Leonard Wassenaar

25948083

ITRW 321

SU3 Summary

**Distributed Database Management Systems**

Distributed database management system:

A DDBMS is a database management system that supports a database that is distributed to multiple sites.

Why is the distributed database more desirable than a centralized database?

Because the centralized database is known for;

* Having performance degradation
* Being costly
* Not being reliable
* Having scalability issues
* Organizational rigidity

Distributed processing:

More than one site that shares the logical processing of the database on a network.

Distributed database:

A database that is logically related, stored in 2 or more physical sites that is independent.

DDBMS characteristics:

* Interface
* Validation of syntax
* Transformation for requests decomposition
* Query optimizer
* Mapping for data location
* Input/output interface
* Formatting of data
* Security
* Backup and recovery
* Database administration features
* Concurrency control
* Management of transactions

DDBMS Components:

* Computer workstations and software components
* Network software and hardware components
* Media for communication
* Transaction processor – Software that requests data.
* Data processor – Software that stores and fetches data at a site.

Data and process distribution levels:

|  |  |
| --- | --- |
| Single-site processing, single-site data | Processing takes place on a host computer and the data is stored on the disk of the host computer. |
| Multiple-site processing, single-site data | Different processors on several computers sharing the same repository. |
| Multiple-site processing, multiple-site data | Supports several data processors at different sites. |

Distributed database transparency features:

|  |  |
| --- | --- |
| Distribution transparency | Enables a distributed database to be regarded as a single database.   * Fragmentation transparency * Location transparency * Local mapping transparency |
| Transaction transparency | Transaction can update data at 2 or more network sites. |
| Failure transparency | Make sure that the system will be operational in the event of a network failure. |
| Performance transparency | Enables the system to perform like if it was centralized. |
| Heterogeneity transparency | Enables the integration of several local database management systems. |

Distributed database design:

|  |  |
| --- | --- |
| Data fragmentation: Enables an object to be broken into separate fragments. | |
| Horizontal fragmentation | Divide a relation into fragments of rows. |
| Vertical fragmentation | Divide a relation into column subsets. |
| Mixed fragmentation | Combination of horizontal and vertical fragmentation. |

Data replication:

Occurs when duplicate database fragments are stored on different sites. Provides fault tolerance and performance.

|  |  |
| --- | --- |
| Full replicated database | Multiple copies of each database fragments are stored at different sites. |
| Partial replicated database | Multiple copies of some database fragments are stored at different sites. |
| Unreplicated database | Each fragment is stored at one site. |

Data Allocation:

Determines where the data should be stored.

|  |  |
| --- | --- |
| Centralized | The whole db is stored at a single site. |
| Portioned | Divided and stored on 2 different sites. |
| Replicated | 1 or more fragments of the database is stored at different sites. |

Data allocation algorithms factors:

* Performance
* Size that an entity must maintain with other entities.
* Type of transactions
* Disconnected operation for users.

CAP theorem:

* Consistency – All nodes must see the data at once.
* Availability – No request that is fetched must go missing.
* Partition tolerance – The system operation must continue in the event of a failure.