F Übung 1

- Unter ~iwi425/PPS/streams finden Sie die Sourcen des Streams-Benchmark zur Bestimmung der Speicherbandbreite und der Rechenleistung in Abhängigkeit von der Arbeitsmenge.
  Arbeiten Sie auf einer unbelasteten Workstation vom Typ SUN Ultra-10 (300 MHz Ultra-SPARC IIi, 128 MB Speicher, 512 KB physical Cache, 64 entry Data-TLB)
  - ◆ Ändern Sie den Benchmark (C-Variante) derart ab, daß Sie bei einem Workingset von 32 KB (Summe aller beteiligten Cache-Zeilen) bei jedem Speicherzugriff einen TLB-Zugriffsfehler erzeugen.
    Vergleichen Sie Ihre Ergebnisse mit der einer TLB-freundlichen Variante.
  - ◆ Weisen Sie den Effekt des zufälligen Page-Coloring nach.
    Allokieren Sie im Streams Benchmark 0.5 MB Speicher und lassen die Messung durchlaufen, beenden den Prozeß aber nicht. Jetzt starten Sie mehrfach den gleichen Prozeß mit 100 MB Speicheranforderung. Heftige Auslagerung von Seiten ist die Folge. Lassen Sie jetzt den zuerst gestarteten Benchmark weiterlaufen und messen Sie den Druchsatz.

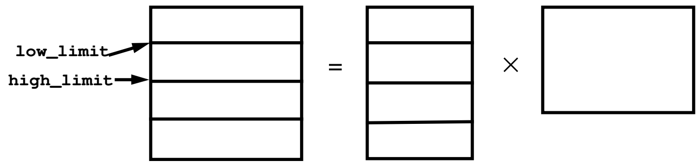
F Übung 2

- Arbeiten Sie auf einer unbelasteten Workstation vom Typ SUN Ultra-10 (300 MHz Ultra-SPARC IIi, 128 MB Speicher, 512 KB physical Cache, 64 entry Data-TLB)
  - ◆ Schreiben Sie ein 2-dimensionales Feld mit Integerzahlen (Feldgröße 110 MB) in eine Datei unter /usr/tmp.
    - Schreiben Sie mit dem **`write()`** call und analysieren Sie unterschiedliche lange Puffergrößen (8190 Bytes, 8192 Bytes).
    - Allokieren Sie eine Datei von 110 MB (**`open()`**, **`lseek()`**), mappen Sie die Datei in den Addressraum (mmap), initialisieren Sie das Feld (z.B. alle Feldelemente auf 1) und machen sie die Änderungen persistent.
    - Summieren sie alle Element des in die Datei gespeicherten Feldes auf. Wählen Sie einen Ansatz mit **`read()`** und einen mit **`mmap()`**.

F Übung 3

- Implementieren Sie analog zur Matrix-Vektoroperation eine parallele Matrixmultiplikation mit Pthreads

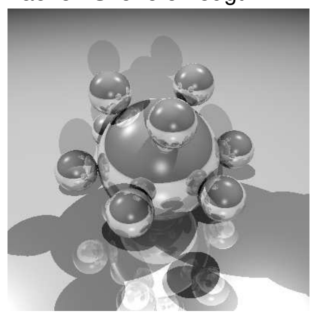
$$C[i, j] = \sum_{k=1}^{s} A[i, k] \cdot B[k, j] \,;\; 1 \le i \le n, 1 \le j \le m$$



  - ◆ Vereinfachung: Quadratische Matrizen der Größe n×n ; n = dimension
  - ◆ Compilieren Sie mit /opt/SUNWspro/bin/cc -mt ..... -lpthread
  - ◆ Variieren Sie die Zahl der Kernel-Threads.
  - ◆ Messen Sie den Speedup auf dem Rechner cssun mit 1-8 Kernel-Threads in Abhängigkeit von der Matrixgröße.

F Übung 4

- Raytracer eignen sich hervorragend zur Parallelisierung, da jeder Bildpunkt unabhängig von den anderen berechnet werden kann.
- In `~iwi425/PPS/ray/` findet sich ein kleiner Raytracer, der das Bild einer einfachen Szene erzeugt.



- Start des raytracers mit **`tr | /local/bin/xv -`**

F Übung 4 (2)

- Parallelisieren Sie den Raytracer mit Hilfe von Pthreads.
  - ◆ Unterteilen Sie hierzu das Bild in einzelne Streifen und lassen sie die Streifen von eigenen Threads berechnen.
  - ◆ Vergessen sie nicht, in Ihr Program den Aufruf von `thr_setconcurrency(8)` einzubauen, um auf der `cssun` mit allen 8 Prozessoren rechnen zu können!
  - ◆ Warum ist die einfache Aufteilung in 8 Streifen nicht optimal? Die Rechenzeit verkürzt sich, wenn mehr Streifen und damit (Userlevel-)Threads eingesetzt werden. Was ist die optimale Anzahl von Threads?
  - ◆ Vergrößern Sie den Bildbereich, so daß ein starkes Lastungleichgewicht bei 8 Threads entsteht.
    - Implementieren Sie jetzt mit Hilfe von Thread-Cancellation eine Repartitionierung der noch nicht berechneten Streifen.
    - Lösen Sie diese Aufgabe auch durch den Einsatz von Signalen (SIGUSR1).

F Übung 5

- In dieser Aufgabe soll ein HTTP Server parallelisiert werden. Der serielle Source findet sich in `~iwi425/PPS/tinyhttp/`
- Zum Testen kann der WWW Baum des RRZE verwendet werden. Ein Aufruf wäre:
  `./tinyhttp 2345` /home/rzhome/iwi4/iwi425/PPS/wwwcip
  Danach könnte zum Beispiel über die URL
  `http://cssun.rrze:12345/docs/RRZE/`
  auf den Server zugegriffen werden können.
- Da der Server zu langsam arbeitet, sollen die einzelnen Requests parallel von Threads abgearbeitet werden.
- Implementieren Sie den Server mit dem Boss-Worker Modell.
- Setzen Sie einen Signalhandler auf, der die Anwendung geordnet mit Cancellation terminiert.