

## "Ressourcenmanagement im kommerziellen Rechenzentrum"

Diskussion und Erarbeitung gemeinsamer Empfehlungen zur Ressourcensteuerung von CPU, Hauptspeicher und IO Bandbreite.

# *Erwartungen an das Ressourcenmanagement*

- Einhaltung von Service Levels (minimaler Service & maximaler Service)
- Applikationen von anderen abgrenzen
- Ausschluss von Ressourcenübergriffen
- Änderungen *on-the-fly* durchführen
- Einsparung von ressourcenbezogenen Lizenzen
- Vorhersagbare Performance
- Standardisierung der Applikationen
- Dynamischer Lastausgleich

# Technische Möglichkeiten in Solaris 10

## CPU

- Einsatz des Fairshare Shedulers
- Zuweisen von Prozessorsets
- Nutzen von Resource Pools

## Memory

- Ressource Capping
- Ressource Controls
- Ressource Pools
- Projekte

## Bandbreite

- Flowcontrole
- IP QoS
- IO Limit
- IO Share

## IPC

- Anzahl Semaphoren und Shared Memmory
- Ressource Controls
- Projekte

## ***Welche Probleme müssen gelöst werden***

- Strikte vs. dynamische Ressourcenzuordnung
- Der Administrationsaufwand soll minimiert werden
- Teilweise nur mit bestimmten Patchständen / Versionen nutzbar
- Monitoring und Erkennen von Ressourcenmangel
- Wie erfolgt die Integrationen in einen heterogenen Cluster
- Ressourcenmanagement sowohl innerhalb von Zonen als auch für einzelne Applikationen

**Viele Stellschrauben -> woran dreht man am besten**

**???**

# Ressourcenmanagement – mögliche Lösungen

- Wie garantiere ich CPU Zeit – beschränke aber trotzdem die Anwendung?
  - Wenn aus Lizenzgründen nur eine bestimmte Anzahl an CPUs sichtbar sein soll:
    - *Prozessorsets limitieren die sichtbaren Prozessoren in Zonen*
  - Wenn sich die Anzahl der genutzten Prozessoren aus der Last der Anwendung ergeben soll:
    - *Prozessorsets in Resourcepools erlauben ein dynamisches, lastabhängiges zuweisen von CPU Ressourcen*
    - *ACHTUNG: Anwendungen können evtl. zusätzliche Ressourcen nicht nutzen, da sie beim Start der Anwendung nicht vorhanden waren*
- Was ist zu beachten?
  - Prozessorsets sind statisch. Gebunden Prozessoren stehen anderen Anwendungen nicht zur Verfügung
  - Einsatz von Fair Share Scheduler benötigt clusterweite einheitliche Berechnungsgrundlage
  - FSS muss für die Nutzung auf dem Knoten aktiviert sein

# *Ressourcenmanagement – mögliche Lösungen*

- Projekte für Applikationen (Clusterprojekte)
- Prozesse oder Tasks die in der Applikation gestartet werden gehören zu diesem Projekt
  - Einfache Handhabung von Resource Controls
  - Ressourcen können dynamisch beim Start festgelegt werden (Swap, Memory, IPC Parameter)
  - Ressourcen können für das Projekt dynamisch geändert werden
  - Einfache Erkennung von IPC Segmenten zum Projekt

# *Monitoring von Applikationen*

- Überschreitungen von Resource Controls können in der messages geloggt werden
  - Auswertung muss über externe Tools oder logging Host erfolgen
- Monitoring von Applikationen im Cluster
  - Überwachung von IP Traffic und Steuerung mit IP QoS
  - dynamisches Anpassen von IO Limits / IO Shares
  - Auswertung der messages – Anpassung von Ressourcen

# *Ressourcenmanagement – welche Probleme bleiben ?*

- Ressourcenmanagement muss für alle Applikationen durchgeführt werden
- Möglichst einheitlichen Systemstand (Patchlevel / OS Version)
- Adminaufwand wird nicht einfacher
  - Mehre Möglichkeiten Fehler zu machen
  - Muss sinnvoll eingesetzt werden
  - Es kann schnell vorkommen das Applikationen nicht starten oder keine Ressourcen haben
  - Systemlast trotzdem hoch



## *Empfehlenswerte Literatur*

- Systemverwaltungshandbuch: Solaris Container – Ressourcenverwaltung und Solaris Zones
  - <http://docs.sun.com/app/docs/doc/820-2316?l=de>
- Solaris 10 Container Leitfaden - Funktionsweisen, Anwendungsfälle, Best Practices und Kochbücher
  - <http://de.sun.com/teleweb/virtualisierung/pdf/ptt-leitfaden-solaris-container-v2.0.pdf>
- Solaris Containers: Consolidating Servers And Applications With Solaris Containers
  - [https://www.sun.com/offers/details/solaris\\_containers.xml](https://www.sun.com/offers/details/solaris_containers.xml)