188.427 VU E-Commerce 2016WS

Patrick Sommer 0925011, Julia Filler 1225408, Constantin Brîncoveanu 1225561

Baseline

Selected Baseline: Retry

- improvements:
  - request will be tried to send up to 5 times
  - > 5 times - the request will be send to another edge in the near of the user

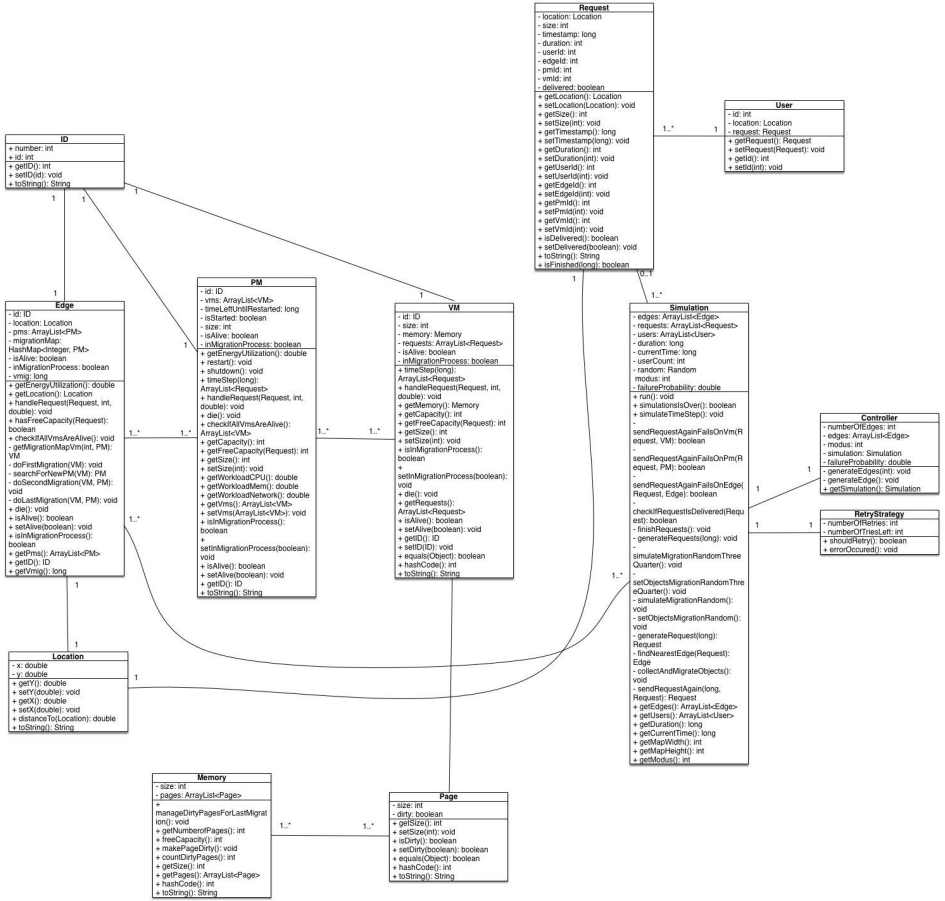
Selected programming language: Java

Developments

Modes:

- no retry - migrate random
- no retry - migrate random + SLA (ensure that there is enough time to migrate) and migrate if PM is 3/4 full
- retry - migrate random
- retry - migrate random + SLA (ensure that there is enough time to migrate) and migrate if PM is 3/4 full

