

Ressourcen- und Performance-Management für SAP auf Power-Systemen

Jochen Hein
Auf der Fittell 18
53347 Alfter
Deutschland
<jochen@jochen.org>

1 Abstract

Auf modernen Plattformen wie der IBM-Power-Serie können Ressourcen, wie CPUs, Hauptspeicher und IO-Ressourcen recht flexibel verwaltet werden. Unter welchen Bedingungen und für welche Ressourcen ist der Einsatz der Virtualisierung für Performance-intensive Systeme wie SAP R/3 sinnvoll bzw. notwendig? Welche Strategien sollte man bei der Installation von SAP Systemen verfolgen, um für die Zukunft möglichst flexibel zu bleiben?

Der Vortrag beleuchtet die Auswirkungen der Virtualisierung auf etablierte Monitoring-Systeme (innerhalb und außerhalb SAP). Welche Möglichkeiten gibt es, die Virtualisierung sinnvoll zu überwachen und zu erkennen, dass die eingestellten Parameter nicht (mehr) sinnvoll sind? Ein Beispiel für die übersichtliche Darstellung vieler, auch selbst definierter Performance-Daten wird das Monitoring-Tool Ganglia (<http://ganglia.info>) sein.

Der Vortrag wird von Erfahrungen innerhalb von großen SAP-Umgebungen berichten. Dabei werden sowohl technische als auch organisatorische Rahmenbedingungen und der Einsatz von kommerziellen und frei verfügbaren Monitoring-Tools beleuchten.

2 Ressourcen-Management für SAP-Systeme

Große SAP-Systeme stellen erhebliche Anforderungen an die verfügbaren Ressourcen. Die wichtigsten sind dabei CPU-Leistung, Hauptspeicher und Plattenplatz. Auch die Anforderungen an die Netzwerk-Anbindung (LAN und SAN) sind nicht zu verachten. Wie kann man eine große SAP-Landschaft so betreiben, dass stets die notwendigen Ressourcen verfügbar sind?

Dazu kommen die Regeln, die z.B. im Rahmen von Outsourcing-Verträgen für den Umgang mit Ressourcen definiert wurden. In unserer Umgebung wurde vereinbart, dass die CPU-Leistung der Systeme mit Hilfe von SAPSen gemessen werden. Dabei geben die Hardware-Hersteller für jeden Maschinentyp bzw. Ausbaustufe derselben eine gewisse Leistung für den SAP-Standard-Benchmark (SAPS) vor. Für unseren Kunden ist die Bereitstellung einer gewissen Anzahl SAPSe vereinbart. Mehraufwand könnte zu Nachforderungen führen, Minderaufwand zu Verrechnungen. Unabhängig von der kaufmännischen Bewertung der Systemleistung muss ein performanter Systembetrieb sichergestellt werden.

2.1 CPU-Ressourcen

In einer Power-Maschine sind – je nach Ausstattung eine bestimmte Anzahl CPUs eingebaut (bzw. bei Capacity-On-Demand aktiviert). Wenn man auf einer physischen Maschine genau ein Betriebssystem und eine SAP-Installation betreibt, so ist die Aufteilung statisch und nur durch Hardware-Umbau zu

verändern. Die Praxis zeigt jedoch, dass die Anforderungen der Systeme sich im Lauf der Zeit ändern (meist: wachsen). In bestimmten Projektphasen wird mehr Leistung benötigt, in anderen weniger.

Hier erwarten Kunden, dass Ressourcen flexibel und schnell verlagert werden können – ohne einen zeitaufwändigen Systemumzug¹.

Für eine LPAR wird eine sogenannte “entitled capacity” eingestellt. Dieser Wert gibt an, wie viel CPU-Leistung die LPAR in jedem Fall erhält, auch wenn alle LPARs des Rechners unter Volllast arbeiten. Benötigt aber eine LPAR weniger CPU-Leistung, so wird diese im CPU-Pool für andere LPARs verfügbar. 1.2Memory

Die Konfiguration eines SAP-Systems ist, zumindest was den minimalen, normalen und maximalen Speicherbedarf betrifft, statisch. Für die Anpassung der Größen von Puffern oder Arbeitsbereichen ist in der Regel ein Durchstarten des SAP-Systems erforderlich. Daher arbeiten wir mit einer statischen, nicht-virtualisierten Memory-Konfiguration. Eine Faustregel für uns ist, dass wir den gesamten Hauptspeicher im physischen Rechner durch die Anzahl der physischen CPUs teilen. Dieser Wert gibt an, wie viel Hauptspeicher wir einer LPAR zuordnen, die eine “entitled Capacity” von 1 hat.

