PART 1

Identify all aspects of the university requiring information to be stored about, otherwise called entities in relational database management systems. You are expected to come up with no lesser than eight entities. These are some of the design rules taken into consideration.

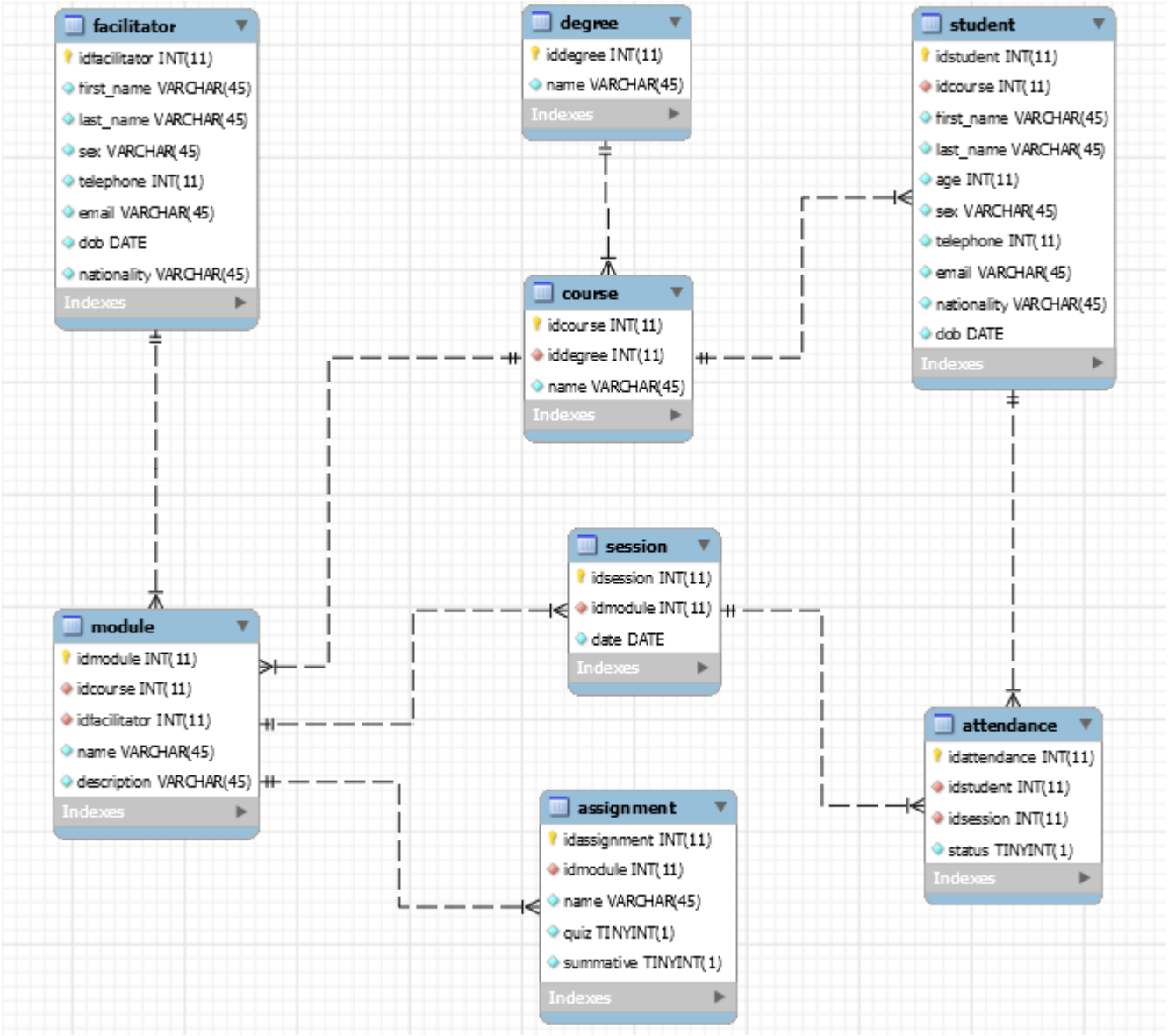
- The university has several accredited degrees.
- The degree offers several courses.
- There are modules taken in the courses enrolled by students.
- A student can only be in one course of study, thus pursuing BSc in Computer Science or BA in Global Challenges.
- Modules have assignments.
- Modules happen during class sessions.
- Attendance is taken during class sessions.
- A facilitator instructs a module.

