Der Datentyp Bigint

- speichert beliebig große Ganzzahlen in einem großen Array
- Bereitstellung der benötigten Operationen (+,*,=,<<), mittlerweile auch -, /, %, ==, etc.
- Operationen möglichst optimiert: Bei Addition werden nur überlappende Teile addiert, der Rest kopiert
- Darstellung der Zahl mit jeweils 4 Dezimalziffern pro 16 Bit
- Im Dezimalsystem, da ansonsten Textausgabe unmöglich
- Warum 16 Bit?

  → 32bit Hardware - Multiplikation ausnutzen
- Auf Transputer Darstellung von Zahlen bis 2.000.000 Ziffern, einzig durch Speicher beschränkt

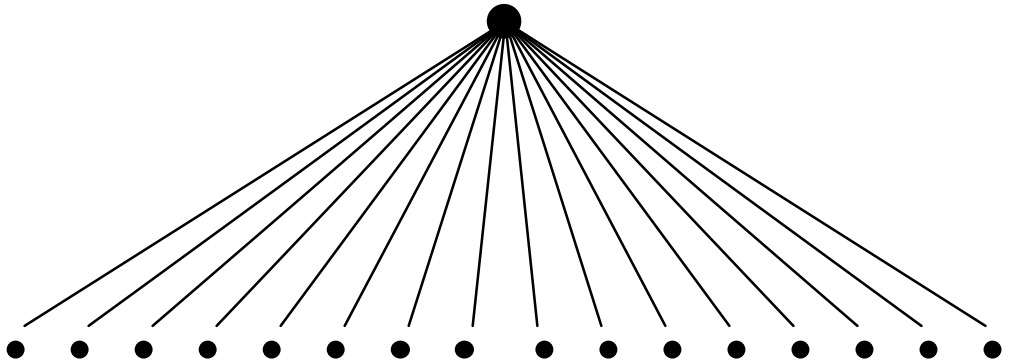
Paralleles Multiplizieren

$$
\begin{aligned}
10242048 * 40968192 &= 10242048 * 40960000 \\
&+ 10242048 * 00008192 \\
&= (10242048 * 4096) * 10000 \\
&+ (10242048 * 8192) \\
&= 419598188937216
\end{aligned}
$$

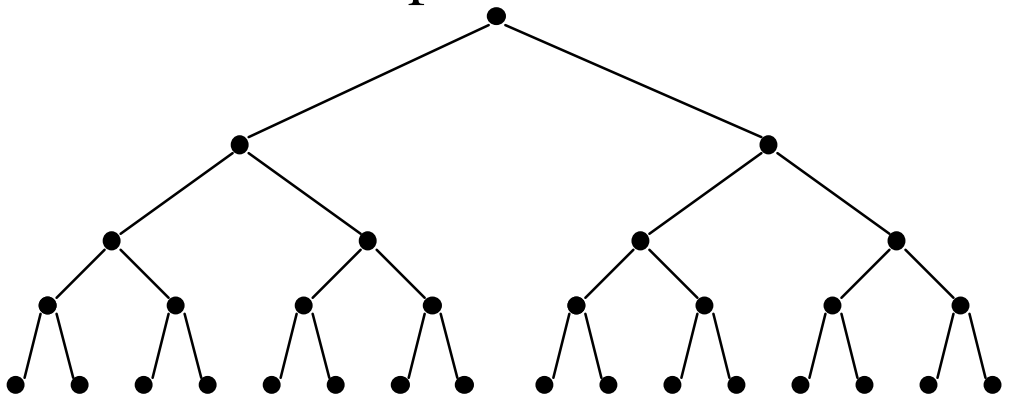
- Eine Zahl wird geteilt
- Sequentiell: 64 Multiplikationen
- Parallel: 32+32 Multiplikationen + 7 Additionen + 1 Shift
- Multiplikationen sind optimal parallelisierbar, da unabhängig
- Additionen und Shifts sind "gratis"
- Zahl kann in beliebig viele Teile geteilt werden, d.h. beliebige Parallelisierbarkeit