UML


Request
- location: Location
- size: int
- timestamp: long
- duration: int
- userId: int
- edgeId: int
- pmId: int
- vmId: int
- delivered: boolean
+ getLocation(): Location
+ setLocation(Location): void
+ getSize(): int
+ setSize(int): void
+ getTimestamp(): long
+ setTimestamp(long): void
+ getDuration(): int
+ setDuration(int): void
+ getUserId(): int
+ setUserId(int): void
+ getEdgeId(): int
+ setEdgeId(int): void
+ getPmId(): int
+ setPmId(int): void
+ getVmId(): int
+ setVmId(int): void
+ isDelivered(): boolean
+ setDelivered(boolean): void
+ toString(): String
+ isFinished(long): boolean
User
- id: int
- location: Location
- request: Request
+ getRequest(): Request
+ setRequest(Request): void
+ getId(): int
+ setId(int): void
ID
+ number: int
+ id: int
+ getID(): int
+ setID(id): void
+ toString(): String
Edge
- id: ID
- location: Location
- pms: ArrayList<PM>
- migrationMap: HashMap<Integer, PM>
- isAlive: boolean
- inMigrationProcess: boolean
- vmIq: long
+ getEnergyUtilization(): double
+ getLocation(): Location
+ handleRequest(Request, int, double): void
+ hasFreeCapacity(Request): boolean
+ checkIfAllVmsAreAlive(): void
- getMigrationMapVm(int, PM): VM
- doFirstMigration(VM): void
- searchForNewPM(VM): PM
- doSecondMigration(VM, PM): void
- doLastMigration(VM, PM): void
+ die(): void
+ isAlive(): boolean
+ setAlive(boolean): void
+ isInMigrationProcess(): boolean
+ getPms(): ArrayList<PM>
+ getID(): ID
+ getVmiq(): long
PM
- id: ID
- vms: ArrayList<VM>
- timeLeftUntilRestarted: long
- isStarted: boolean
- size: int
- isAlive: boolean
- inMigrationProcess: boolean
+ getEnergyUtilization(): double
+ restart(): void
+ shutdown(): void
+ timeStep(long): ArrayList<Request>
+ handleRequest(Request, int, double): void
+ die(): void
+ checkIfAllVmsAreAlive(): ArrayList<VM>
+ getCapacity(): int
+ getFreeCapacity(Request): int
+ getSize(): int
+ setSize(int): void
+ getWorkloadCPU(): double
+ getWorkloadMem(): double
+ getWorkloadNetwork(): double
+ getVms(): ArrayList<VM>
+ setVms(ArrayList<VM>): void
+ isInMigrationProcess(): boolean
+ setInMigrationProcess(boolean): void
+ isAlive(): boolean
+ setAlive(boolean): void
+ getID(): ID
+ toString(): String
VM
- id: ID
- size: int
- memory: Memory
- requests: ArrayList<Request>
- isAlive: boolean
- inMigrationProcess: boolean
+ timeStep(long): ArrayList<Request>
+ handleRequest(Request, int, double): void
+ getMemory(): Memory
+ getCapacity(): int
+ getFreeCapacity(Request): int
+ getSize(): int
+ setSize(int): void
+ isInMigrationProcess(): boolean
+ setInMigrationProcess(boolean): void
+ die(): void
+ getRequests(): ArrayList<Request>
+ isAlive(): boolean
+ setAlive(boolean): void
+ getID(): ID
+ setID(ID): void
+ equals(Object): boolean
+ hashCode(): int
+ toString(): String
Simulation
- edges: ArrayList<Edge>
- requests: ArrayList<Request>
- users: ArrayList<User>
- duration: long
- currentTime: long
- userCount: int
- random: Random
- modus: int
- failureProbability: double
+ run(): void
+ simulationIsOver(): boolean
+ simulateTimeStep(): void
- sendRequestAgainFailsOnVm(Request, VM): boolean
- sendRequestAgainFailsOnPm(Request, PM): boolean
- sendRequestAgainFailsOnEdge(Request, Edge): boolean
- checkIfRequestIsDelivered(Request): boolean
- finishRequests(): void
- generateRequests(long): void
- simulateMigrationRandomThreeQuarter(): void
- setObjectsMigrationRandomThreeQuarter(): void
- simulateMigrationRandom(): void
- setObjectsMigrationRandom(): void
- generateRequest(long): Request
- findNearestEdge(Request): Edge
- collectAndMigrateObjects(): void
- sendRequestAgain(long, Request): Request
+ getEdges(): ArrayList<Edge>
+ getUsers(): ArrayList<User>
+ getDuration(): long
+ getCurrentTime(): long
+ getMapWidth(): int
+ getMapHeight(): int
+ getModus(): int
Controller
- numberOfEdges: int
- edges: ArrayList<Edge>
- modus: int
- simulation: Simulation
- failureProbability: double
- generateEdges(int): void
- generateEdge(): void
+ getSimulation(): Simulation
RetryStrategy
- numberOfRetries: int
- numberOfTriesLeft: int
+ shouldRetry(): boolean
+ errorOccured(): void
Location
- x: double
- y: double
+ getY(): double
+ setY(double): void
+ getX(): double
+ setX(double): void
+ distanceTo(Location): double
+ toString(): String
Memory
- size: int
- pages: ArrayList<Page>
+ manageDirtyPagesForLastMigration(): void
+ getNumberofPages(): int
+ freeCapacity(): int
+ makePageDirty(): void
+ countDirtyPages(): int
+ getSize(): int
+ getPages(): ArrayList<Page>
+ hashCode(): int
+ toString(): String
Page
- size: int
- dirty: boolean
+ getSize(): int
+ setSize(int): void
+ isDirty(): boolean
+ setDirty(boolean): boolean
+ equals(Object): boolean
+ hashCode(): int
+ toString(): String