Da ein SAP-System heutzutage einen gewissen Speicherbedarf hat, ist es nur selten sinnvoll, weniger als eine CPU zuzuordnen.

2.2 SAN-Storage

Auf unseren Systemen arbeiten wir mit dem Logical Volume Manager von AIX. Damit ist es uns möglich, sehr schnell und online zusätzlichen SAN-Storage anzuschließen, aber auch nicht mehr benötigten Storage wieder abzugeben (was leider nicht immer online geht und auch die Systemperformance beeinträchtigt).

In der Zukunft planen wir den Einsatz des SAN Volume Controllers der IBM. Dieser ermöglicht es, auch SAN-Storage zu virtualisieren. Als besonders nützlich werden sich vermutlich die folgenden Features erweisen:

- virtuelle SAN-Platten können aus verschiedenen Storage-Systemen zusammengesetzt werden und können dynamisch verändert werden. Der Host sieht dabei nur eine SAN-Platte, die durch den SAN Volume Controller bereit gestellt wird – die Physik dahinter ist nicht mehr relevant.
- Es ist möglich, im laufenden Betrieb alle Daten von einer Storage-Box weg zu verlagern. Das kann es uns – genügend Storage-Freiplatz vorausgesetzt ermöglichen, einen Firmware-Upgrade der Storage-Box ohne Downtime aller Systeme durchzuführen.
- Wir können Flash-Copys zwischen beliebigen Storage-Subsystemen durchführen, auch wenn diese Option dort nicht aktiviert ist. Damit war es uns in einem Kundenprojekt möglich, innerhalb von 20 Minuten eine SAP-Systemkopie auf einer Testmaschine durchzuführen, damit wir eine Fehlersituation des produktiven Systems nachstellen konnten. Dabei war der SAN-Storage auf zwei Storage-Subsystemen verteilt.

In unseren Testszenarien war kein Performance-Verlust nachzuweisen. Und die sehr viel größere Flexibilität wird uns das Leben sehr erleichtern können.

2.3 Netzwerk

Auf der Power-Plattform ist es möglich, auch die PCI-Karten zu virtualisieren (Virtual I/O). Dabei stellt eine LPAR, der die Karten zugeordnet sind, diese in virtualisierter Form den anderen LPARs zur Verfügung. Das ist notwendig, da eine LPAR in der Minimal-Konfiguration nur noch 0,1 CPUs haben muss – wenn für jede LPAR eine Netzwerk-Schnittstelle und ein SAN-Adapter notwendig wären, dann würde ein üblicher Hardware-Ausbau nicht mehr sinnvoll zu betreiben sein.

¹Wir planen unsere Systeme nur noch in sogenannte “virtuelle” Hostnamen zu installieren. Damit können wir mit sehr geringem Aufwand auf eine neue Hardware umziehen. Falls denn die Virtualisierung an ihre Grenzen stößt...

Wir setzen im SAP-Umfeld keine VIO-Server ein. Das liegt zum einen daran, dass wir auch auf einer voll ausgebauten P570 maximal acht LPARs konfigurieren und dort noch in der Lage sind mit physischen Schnittstellen zu arbeiten. Auch zeigt die Erfahrung, dass SAP-System große Anforderungen an die Systemperformance stellen – und da ist Physik einfach nicht zu schlagen.

3 Monitoring

Solange die Systeme die zur Verfügung stehenden Ressourcen nicht vollständig auslasten ist der Bedarf an Performance-Monitoring gering. Wenn allerdings die ersten Engpässe auftreten, dann sind die Wünsche plötzlich groß. Allerdings hält die Virtualisierung für klassische Monitoring-Tools einige Fallstricke bereit. Da unser zentrales Steuerungsmittel die CPU-Leistung ist, werden wir hier auch nur diese betrachten.