Berechnungsstrategie

- Strategie 1 - Slaves rechnen, Master multipliziert



- Strategie 2 - Slaves multiplizieren



Berechnungsstrategie

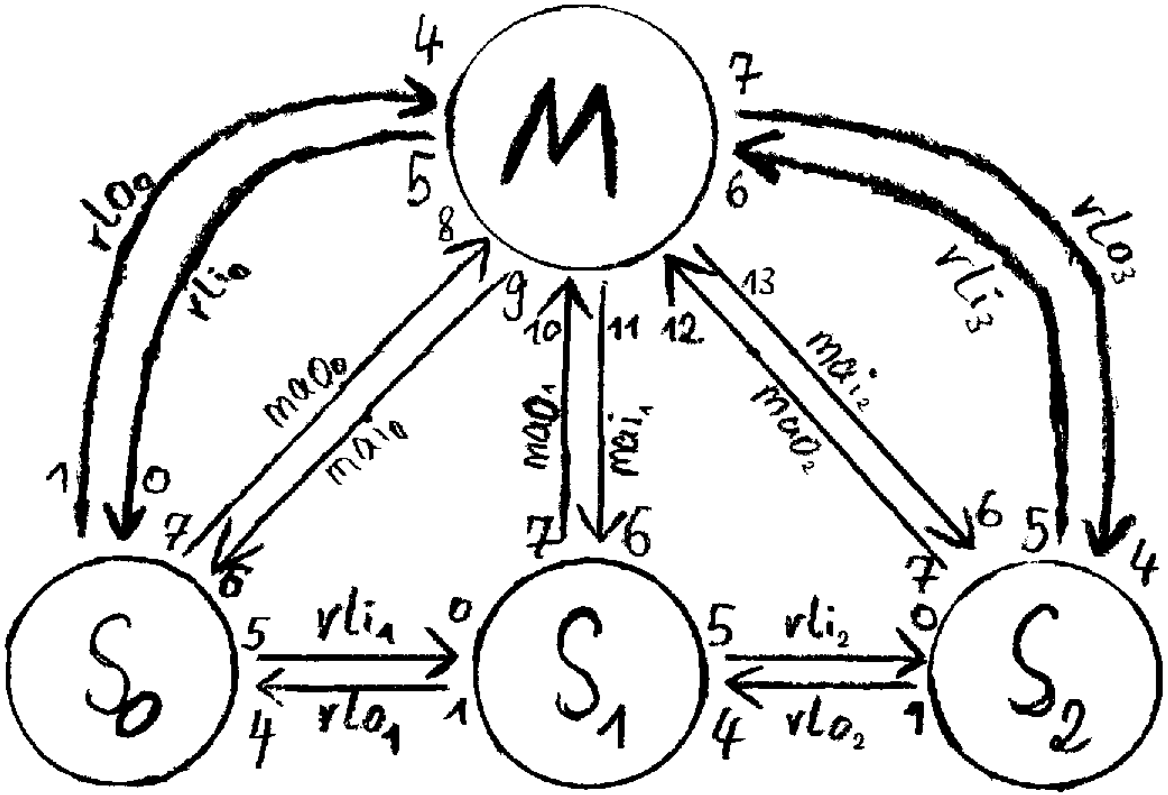
- Aufteilen in große Anzahl Teilintervalle
- Verschicke jeweils ein Intervall an einen Prozeß zum ausmultiplizieren $\prod_{i=anfang}^{ende} i$
- Wenn zwei Teilergebnisse vorliegen, verschicke diese zur Multiplikation. Sind mehere Prozesse verfügbar, parallelisiere die Aufgabe.
- Ansonsten erhält der Prozeß ein neues Teilintervall
- Das Ergebnis der letzten Multiplikation ist gleichzeitig das Endergebnis
- Vorteil der Strategie: flexibel, da unabhängig von der Prozeßanzahl.