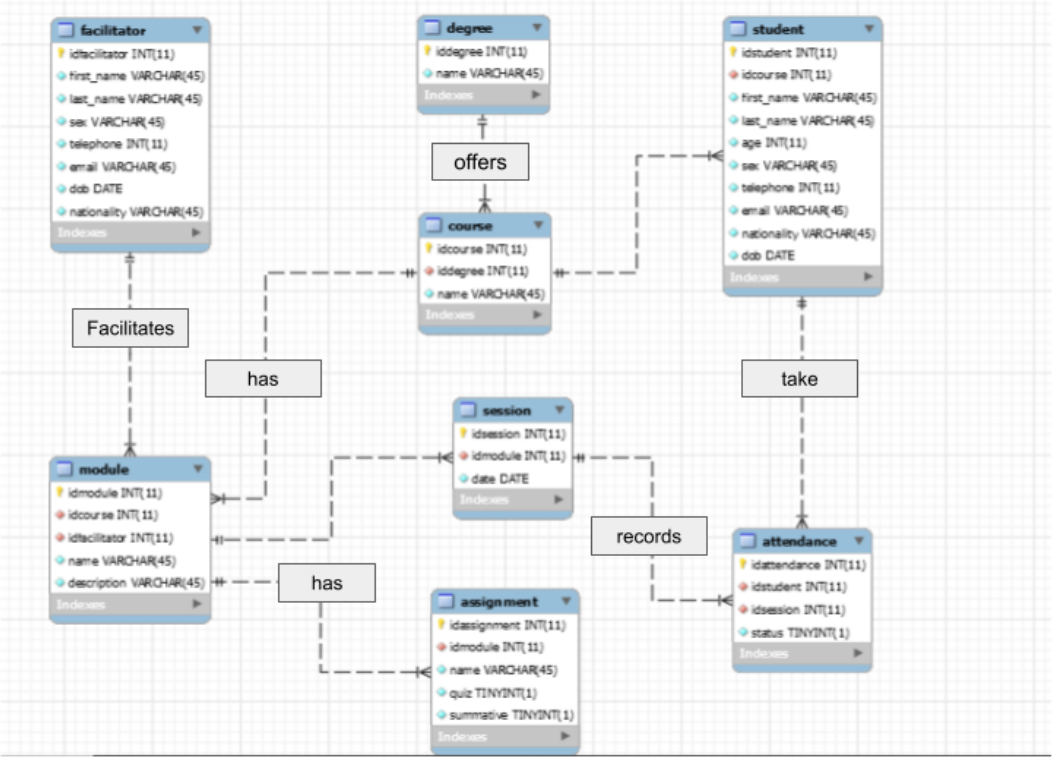
These are the essential entities captured by our ERD design, and they store general information of the particular module.

- Degree
- Student
- Facilitator
- Course
- Module
- Session
- Attendance
- Assignments

The next is to establish how the various objects are related in your conceptual model, you will be required to map out any relationships between the entities, appropriately using foreign keys to indicate your relationships. Any many to many relationships and any other conflicts should be resolved within your ER diagram through normalization. Attributes for each entity should be clear and meaningful, properly denoting primary

keys and foreign keys within your diagram. (Logical model). The below ERD has the information about the attributes, and their relationships through foreign keys.





You then will go ahead and implement the E.R diagram resulting from the stage 2, (Qn 2), your implementation should mirror your logical design diagram, you will then populate your database with no lesser than 10 records for each entity. You will create at least three users for your database, granting them varying privileges of access to your database.