3.1 Virtual CPUs

Bei der LPAR-Konfiguration wird, neben der “entitled Capacity” auch die Anzahl der virtuellen CPUs hinterlegt. Diese Anzahl ist bei uns gleich der Anzahl physischer CPUs im System. Hintergrund ist, dass wir dynamisch bei Bedarf bis zur Maximal-Anzahl die CPUs verwenden möchten.

Was erstmal schick klingt, hat einen gravierenden Nachteil. Klassische Monitoring-Tools berechnen die CPU-Auslastung bezogen auf die virtuellen CPUs. Nehmen wir an, das System habe 16 CPUs, unsere LPAR ist mit einer CPU “entitled Capacity” definiert. Wenn das System genau die “entitled Capacity” an Leistung verbraucht, so erscheint im Monitoring nur 1/16 der CPU ausgelastet – was aber ein vollkommen unbrauchbares Bild liefert.

Das Tool `topas` zeigt eine Liste der aktiven Prozesse und vieler anderer System-Parameter. Im Kopf der Anzeige wird die aktuelle CPU-Anzahl im Vergleich zur “entitled Capacity” angezeigt (Bild 1). Zur dynamischen Anzeige ganz nützlich, aber für tiefer gehende Analysen nicht geeignet.

```

Topas Monitor for host:      localhost      EVENTS/QUEUES      FILE/TTY
Wed Feb 11 10:12:13 2009   Interval: 2    Cswitch    66989    Readch    0.0G
                               Syscall    328.5K    Writech    0.0G
Kernel    4.1    |##          | Reads     1604    Rawin     0
User      94.5  |#####     | Writes    1727    Ttyout    703
Wait      0.3  |#           | Forks     3       Igets     0
Idle      1.2  |#           | Execs     3       Namei     586
Phyisc =  8.12                %Entc= 162.4  Runqueue  21.0    Dirblk    0
                               Waitqueue  1.5

```

Figure 1: Ausschnitt aus der Anzeige von `topas`

Ein erster Ansatz war für uns, dass wir ein Monitoring aufgesetzt haben, das uns die Auslastung der CPUs bezüglich der “entitled Capacity” anzeigt. Hier haben wir auf das AIX-Tool `vmstat` zurück gegriffen. Dieses zeigt in der letzten Spalte an, wie viel CPU-Leistung im Vergleich zur nominell bereitgestellten CPU-Leistung verbraucht wird. Im Rahmen einer optimalen Systemauslastung möchten wir hier weder permanente überlast, noch unterforderte Systeme haben.

In der ersten Näherung haben wir uns der Überlast gewidmet – allerdings nur als Snapshot. Im zweiten Schritt haben wir die relative Last mit Zeitstempeln protokolliert, so dass wir auch einen Tagesverlauf (manuell) darstellen konnten.

Ein derartiges Zahlengrab ist aber auf Dauer nicht handhabbar. Heute haben wir mit Hilfe von Ganglia eine graphische Darstellung der “entitled Capacity” (Bild 2) und der verwendeten CPU-Leistung (Bild 3). Es ist möglich, die beiden Grafiken auch durch Ganglia übereinander legen zu lassen, das ist bei diesem System allerdings noch nicht konfiguriert. Leider sind auch die Skalen unterschiedlich, so dass man etwas genauer hinsehen muss.

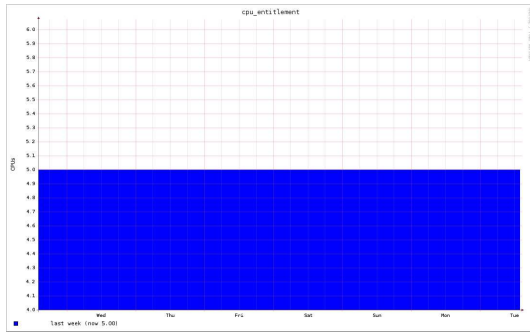


Figure 2: entitled Capacity

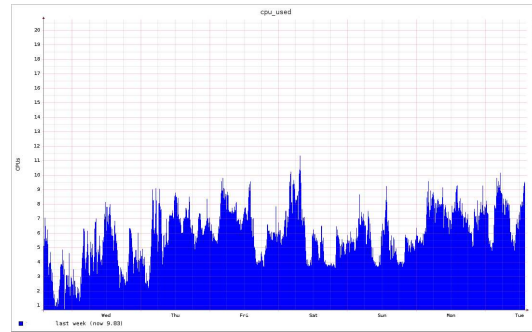


Figure 3: Verwendung der CPU

Wir erkennen aber, dass es sowohl Zeiten mit sehr hoher Last (200% der “entitled Capacity”) als auch Zeiten mit niedriger Last gibt. Aber wo kommen die Ressourcen für die Überlast-Zeiten her?

