

Exercise 2 - Multi-Processor DBMS

1. Shared-Everything, Shared-Disk, Shared-Nothing

There is a simply structured databases about customers and their billing information (e.g. telephone costs) of a company. There are about 100 users of the database at any time resulting in approximately 10 transactions per second. Typical transactions are read transaction consisting of only one query of one of the 2 following patterns:

```
SELECT town, count(*)  
FROM customer  
GROUP BY town
```

or

```
SELECT *  
FROM customer NATURAL JOIN bills  
WHERE name='yyy'
```

1. Given is the following **Shared-Everything** scenario: a multi-processor machine with 8 processors is running to serve the customer management application made accessible via a Web-Interface.
 - (a) How could the performance of the system benefit from the parallel architecture?
 - (b) What advantages/disadvantages exist regarding availability and scalability?
2. Given is the following **Shared-Disk** scenario: there is a cluster of 3 DB-Servers all accessing the database stored in a network file system serving the Customer Relationship Management system running as an application in the Local Area Network (LAN).
 - (a) What could be methods for load balancing in this scenario?
 - (b) In this context: what is the problem of cache invalidation? Under which changed circumstances would the problem exist?
3. Given is the following **Shared-Nothing** scenario: the company has three subsidiaries in Munich, Hamburg, an Berlin, connected in a Wide Are Network (WAN), each running their own database system storing only data of customers from the local area South, North, and East, respectively.
 - (a) What are the advantages of storing the data in this fragmented fashion? What would the query processing require if all data was stored only in Berlin?
 - (b) What positive effect would the replication of (part of) the data have on the system?