Please find my script here: [MySql Script in Piaza.io]

Registrar Roles and Privileges

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, FILE, INDEX, ALTER,
CREATE TEMPORARY TABLES, EXECUTE, CREATE VIEW, SHOW VIEW, CREATE ROUTINE,
ALTER ROUTINE, EVENT, TRIGGER ON *.* TO 'ALU_Registrar'@'%' IDENTIFIED BY
PASSWORD '*39F225903919EC944FC2A73B775A7E5A77A8A83F';

GRANT ALL PRIVILEGES ON `alu\_registrar\_%`.* TO 'ALU_Registrar'@'%';
```

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE
TEMPORARY TABLES, EXECUTE, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER
ROUTINE, EVENT, TRIGGER ON `advanced_db[summative]`.* TO 'ALU_Registrar'@'%';
```

Staff Roles and Privileges

```
GRANT SELECT, INSERT, UPDATE, DELETE, FILE ON *.* TO 'ALU_staff'@'%'
IDENTIFIED BY PASSWORD '*FBEE7A343651776059279BF10B97495ED9FB2DDC';

GRANT ALL PRIVILEGES ON `alu\_staff\_%`.* TO 'ALU_staff'@'%';

GRANT SELECT, INSERT, UPDATE, DELETE ON `advanced_db[summative]`.* TO
'ALU_staff'@'%';
```

Admin Roles and Privileges

```
GRANT ALL PRIVILEGES ON *.* TO 'Alu_admin'@'%' IDENTIFIED BY PASSWORD
'*6E573A3ADD6A4A5A2443EB5173C7635DF1AB015F' WITH GRANT OPTION;

GRANT ALL PRIVILEGES ON `alu\_admin\_%`.* TO 'Alu_admin'@'%';

GRANT ALL PRIVILEGES ON `advanced_db[summative]`.* TO 'Alu_admin'@'%' WITH
GRANT OPTION;
```

PART 2:

