



Mag. iur. Dr. techn. Michael Sonntag

KV Betriebssysteme

Virtual Memory

Institut für Informationsverarbeitung und Mikroprozessortechnik (FIM)
Johannes Kepler Universität Linz, Österreich

E-Mail: sonntag@fim.uni-linz.ac.at
<http://www.fim.uni-linz.ac.at/staff/sonntag.htm>

© Michael Sonntag 2004

Fragen?

Bitte gleich stellen!

© Michael Sonntag 2004

Virtual Memory

- Warum und Wozu?
 - Sowohl Daten als auch Programme müssen bei der Ausführung im Speicher liegen
 - » Aber nur während der **tatsächlichen Ausführung!**
 - » Zustand: Waiting → Speicher könnte anderweitig verwendet werden
 - Idealzustand: **Gesamter logischer Speicher wird auf physischen abgebildet**
 - Ev. mit Löchern, wo er nicht verwendet wird
 - » Problem: Speicher (in diesen Mengen!) ist immer noch **teuer**
- Grundgedanke:
 - Nur was **tatsächlich gerade jetzt (oder bald) gebraucht wird** (werden wird) ist im physischen Speicher
 - » Der Rest wird **zwischenzeitlich ausgelagert**
 - Typischerweise: Festplatte

Michael Sonntag Virtueller Speicher, Performance Monitoring 3

Virtual Memory

- Virtueller Speicher (=“VM“):
 - Mehr Speicher ist logisch vorhanden und zur Verfügung, als physische tatsächlich an RAM existiert
 - Für das Programm selbst völlig transparent
 - » Aufgabe des Betriebssystems!
- (Praktische) Voraussetzungen:
 - Einteilung des Speichers in gleichgroße Einheiten
 - » Es werden nur gesamte “Pakete” ein-/ausgelagert (Verwaltung!)
 - » Keine externe Fragmentierung
 - Prozesse können vom OS ohne Fehler unterbrochen werden, wenn eine Seite fehlt und später fortgesetzt werden
 - » Jeder Speicherzugriff muß “on hold” gestellt werden können
 - » “Page fault”: Löst Aufruf des VM-Manager aus

Michael Sonntag Virtueller Speicher, Performance Monitoring 4

Begriffs-Wiederholung

- Physischer Speicher
 - RAM-Bausteine im Rechner
- Logischer Speicher
 - Sicht des Programmes auf den Speicher
- Virtueller Speicher
 - Was das OS vorgibt, an Speicher zur Verfügung zu stellen
- Hintergrundspeicher
 - Auslagerung von logischem Speicher auf Festplatte
- Page (Seite)
 - Einheit im logischen Speicher (tatsächliche Dateneinheit)
- Frame (Seitenrahmen)
 - Einheit im physischen Speicher die Pages belegen können
 - » Potentieller Platz, “Raumeinteilung”

Michael Sonntag Virtueller Speicher, Performance Monitoring 5

Verwaltungs-Voraussetzungen

- Algorithmus zur Bestimmung, welche Page(s) eingelagert werden sollen
 - Trivial: Was referenziert wird (schlechte Performance!)
 - Abhilfe: Prefetching
- Auswahl des Frames, den eine Page erhalten soll
 - Freispeicherverwaltung; wird auch ohne VM benötigt!
- Auswahl der Page, die ausgelagert/verworfen werden soll
 - Wenn der physische Speicher voll ist und eingelagert werden muß, so ist eine derzeit vorhandene Page zu “entfernen”
 - » Unveränderte Daten: Verwerfen
 - Können von “originalem” Platz jederzeit neu geladen werden
 - Problem: Wie findet man die dort?
 - » Geänderte Daten: In den Auslagerungsspeicher schreiben
- Anzahl der Frames, die ein Prozess erhält

Michael Sonntag Virtueller Speicher, Performance Monitoring 6



VM-Algorithmen: Grundlagen

- Page Reference String
 - Welche Pages werden in welcher Reihenfolge von einem konkreten Programmablauf verlangt
 - » Zum Testen/Vergleichen von Strategien
 - » Vorhersage teilweise möglich!
 - Beispiel: Code wird meist linear durchlaufen
 - Daten werden oft sequentiell verarbeitet
 - Konkret: Folge der Seitennummern, auf die zugegriffen wird
- Strategien: Probleme und Limitierungen
 - Belady's Anomalie: Mehr physischer Speicher hilft nicht immer
 - "Grenzwert": VM kann nicht beliebig groß werden
 - » Extrem: 1 kB physisches RAM, 1MB logisch verwendet
 - Siehe: Seitenflattern!
 - "Beste" Strategie hängt von verwendeten Programmen ab!

