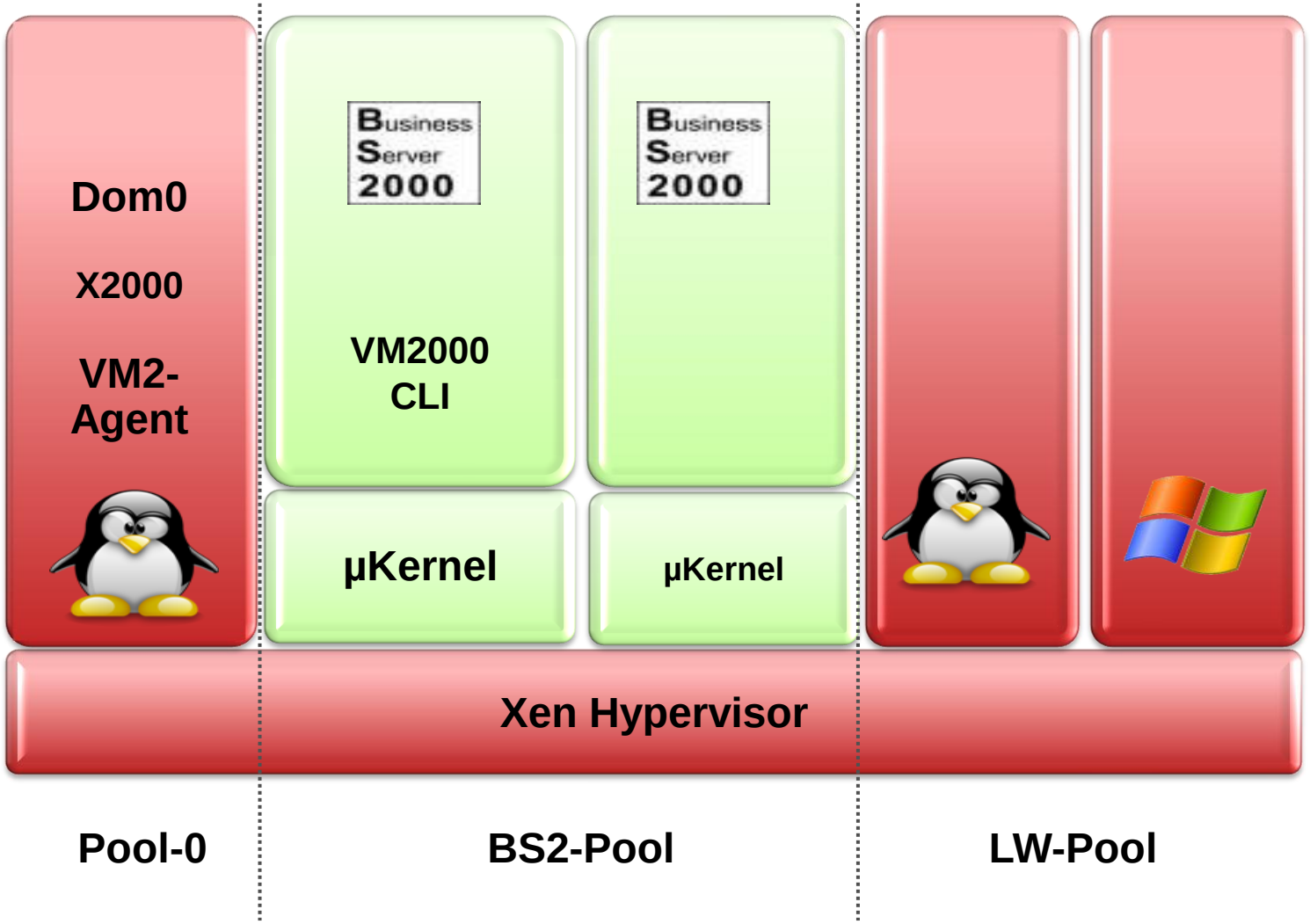

- lokale Zugriffe schneller als remote
- 1 Zugriff pro Link auf einmal (mehrere pro cpu)
- Gesamtdurchsatz durch Lokalität und Anzahl der Links und Auslastung der Links bestimmt

# SQ200 Grundlagen

FUJITSU

- HW
  - x86 Nehalem-Prozessor (NUMA Link-basiert)
  - 4 Sockets à 6 Cores
  - Speicher mit Mirror-Mode (Intra-Socket)
- SW
  - XEN Hypervisor (erweitert um Cpupools)
  - Domain 0: SLES 11 SP1
  - DomU: BS2000, Linux, Windows
- NUMA Optimierungen
  - XEN Cpupools möglichst Socket-lokal
  - XEN Scheduler versucht, vCPUS auf einem Socket zu halten
  - Physikalischer Speicher der DomUs bevorzugt am lokalen Socket