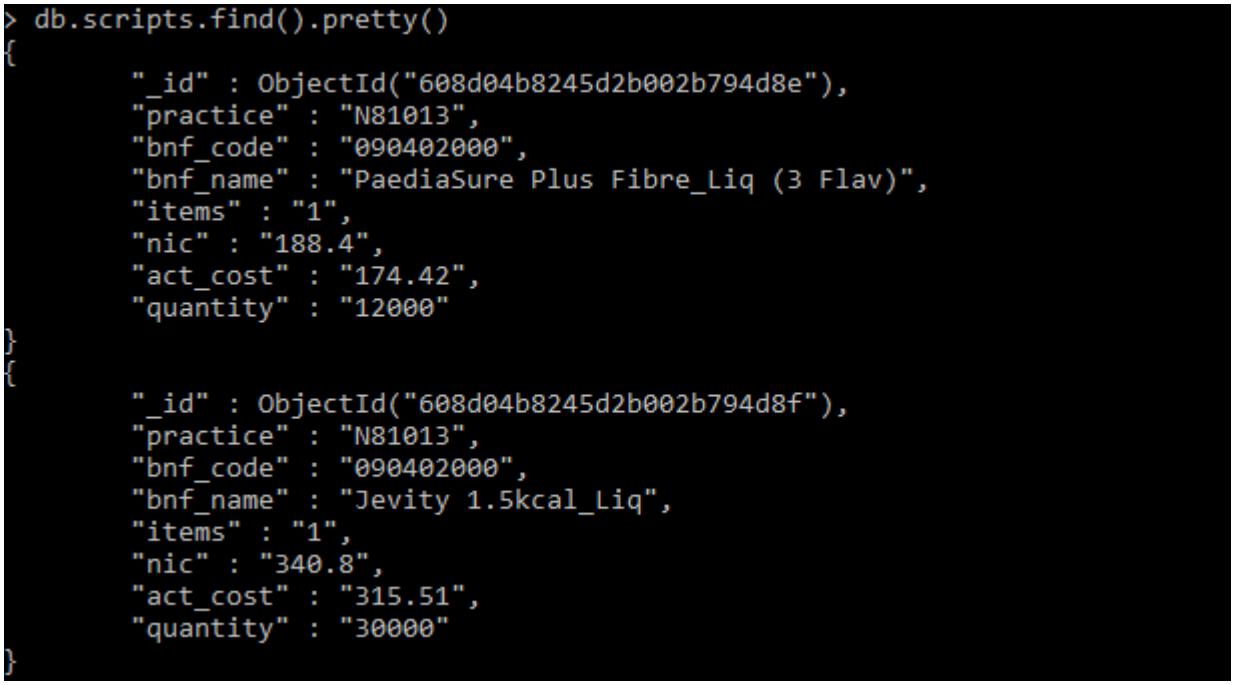
Question 1: summary_statistics

Calculate the total, mean, standard deviation, and quartile statistics of the 'items', 'quantity'', 'nic', and 'act_cost' fields. To do this we can write some functions to calculate the statistics and apply the functions to our data structure. But DataFrame has a describe method that will calculate most (not all) of these things for you.

Submit the summary statistics to the grader as a list of tuples: [('act_cost', (total, mean, std, q25, median, q75)), ...]

```
use AdvancedDB_Summative
show collections
db.scripts.find({}).pretty()
```

_id	practice	bnf_code	bnf_name	items	nic	act_cost	quantity
608d04b8245d2...	N81013	1001010K0	Indometacin_C...	2	8.54	7.93	196
608d04b8245d2...	N81013	1001010N0	Mefenamic Aci...	1	18.45	17.09	84
608d04b8245d2...	N81013	1001010P0	Naproxen_Tab ...	73	145.78	135.97	4389
608d04b8245d2...	N81013	1001010P0	Naproxen_Tab ...	39	113.09	105.49	2230
608d04b8245d2...	N81013	1001010P0	Naproxen_Tab ...	2	8.91	8.27	168
608d04b8245d2...	N81013	1001010P0	Naproxen_Tab ...	1	7.01	6.5	56
608d04b8245d2...	N81013	1001010P0	Naprosyn EC_T...	2	17.12	15.88	112
608d04b8245d2...	N81013	1001010X0	Nabumetone_...	1	7.53	6.98	56
608d04b8245d2...	N81013	1001022K0	Methylpred Ac...	7	24.08	22.38	7
608d04b8245d2...	N81013	1001022K0	Methylpred Ac...	5	30.9	28.66	5
608d04b8245d2...	N81013	1001022K0	Methylpred Ac...	5	44.8	41.54	5
608d04b8245d2...	N81013	1001022K0	Methylpred Ac...	2	7.88	7.32	2



```
> db.scripts.find().pretty()
{
        "_id" : ObjectId("608d04b8245d2b002b794d8e"),
        "practice" : "N81013",
        "bnf_code" : "090402000",
        "bnf_name" : "PaediaSure Plus Fibre_Liq (3 Flav)",
        "items" : "1",
        "nic" : "188.4",
        "act_cost" : "174.42",
        "quantity" : "12000"
}
{
        "_id" : ObjectId("608d04b8245d2b002b794d8f"),
        "practice" : "N81013",
        "bnf_code" : "090402000",
        "bnf_name" : "Jevity 1.5kcal_Liq",
        "items" : "1",
        "nic" : "340.8",
        "act_cost" : "315.51",
        "quantity" : "30000"
}
```

Total Counts

Items

```
db.scripts.aggregate( [{$count:"items" }])
//Ans:
{
    "items" : NumberInt(973193)
}
```

Quantity

```
db.scripts.aggregate( [{$count:"quantity" }])
//Ans:
{
    "quantity" : NumberInt(973193)
}
```

Nic

```
db.scripts.aggregate( [{$count:"nic" }])
//Ans:
{
    "nic" : NumberInt(973193)
```

```
}
```

Act_cost

```
db.scripts.aggregate( [{$count:"act_cost" }])
//Ans:
{
    "acr_cost" : NumberInt(973193)
}
```

Total Sum

Items

```
db.scripts.aggregate([ { $match: {} }, { $group: { _id : null, sum : { $sum: "$items" } } }])
//Ans:
{
    "_id" : null,
    "sum" : NumberInt(8888304)
}
```

Quantity

```
db.scripts.aggregate([ { $match: {} }, { $group:{ _id : null, sum : { $sum: "$quantity" } }}])

//Ans:
{
    "_id" : null,
    "sum" : NumberInt(721457006)
}
```