Michael Sonntag Virtuellet Speicher, Performance Monitoring 7

Seiteneretzungsstrategien

- Optimal: Welche Seite am längsten nicht referenziert wird
 - Voraussetzung: Eingebautes Orakel für Blick in die Zukunft!
- LRU (Least Recently Used): Jene Seite, die am längsten nicht referenziert wurde
 - Annahme: Zukunft ≈ Vergangenheit
- NUR (Not Used Recently): Einfachere Version von LRU
 - Weniger Aufwand bei der Implementierung
 - » Page Replacement kommt in einem OS **SEHR OFT** vor!
- FIFO, LIFO: First/Last In, First Out
 - Einfache Verwaltung, meist schlechte Performance
 - Berücksichtigen keine Taskwechsel, ev. gut für Singletasking!
- ... (meist basierend auf mathematischen Modellen)
 - "Maßgeschneidert": Wenn die Programme genau bekannt sind

Michael Sonntag Virtuellet Speicher, Performance Monitoring 8

VM: Windows

- Eigene Seitentabelle für jeden Prozess (potentiell 4 GB)
 - Wird dynamisch erstellt, d.h. nur soweit sie gebraucht wird
 - Kann selbst ausgelagert sein
 - » Mehrere Pagefaults für einzigen Speicherzugriff möglich
- Bis zu 16 Auslagerungsdateien möglich
 - Nur auf verschiedenen Disks (logisch, nicht physisch!)
 - » Jeweils max. 4 GB groß
 - Nachteil: Echte Datei, kann daher fragmentiert sein
 - » Dann schlechte Performance
 - » Vorher defragmentieren!
- Richtwert: VM maximal 2*RAM
 - 256 MB RAM → 512 MB Auslagerungsdatei

Michael Sonntag Virtuellet Speicher, Performance Monitoring 9

VM: Windows

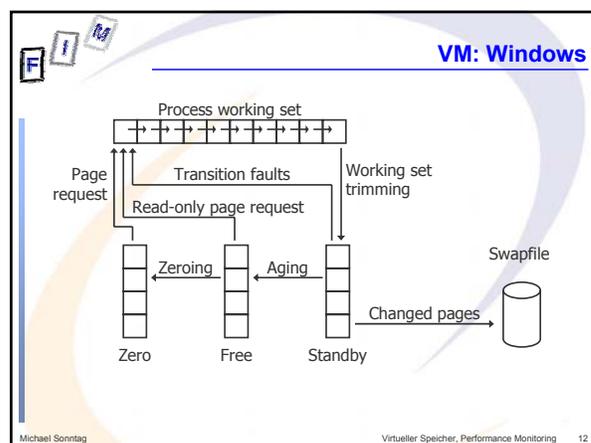
- Fetching: On Demand
 - Warten auf Seitenfehler
 - Es werden aber gleichzeitig mehrere Pages eingelagert
 - » Davor und dahinter liegende
 - Windows XP: Statistik oft benötigter Pages ("Logical Prefetch")
 - » Werden schon bei Programmstart eingelagert
- Placing: Erster freier Frame wird verwendet
- Replacement: FIFO
 - Älteste Seite des working set wird ausgelagert
 - » Daher eigentlich LRU-Algorithmus!
 - Working set ist für jeden Prozess separate

Michael Sonntag Virtuellet Speicher, Performance Monitoring 10

VM: Windows

- Statuswerte von Frames:
 - Valid: Von aktivem Prozess verwendet
 - Modified: Wurde beschrieben, ist aber nur im Speicher
 - » Noch nicht auf die Festplatte geschrieben
 - Standby: Aus dem working set eines Prozesses entfernt
 - » Kann aber bei Bedarf wieder in den Zustand "Valid" gehen!
 - Free: Frei, muß aber noch gelöscht werden
 - » Außer, sie wird als "read-only" verwendet
 - Zeroed: Mit '0' gefüllt, kann beliebig verwendet werden
 - Bad: Hardwarefehler in diesem Bereich
- Jede Sekunde (oder bei Bedarf) erfolgt "trimming"
- Standby → Free je nach Bedarf von Seiten
- Free → Zeroed durch Thread sehr niedriger Priorität