3.2 CPU Pool

Beim klassischen Monitoring betrachtet man jeweils eine LPAR für sich. Da jedoch alle unsere LPARs auf einem Rechner einen CPU-Pool dynamisch unter sich aufteilen, wollen wir auch hier ein sinnvolles Monitoring erreichen. Allerdings ist es schwer, eine “globalere” Sicht aus einer LPAR heraus zu finden.

Hier haben wir das Tool Ganglia für uns entdeckt. Neben vielen anderen (auch selbst definierten Parametern) bietet es uns die Möglichkeit, die CPU-Virtualisierung sinnvoll graphisch darzustellen.

Das Bild 4 zeigt einen Blick auf einen Rechner, auf dem mehrere LPARs in Betrieb sind.

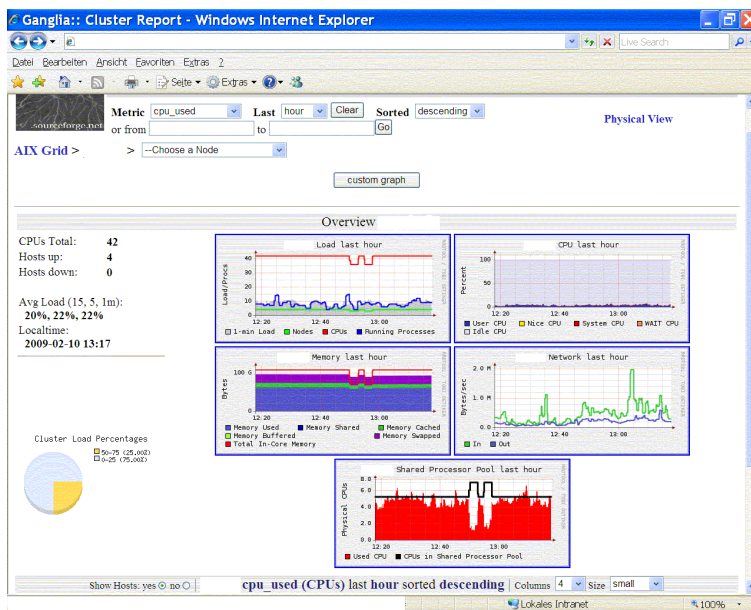


Figure 4: physikalischer Rechner

Hier sehen wir, dass der CPU-Pool nicht alle CPUs des Rechners umfaßt – eine LPAR hatte fest zwei CPUs zugeordnet. Beim zweimaligen Reboot wurden diese CPUs allerdings dem Pool hinzugefügt und standen den anderen LPARs zur Verfügung.

Von einem anderen System haben wir die Auslastung des CPU-Pools im Detail dargestellt (Bild 5).

Hier sehen wir, dass der CPU-Pool am Mittwoch um eine CPU vergrößert wurde. Am Montag ist die Systemlast deutlich gesunken – hier war ein Prozess in eine CPU-Loop gegangen. Auffällig ist hier, dass die Anzahl der benutzen CPUs größer als der CPU-Pool ist – das ist noch ein Fehler in unserer

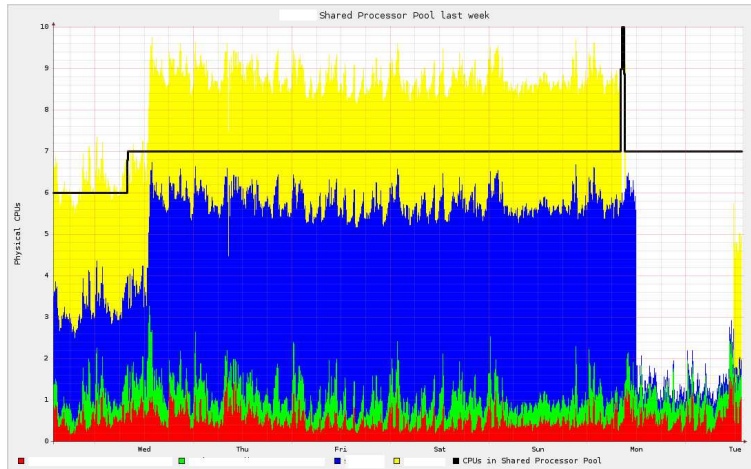


Figure 5: Auslastung eines CPU-Pools im Detail

Ganglia-Konfiguration. Die LPARs, die nicht an der Virtualisierung teilnehmen, dürfen in dieser Grafik nicht berücksichtigt werden.