Beispiel

Berechnung von 1000! auf 3 Prozessen, "Scheibengröße" 250

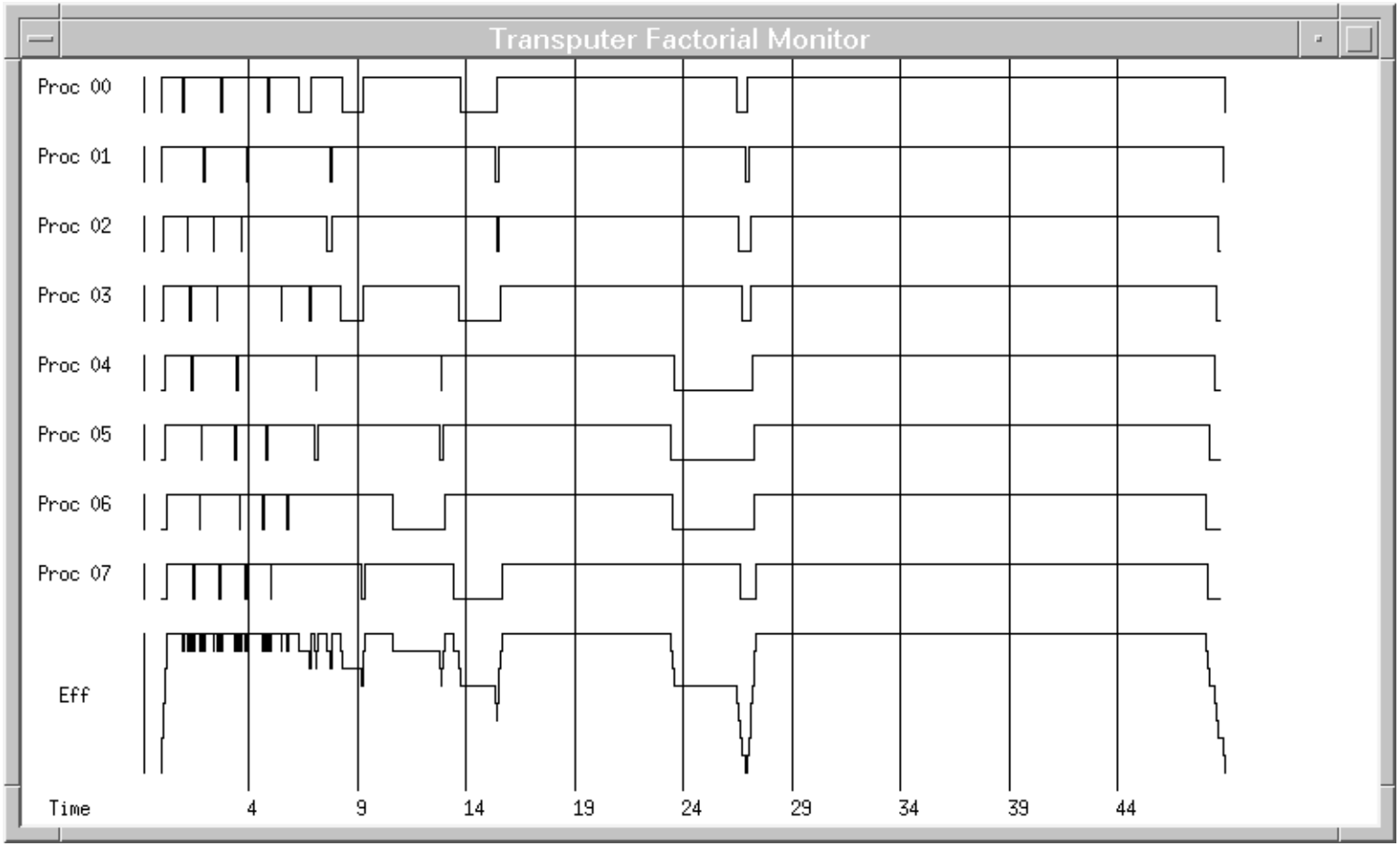
① P1 multipliziert von 1-250, P2 von 251-500, P3 501-750
② Prozeß 1 liefert Ergebnis und erhält 751-1000 zur Berechnung
③ Prozeß 2 liefert Ergebnis. Zwei Teilergebnisse sind bekannt, diese müssen multipliziert werden → Aufgabe an P2
④ Prozeß 3 liefert Ergebnis.
⑤ Prozeß 1 liefert 751-1000. Zwei Teilergebnisse sind bekannt, und zwei Prozesse (P1, P3) sind frei; Multiplikation wird parallelisiert
⑥ Prozeß 2 liefert Multiplikationsergebnis
⑦ Prozesse P1, P3 liefern Multiplikationsergebnis.
⑧ Finale Multiplikation wird auf alle 3 Prozesse verteilt.

Kommunikationsstruktur



bidirektionaler Ring und bidirektionale Farm gleichzeitig
erreicht durch "CDL-Datei-Generator"

Berechnungsbeispiel



Fakultätsberechnung

$$n! = \prod_{i=1}^{n} i$$