Michael Sonntag Virtuellet Speicher, Performance Monitoring 11





VM: Linux 2.4

- Swap-"Datei" ist eine eigene Festplatten-Partition
 - Daher zwangsweise unfragmentiert
 - Spezielles "Dateisystem"
 - Maximal 32 pro System
- Auslagerung ist eigener Thread (kswapd), der bei freier Zeit (oder bei Bedarf) Seiten auf die Festplatte schreibt
 - Unter- und Obergrenze
 - » Untergrenze: Es wird ausgelagert, bis mind. so viele Frames frei
 - » Obergrenze: Es wird überhaupt nichts ausgelagert
- Bei Frame-Bedarf werden der Reihe nach alle Prozesse gesondert nach NRU-Pages durchsucht → Fairness
 - Gefahr von Thrashing: Viel Speicherbedarf führt dazu, daß alle Prozesse Frames hergeben müssen, und daher alle Probleme haben (anstatt daß einer Priorität erhält)

Michael Sonntag Virtuellet Speicher, Performance Monitoring 13

VM: Linux 2.4

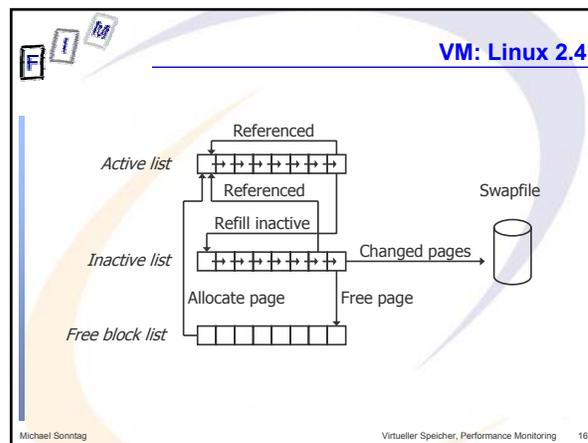
- Fetching: On demand
 - "Spezialbehandlung" für streaming IO
 - » Vergangene Seiten (bereits geschrieben) erhalten sehr schlechten Age-Wert
 - » Zukünftige Seiten (bald zu lesen) erhalten besseren Age-Wert
- Placement: Binary Buddy Algorithm für Reservierung
 - Nach Möglichkeit wird eine der CPU des betroffenen Prozesses zugeordnete Speicherbank ausgewählt
- Kernel 2.6: Optimierungen für shared pages
 - "Rückwärts-Tabelle" um von einem Frame zu allen Prozessen zu kommen, zu welchen diese Page gehört

Michael Sonntag Virtuellet Speicher, Performance Monitoring 14

VM: Linux 2.4

- Replacing: LRU (aber nicht exakt; Elemente von LFU)
 - » LRU = Least Recently Used
 - » LFU = Least Frequently Used
- Liste von aktiven (=referenzierten) und inaktiven (=clean oder dirty, d.h. zu löschend oder auszulagernden) Pages
 - » Liste ist global; enthält alle Prozesse
 - » Inaktiv und clean: Können Prozess wieder zugeordnet werden
- Referenzierte Pages werden Aging (allerdings nicht exakt Zeitbezogen) unterworfen
- Schreiben auf Auslagerungspartition erfolgt erst, wenn keine inaktiven freien Frames mehr vorhanden sind
 - Weniger Disk-IO; mehrere Pages auf einmal schreiben

Michael Sonntag Virtuellet Speicher, Performance Monitoring 15



Literatur

- Understanding the Linux Virtual Memory Manager: <http://www.csn.ul.ie/~mell/projects/vm/guide/html/understand/understand-html.html>
- Page replacement in Linux 2.4 memory management: http://www.usenix.org/events/usenix01/freenix01/full_papers/riel/riel_html/index.html
- Windows NT page replacement policies: <http://www.demandtech.com/Resources/Papers/WinMemMgmt.pdf>

Michael Sonntag Virtuellet Speicher, Performance Monitoring 17