Nic

```
db.scripts.aggregate([ { $match: {} }, { $group:{ _id : null, sum : { $sum: "$nic" } }}])

//Ans:
{
    "_id" : null,
    "sum" : 71100424.84
}
```

Act_cost

```
db.scripts.aggregate([ { $match: {} }, { $group:{ _id : null, sum : { $sum: "$act_cost" } }}])

//Ans:
{
    "_id" : null,
    "sum" : 66164096.12
}
```

Mean

Items

```
db.scripts.aggregate([ { $match: {} }, { $group: { _id : null, avg : { $avg: "$items" } } }])
//Ans:
{
    "_id" : null,
    "avg" : 9.133135976111625
}
```

Quantity

```
db.scripts.aggregate([ { $match: {} }, { $group:{ _id : null, avg : { $avg: "$quantity" } }}])

//Ans:
{
    "_id" : null,
    "avg" : 741.3298348837282
}
```

Nic

```
db.scripts.aggregate([ { $match: {} }, { $group:{ _id : null, avg : { $avg: "$nic" } }}])

//Ans:
{
    "_id" : null,
    "avg" : 73.05891517920907
}
```

Act_cost

```
db.scripts.aggregate([ { $match: {} }, { $group:{ _id : null, avg : { $avg: "$act_cost" } }}])

//Ans:
{
    "_id" : null,
    "avg" : 67.98661326170657
}
```

Standard Deviation

Items

```
db.scripts.aggregate([ { $match: {} }, { $group:{ _id : null, stv : { $stdDevSamp: "$items" } }}])

//Ans:
{
    "_id" : null,
    "stv" : 29.204198282803787
}
```

Quantity

```
db.scripts.aggregate([ { $match: {} }, { $group:{ _id : null, stv : { $stdDevSamp: "$quantity" } }}])

//Ans:
{
    "_id" : null,
    "stv" : 3665.426958468006
}
```

Nic

```
db.scripts.aggregate([ { $match: {} }, { $group:{ _id : null, stv : { $stdDevSamp: "$nic" } }}])

//Ans:
{
    "_id" : null,
    "stv" : 174.40170332302165
}
```

Act_cost

```
db.scripts.aggregate([ { $match: {} }, { $group:{ _id : null, stv : { $stdDevSamp: "$act_cost" }}}])

//Ans:
{
    "_id" : null,
    "stv" : 174.40170332302165
}
```

Quartile

Items

```
db.scripts.aggregate([
  { $sort: { items: 1 } },
  { $group: { _id: null, items: { $push: "$items" } } },
  {
    $project: {
      _id: "Items Quartiles",
      "25th": {
        $arrayElemAt: [
          "$items", { $floor: { $multiply: [0.25, { $size: "$items" }] } },
        ],
      },
      "50th": {
        $arrayElemAt: [
          "$items", { $floor: { $multiply: [0.5, { $size: "$items" }] } },
        ],
      },
      "75th": {
        $arrayElemAt: [
          "$items", { $floor: { $multiply: [0.75, { $size: "$items" }] } },
        ],
      },
    },
  },
],{allowDiskUse: true});

//Ans:
{
    "_id" : "Items Quartiles",
    "25th" : 1.0,
    "50th" : 2.0,
    "75th" : 6.0
}
```

Quantity

```
db.scripts.aggregate([
  { $sort: { quantity: 1 } },
  { $group: { _id: 0, quantity: { $push: "$quantity" } } },
  {
    $project: {
      _id: "quantity Quartiles",
      "25th": {
        $arrayElemAt: [
          "$quantity", { $floor: { $multiply: [0.25, { $size: "$quantity" }] } },
        ],
      },
      "50th": {
        $arrayElemAt: [
          "$quantity", { $floor: { $multiply: [0.5, { $size: "$quantity" }] } },
        ],
      },
      "75th": {
        $arrayElemAt: [
          "$quantity", { $floor: { $multiply: [0.75, { $size: "$quantity" }] } },
        ],
      },
    },
  },
],{allowDiskUse: true});

//Ans:
{
    "_id" : "quantity Quartiles",
    "25th" : 28.0,
    "50th" : 100.0,
    "75th" : 350.0
}
```