# SQ200 Grundlagen



# Hintergrund

FUJITSU

- Auf der Intel-Plattform läuft BS 2000 als DomU unterhalb von Xen
- Softwarekosten richten sich (auch) nach der Leistungsfähigkeit der Hardware:
  - Der Kunde kauft eine bestimmte Hardware, und möchte deren Leistungsfähigkeit garantiert haben
  - Der Hersteller verkauft eine bestimmte Hardware und möchte deren Leistungsfähigkeit garantieren können
- Um die Garantie abgeben zu können, müssen Sicherheitsreserven einkalkuliert werden

# Ressourcenzuordnung

FUJITSU

- Im Bereich der CPUs wird die Performance unter Xen durch CPU-Pools garantiert
- Und der Rest (Hauptspeicher, Massenspeicher, Kommunikationskanäle)?
- Die Technik bringt uns eine inhärente Einteilung:
  - Massenspeicher und Kommunikationskanäle sind „an die richtigen“ CPUs anzuschließen, der Einfluss des Betriebssystems ist gering
  - Beim Hauptspeicher kann das Betriebssystem sehr viel mehr optimieren

# Hauptspeicher-Zuordnung

FUJITSU

- Bei hochparallelen Systemen ist der Speicher pro Sockel i.d.R. begrenzt
- Einem CPU-Pool kann also eventuell nicht genügend lokaler Speicher zugeordnet werden
- Idee: Ein CPU-Pool „klaut“ den Speicher eines anderen direkt benachtbarten CPU-Pools
- Alternative: Ersetze CPUs durch Dummies, die nicht rechnen können und keine Gebühren für Lizenzen verursachen

# Mögliche Ansätze

FUJITSU

- Speicher einer Domäne wird gleichmäßig über die Hardware verteilt:
  - Vorteil: optimale Ausnutzung der Hardware
  - Nachteil: erhöhter Kommunikationsaufwand und gleichmäßige Verteilung der Engpässe
- Speicher einer Domäne liegt möglichst auf oder nah bei seinen CPUs:
  - Vorteil: geringer Kommunikationsaufwand und kleine Gefahr externer Einflüsse
  - Nachteil: lokale Hotspots können zu Engpässen führen

# Virtualisierung

FUJITSU

- Virtualisierung macht aus der Menage a deux eine Menage a troi
  - Der Hypervisor muss dem Gast-System Speicher passend zu den vorgesehenen CPUs zuteilen
  - Im Gast-System muss der Scheduler die Threads passend auf den CPUs ausführen
- Wer darf bzw. muß wem Vorgaben machen?
- Was passiert bei einer Rekonfiguration des Systems zur Laufzeit? Im schlimmsten Falle wird der CPU-Pool und damit der Speicher „zerrissen“

# Dynamische Rekonfiguration

FUJITSU

- Hypervisor muss dem Gast-OS mitteilen, wenn dynamisch Rekonfigurationen vorgenommen wurden, Gast-System muss sich anpassen können
- Gast-System muss dem Hypervisor einen Wunschzettel übergeben können:
  - Wird Speicher vom Hypervisor allokiert, braucht der Hypervisor eine Ortsvorgabe
  - Grundsätzlich sollte der Gast dem Hypervisor eine „optimale“ Konfiguration anzeigen können

# Umsetzung

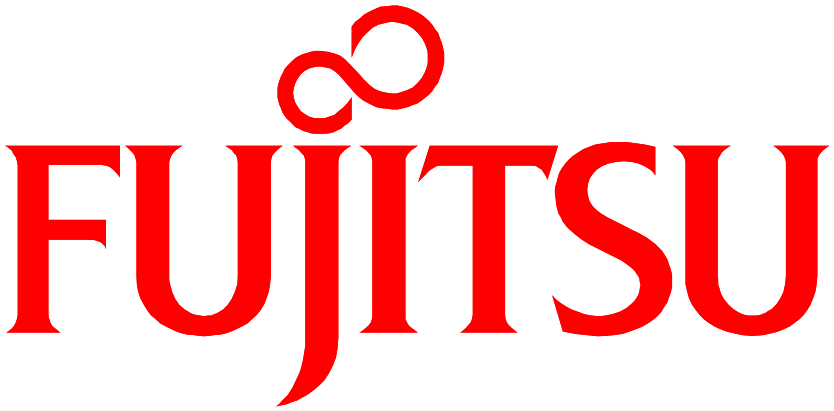
FUJITSU

- Xen-Hypervisor ist um entsprechende APIs zu erweitern:
  - Speicher-Allokation mit CPU-Angabe
  - Speicher-Diebstahl aus benachbarten CPU-Pools
  - Absetzen von Wunschzetteln mit Bezug auf die Speicherkonfiguration
- Gast-System muss periodisch ACPI-Tabellen (insb. SRAT, SLIT) prüfen und sein Scheduling anpassen

# Zusammenfassung

FUJITSU

- Die prinzipiellen Probleme sind klar, der Teufel steckt wie immer im Detail
- Was passiert, wenn die Scheduler von Gast-System und Hypervisor aufeinander treffen
- Wieviel Leistungssteigerung lässt sich durch die Optimierung erreichen?
- Unter Solaris wurde durch NUMA-Optimierung 4-8% erreicht


FUJITSU