



Lösungen zum Übungszettel 7

Aufgabe 1:

a) Hochverfügbarkeitscluster:

Hochverfügbarkeitscluster (engl. High-Availability-Cluster – HA-Cluster) werden zur Steigerung der Verfügbarkeit bzw. für bessere Ausfallsicherheit eingesetzt. Tritt auf einem Knoten des Clusters ein Fehler auf, werden die auf diesem Knoten laufenden Dienste auf einen anderen Knoten migriert. Die meisten HA-Cluster besitzen 2 Knoten.

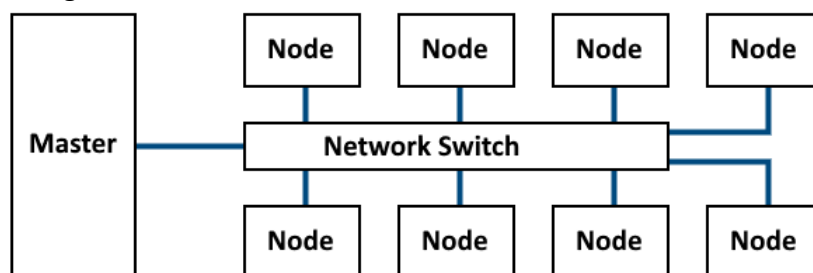
Load-Balancing-Cluster:

Load-Balancing-Cluster werden zum Zweck der Lastverteilung auf mehrere Maschinen aufgebaut. Die Lastverteilung erfolgt in der Regel über eine redundant ausgelegte, zentrale Instanz. Viele Webseiten nutzen dies zum Beispiel um die große Menge an individuellen Seitenaufrufen bewältigen zu können.

High-Performance-Computing-Cluster:

High-Performance-Computing-Cluster (HPC-Cluster) dienen zur Abarbeitung von Rechenaufgaben. Diese Rechenaufgaben werden auf mehrere Knoten aufgeteilt. Entweder werden die Aufgaben in verschiedene Pakete aufgeteilt und parallel auf mehreren Knoten ausgeführt oder die Rechenaufgaben (Jobs genannt) werden auf die einzelnen Knoten verteilt. Die Aufteilung der Jobs übernimmt dabei meistens ein Job Management System. HPC-Cluster finden sich oft im wissenschaftlichen Bereich. In der Regel sind die einzelnen Elemente eines Clusters untereinander über ein schnelles Netzwerk verbunden.

- b) Ein Beowulf-Cluster besteht aus einer gewissen Anzahl von Rechen-Knoten (compute nodes), einem oder mehreren Server-Knoten (server nodes) und in der Regel aus einem (oder mehreren) Zugangs-Knoten (front end), auf dem bzw. denen sich die Nutzer einloggen können. Von dort aus können sie sich die benötigte Menge von Rechen-Knoten für ihre Arbeit reservieren und diese benutzen.





-
- c) Die Bedienung des Clusters erfolgt über einen Master-Knoten, der auch die zu erledigende Aufgabe „Job“ in kleine Teile zerlegt (Decomposition-Programm) und dann mittels eines Job-Scheduling-Programms auf die Knoten verteilt. Die einzelnen Knoten kommunizieren über IP. Der Datenaustausch/die Kommunikation zwischen den auf verschiedenen Knoten laufenden Job-Teilen geschieht in der Regel über standardisierte Bibliotheken, die eine abstrakte Kommunikationsschnittstelle zur Verfügung stellen. Die bekanntesten Vertreter solcher Bibliotheken sind Message Passing Interface (MPI) und Parallel Virtual Machine (PVM).

Aufgabe 2:

Deadlocks (Kommunikation):

Verklebungen können bei paketvermittelnden Netzen in allen Netztopologien auftreten.

Verhinderung: Die Zweier-Verklebung lässt sich durch getrennte Sende- und Empfangspuffer vermeiden, weil durch diese Maßnahme auch dann noch Daten von Nachbarknoten entgegengenommen werden können, wenn die Sendepuffer voll sind. Oft muss man virtuelle Kanäle etc. einführen.

Deadlocks (Prozessebene):

Ein Deadlock tritt immer dann auf, wenn zwei (oder mehr) Prozesse ein Objekt beanspruchen, das jeweils der andere Prozess gerade in Besitz hat und damit blockiert. Deadlocks können in verschiedensten Situationen in der parallelen Programmierung auftreten.

Verhinderung: Ein Ansatz liegt darin, alle für einen Prozess erforderlichen Objekte vor der Ausführung zu sperren und dabei die Sperrzeit so gering wie möglich zu halten. Verschiedene Programmiersprachen bieten Timeouts an, die dann Deadlocks verhindern sollen.

Erkennung von Deadlocks:

Dazu ist der Wartegraph zu bestimmen. Jeder Zyklus im Wartegraph repräsentiert einen Deadlock.