Nic

```
db.scripts.aggregate([
  { $sort: { nic: 1 } },
  { $group: { _id: 0, nic: { $push: "$nic" } } },
  {
    $project: {
      _id: "nic Quartiles",
      "25th": {
        $arrayElemAt: [
          "$nic", { $floor: { $multiply: [0.25, { $size: "$nic" }] } },
        ],
      },
      "50th": {
        $arrayElemAt: [
          "$nic", { $floor: { $multiply: [0.5, { $size: "$nic" }] } },
        ],
      },
      "75th": {
        $arrayElemAt: [
          "$nic", { $floor: { $multiply: [0.75, { $size: "$nic" }] } },
        ],
```

```
      },
    },
  },
],{allowDiskUse: true});

//Ans:
{
    "_id" : "nic Quartiles",
    "25th" : 7.8,
    "50th" : 22.64,
    "75th" : 65.0
}
```

Act_cost

```
db.scripts.aggregate([
  { $sort: { act_cost: 1 } },
  { $group: { _id: 0, act_cost: { $push: "$act_cost" } } },
  {
    $project: {
      _id: "act_cost Quartiles",
      "25th": {  $arrayElemAt: [   "$act_cost",
          { $floor: { $multiply: [0.25, { $size: "$act_cost" }] } } },
      ]},
      "50th": {  $arrayElemAt: [   "$act_cost",
          { $floor: { $multiply: [0.5, { $size: "$act_cost" }] } } },
      ]},
      "75th": {  $arrayElemAt: [   "$act_cost",
          { $floor: { $multiply: [0.75, { $size: "$act_cost" }] } } },
      ]},
    },
  },
],{allowDiskUse: true});


//Ans:
{
    "_id" : "act_cost Quartiles",
    "25th" : 7.33,
    "50th" : 21.22,
    "75th" : 60.67
}
```

Question 2: most_common_item

```
db.getCollection("scripts").aggregate([
  {
    $group: {  _id: "$bnf_name",  bnf_name: { $first: "$bnf_name" }, total: { $sum: "$items" },
    },
  },
```

```
  { $sort: { total: -1 } },
  { $group: { _id: 0, item: { $first: "$$ROOT" } } },
  { $replaceRoot: { newRoot: "$item" } },
  { $project: { _id: 0, bnf_name: 1, total: 1 } },
],{allowDiskUse: true});

//Ans:
{
    "bnf_name" : "Omeprazole_Cap E/C 20mg",
    "total" : 218583.0
}
```

PART 3

Critically discuss distributed, parallel and centralized computing architectures in terms of performance, scalability, security, among other factors. Areas where each architecture can be applied for optimized performance. Your discussion should lean towards data storage and handling but issues like load balancing among other factors should also be addressed.

Centralized Computing Architecture

Intro

Central computing architecture is characterized by a central server where all computing is performed for clients(nodes) connected to the server. The architecture ensures the deployment of main servers' resources, management, and administration. The main central server is responsible for delivering computing resources, processes, and calculations to the nodes(clients) attached [1]. It is mostly employed in single-player games, personal computers, and testing application deployment.

Performance

The architecture allows you to provide users a consistent experience because the nodes are directly connected to the server. Data packets and requests follow a shooter path that optimizes performances. However, the performance could be faulted when there are too many requests or Distributed attacks where attackers write scripts to request the server multiple times [3].

Scalability

Scalability in Centralised architecture allows vertical scaling of certain limits. If the limit is exceeded even if you increase hardware and software resources, the performance will increase optimally; hence its cost ratio is less than 1. It can't allow horizontal scaling because it will refute the central server(master) characteristics since we will have more central servers [2].

Security

The architecture is easy to physically and digitally secure since everything is centralized and all security resources are channeled into protecting one central server. However, the architecture attracts many attackers since when you breach the security of the central server, you gain access to all assets and data stored in the server or traveling through the network [3].

