Table of Contents

File Application

1. A distributed file application consists of one or more local or remote clients that communicate with one or more servers on several machines linked through a network.
2. With this type of application, business operations can be conducted from any geographical location.
3. For example, a corporation may distribute the following types of operations across a large region, or even across international boundaries:
    - Forecasting sales
    - Ordering supplies
    - Manufacturing, shipping, and billing for goods
    - Updating corporate databases
4. State of the art telecommunications and data networks are making distributed operations of this sort increasingly common.
5. Applications developed to implement this type of strategy allow businesses to reduce costs and enhance their offerings of services to customers around the world.
6. The BEA Tuxedo system supports this type of architecture by simplifying the task of

managing a distributed application.

7. Whether an application comprises only one computer or thousands of computers working together over a network, all the elements of that application, including clients, servers, and the networks that connect them, are managed through a single BEA Tuxedo configuration file.

Benefits of a Distributed file Application

1. Scalability

The load an application can sustain can be increased by placing extra server processes in a group; adding machines to the application and redistributing the groups across the machines; replicating a group onto other machines within the application and using load balancing; segmenting a database and using data-dependent routing to reach the groups dealing with these separate database segments.

2. Ease of development/maintainability

The separation of the business application logic into services or components that communicate through well-defined messages or interfaces allows both development and maintenance to be similarly separated and so simplified.

3. Resilience

When multiple machines are in use and one fails, the remainder can continue operation. Similarly, when multiple server processes are within a group and one fails, the others are present to perform work. Finally, if a machine should break, but there are multiple machines within the application, these other machines can be used to perform the work of the

application.

4. Coordination of autonomous actions

If you have separate applications, you can coordinate autonomous actions among the applications. You can coordinate autonomous actions as a single logical unit of work. Autonomous actions are actions that involve multiple server groups and/or multiple resource manager interfaces.

Fault Tolerance

1. Fault-tolerant is the ability of a computer system or component so that, in the event that a component fails, a backup component or procedure can immediately take its place with no loss of service.
2. Fault tolerance can be provi
3. ded with software, or embedded in hardware, or provided by some combination.
4. In the software implementation, the operating system provides an interface that allows a programmer to “checkpoint” critical data at pre-determined points within a transaction.
5. In the hardware implementation (for example, with Stratus and its VOS operating system), the programmer does not need to be aware of the fault-tolerant capabilities of the machine.
6. At a hardware level, fault tolerance is achieved by duplexing each hardware component. Disks are mirrored.
7. Multiple processors are “lock-stepped” together and their outputs are compared for correctness.
8. When an anomaly occurs, the faulty component is determined and taken out of service, but the machine continues to function as usual.

9. A fault-tolerant design enables a system to continue its intended operation, possibly at a reduced level, rather than failing completely, when some part of the system fails.
10. The term is most commonly used to describe computer systems designed to continue more or less fully operational with, perhaps, a reduction in throughput or an increase inresponse time in the event of some partial failure.
11. That is, the system as a whole is not stopped due to problems either in the hardware or the software.
12. An example in another field is a motor vehicle designed so it will continue to be drivable if one of the tires is punctured.
13. A structure is able to retain its integrity in the presence of damage due to causes such as fatigue, corrosion, manufacturing flaws, or impact.
14. Within the scope of an individual system, fault tolerance can be achieved by anticipating exceptional conditions and building the system to cope with them, and, in general, aiming for self-stabilization so that the system converges towards an error-free state.
15. However, if the consequences of a system failure are catastrophic, or the cost of making it sufficiently reliable is very high, a better solution may be to use some form of duplication.

Faults can be classified into one of three categories

1. Transient faults

These occur once and then disappear. For example, a network message doesn’t reach its destination but does when the message is retransmitted.

2. Intermittent faults

Intermittent faults are characterized by a fault occurring, then vanishing again, then reoccurring, then vanishing … These can be the most annoying of component faults. A loose connection is an example of this kind of fault.

3. Permanent faults

This type of failure is persistent: it continues to exist until the faulty component is repaired or replaced. Examples of this fault are disk head crashes, software bugs, and burnt-out power supplies.

Reliability and Availability Techniques

- Two approaches to increasing system reliability are fault avoidance and fault tolerance.
- Fault avoidance results from conservative design practices such as the use of high-reliability parts.
- Though the goal of fault avoidance is to reduce the likelihood of failure, even after the most careful application of fault-avoidance techniques, failures will occur eventually owing to defects in the system.
- In comparison to this approach, fault tolerance appears much better, as fault tolerance approaches the system design with the assumptions that defects would very much likely surface any way during system operational stage, so that the design is orientated towards making the system keep operating correctly with the presence of defects.
- Redundancy is a very classic technique used in both fault avoidance and fault tolerance approaches.

- With the redundancy technique a system could highly likely pass the ten fault response stages listed above.