4 Ganglia

Das Monitoring-Tool Ganglia besteht aus zwei Komponenten: Einem Agenten, der auf jedem überwachten System die Daten sammelt und für den Server bereit stellt. Es ist recht einfach möglich, eigene Daten zu sammeln und durch den Server darstellen zu lassen. Die andere Komponente ist der Ganglia-Server, der die von den Agenten gesammelten Daten speichert und grafisch darstellt.

Die Speicherung der Daten erfolgt in den bekannten Round-Robin-Datenbanken, wie man sie aus RRDTool, mrtg oder cacti kennt. Damit ist sichergestellt, dass die Platten des Servers nicht voll laufen. Allerdings sind nur die Daten maximal der letzten Woche sinnvoll für eine Detail-Analyse zu nutzen. Ältere Daten dienen im allgemeinen nur zur Visualisierung von Trends.

Wichtig für uns war, dass wir ganz gezielt die Power-Virtualisierung betrachten können.

5 nmon

Nun ist es häufig so, dass Anwender sich mit Performance-Klagen melden. In den meisten Fällen betrachten wir zunächst das System mit Tools wie `topas` oder `nmon`, um einen Überblick über die aktuelle Auslastung zu gewinnen. Allerdings betrachtet man dabei immer nur einen Snapshot, so dass die Sichtweise immer eingeschränkt ist.

Wenn wir den Verdacht haben, dass die Virtualisierung an unseren Problemen beteiligt ist, so werfen wir einen Blick ins Ganglia. Anhand der Ausgabe von `lparstst -i` erkennen wir, auf welchem physischen Rechner die LPAR zu Hause ist und können diese dann genauer ansehen.

Dabei erweist es sich immer wieder als nützlich, dass neben der Ansicht der letzten Stunde bzw. des letzten Tages auch Sichten zur letzten Woche, des letzten Monats und des letzten Jahres zur Verfügung stehen. Damit lassen sich recht schnell gewisse Trends erkennen und gegebenenfalls werden Tage offensichtlich, die eine nähere Analyse sinnvoll erscheinen lassen.

Für eine Detail-Analyse lassen wir vom Tool `nmon` täglich eine Datei erzeugen, die die Performance-Daten einer LPAR enthält. Mit Hilfe des `nmon`-Analysers können wir diese Daten grafisch aufbereiten und vielfach interessante Aspekte genauer untersuchen. Dabei heben wir die `nmon`-Daten der Systeme für mehrere Monate auf.

Leider lassen sich nicht alle Daten in einem Tool analysieren.

6 Zusammenfassung

Performance-Monitoring und Tuning hatte schon immer etwas mit schwarzer Magie zu tun. Nur selten ist offensichtlich, wo der Engpass zu suchen ist. Und noch seltener, wie dieser möglichst kostengünstig zu beseitigen ist.

Mit dem Einsatz von Virtualisierung ist die Lage noch viel unübersichtlicher. Gerade in den heute üblichen, sich doch recht schnell ändernden Systemumgebungen und Anwendungsanforderungen ist der Einsatz von sinnvollen Werkzeugen unabdingbar. Unser Ziel war es, mit überschaubarem Einsatz ein grundsätzliches Monitoring der Umgebung unter Berücksichtigung der Virtualisierung zu implementieren.

Dieses Ziel konnten wir mit einigen eigenen Monitoren, sowie frei verfügbaren Tools wie topas, nmon und ganglia erreichen. Allerdings ist es nicht einfach, dass alle notwendigen Parameter korrekt in den verschiedenen Anzeigen auftauchen – hier werden wir in den nächsten Wochen immer wieder kleinere Anpassungen vornehmen.

References

- [1] <http://www.ganglia.info> ist die Homepage zu Ganglia
- [2] <http://www.ibm.com/developerworks/wikis/display/WikiPtype/nmon>