Distributed Computing Architecture

Intro

Distributed Computing Architecture is designed in such a way that components in the architecture can cooperate with each other to achieve a specific goal. The end-users of the system architecture will always feel the results produced are a single coherent system like how Google search works. The architecture is characterized by concurrently running of components, no global clock, and components "fail independently of each other" [4]. In lame man terms distributed computing architecture, multiple computers are connected across a network to work together on large workloads and utilize resources provided by components connected in the network. It is mostly employed in multiplayer games and search engines like google. Data is processed on multiple servers leveraging their computing power.

Performance

The Distributed system has a better performance than centralized systems since they do not rely on only one server. They simply remove the risk of having a single point of failure. Additionally, since they do not rely on servers, they have low latency since most of the servers are geographically distributed, leading to less response time. For the system to be optimized efficiently, the same amount of work is distributed among components in the system. Load balancers can balance the workload statically at compile time if different task sizes are known in advance since the task assigned different processors dynamically at run time [5].

Scalability

Distributed Computing architecture can both be scaled both vertically and horizontally and not contradicting the system like centralized systems. It is much easier to scale employing the scale-out architecture in which higher loads can be handled by adding new hardware or resources to the system than replacing existing hardware. The computing system can be scaled with respect to geography, size, or administration [5].

Security

Distributed Architecture is more susceptible to external attacks compared to centralized systems. It is most difficult because individual components need to be secured in the network, and the network must be secured. Weak encryption can lead attackers to tap into communication made between different systems. Fluctuating resource consumption is a problem in the security of distributed systems. A spike of resources in the architecture is not always a security threat but a legitimate spike of resources used by users.

Parallel Computing Architecture

The computing design depends on parallel hardware with multiple components like processors to execute a program or data of different parts at the same time. Parallel computing is highly efficient and performs parallel data processing when co-located; thus, they are very close to each other in the data center. All calculations and processes are executed simultaneously. The parallel architecture can be scaled vertically and horizontally. There is shared storage for processes and shared memory

to synchronize threads. It is mostly employed in multi-layered applications like medical imaging and financial risk management software. However, maximum performance is bottlenecked when serial computations must be performed; hence the optimized performance is bottlenecked [7].

References

[1]"What is Centralized Computing? - Definition from Techopedia", *Techopedia.com*, 2021. [Online]. Available: https://www.techopedia.com/definition/26507/centralized-computing#:~:text=Centralized%20computing%20is%20a%20type,performed%20on%20a%20central%20server.&text=The%20central%20server%2C%20in%20turn,to%20the%20attached%20client%20machines. [Accessed: 01- May- 2021].

[2]"Comparison - Centralized, Decentralized and Distributed Systems - GeeksforGeeks", *GeeksforGeeks*, 2021. [Online]. Available: https://www.geeksforgeeks.org/comparison-centralized-decentralized-and-distributed-systems/. [Accessed: 01- May- 2021].

[3]"Centralized vs decentralized networks: What's the difference? - Loki", *Loki*, 2021. [Online]. Available: https://loki.network/2019/12/05/centralized-vs-decentralized-networks/. [Accessed: 01- May- 2021].

[4]"What is a Distributed System? How a Distributed System Works", *Articles for Developers Building High Performance Systems*, 2021. [Online]. Available: https://blog.stackpath.com/distributed-system/. [Accessed: 01- May- 2021].

[5]"Efficiency", *Cs.stanford.edu*, 2021. [Online]. Available: https://cs.stanford.edu/people/eroberts/courses/soco/projects/distributed-computing/html/body_efficiency.html. [Accessed: 01- May- 2021].

[6]"Save Time and Improve your Marks with CiteThisForMe, The No. 1 Citation Tool", Cite This For Me, 2021. [Online]. Available: https://www.citethisforme.com/. [Accessed: 01- May- 2021].

[7]2021. [Online]. Available: https://webcache.googleusercontent.com/search?q=cache:pQjVaK7e4hYJ:https://byu.instructure.com/courses/1455/files/296535/download+&cd=14&hl=en&ct=clnk&gl=rw. [Accessed: 01- May- 2021].