Failure handling

```
if(modus == 3 || modus == 4){   failure = Math.random() < failureProbability; }
```

request check if it is delivered: true

request check if it is delivered: false

request modus fails on PM

no success!

request modus fails on VM

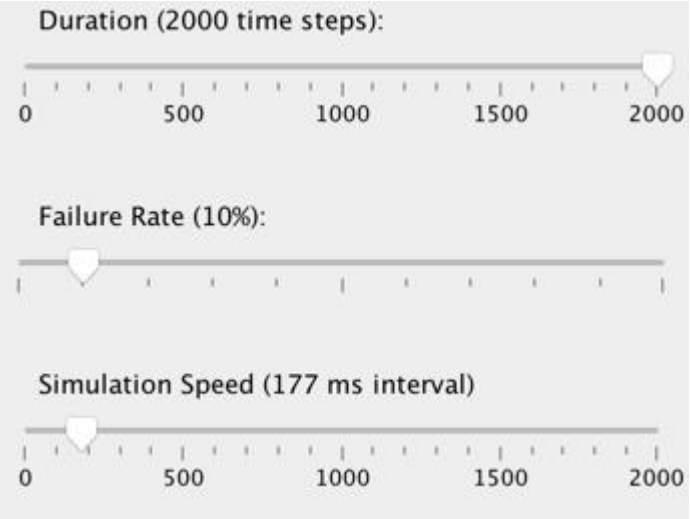
Success!!

SLA's

- Agreed VM characteristics
- Users (requests) Total
- Users satisfied
- Pending requests
- Avg. Requests per Time Step
- Total Energy Consumption
- Average Latency
- Vmig
- Dirty pages rate
- Memory

Results - Settings

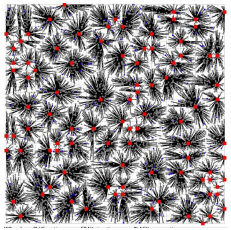
- 3000 VMs, 1000 PMs running in 100 edges
- Duration 2000 (5,04 Min/step = 10080 min = 1 week) time steps
- Failure rate 10%
- Simulation Speed 177 ms intervall
- 3 runs



Results

	no retry - migrate random	**no retry - migrate random + SLA**
Run 1	PM is full after a few steps (126), after round 2000 migrations, Vmig = 10-550 Users (Requests) Total: 3834 Users Satisfied: 3614 (94%) Pending Requests: 220 Avg. Requests per Time Step: 30.0 Total Energy Consumption: 924725 Average Latency: 86ms Vmig: 14375 Dirty Pages: 3221 Memory: 240100	steps 38, round 2229 migrations, Vmig = 5 - 270 Users (Requests) Total: 1465 Users Satisfied: 1164 (79%) Pending Requests: 301 Avg. Requests per Time Step: 30.0 Total Energy Consumption: 529625 Average Latency: 85ms Vmig: 7325 Dirty Pages: 1465 Memory: 221800
Run 2	PM is full after a few steps (137), after round 2050 migrations, Vmig = 10-780 Users (Requests) Total: 3935 Users Satisfied: 3583 (91%) Pending Requests: 352 Avg. Requests per Time Step: 30.0 Total Energy Consumption: 957350 Average Latency: 85ms Vmig: 13550 Dirty Pages: 3366 Memory: 235150	steps 37, round 2185 migrations, Vmig = 5 - 110 Users (Requests) Total: 734 Users Satisfied: 424 (58%) Pending Requests: 310 Avg. Requests per Time Step: 30.0 Total Energy Consumption: 365375 Average Latency: 80ms Vmig: 3660 Dirty Pages: 735 Memory: 186450
Run 3	PM is full after a few steps (128), after round 2036 migrations, Vmig = 10-665 Users (Requests) Total: 3834 Users Satisfied: 3504 (91%) Pending Requests: 330 Avg. Requests per Time Step: 30.0 Total Energy Consumption: 939125 Average Latency: 82ms Vmig: 12410 Dirty Pages: 3285 Memory: 228600	steps 35, round 2410 migrations, Vmig = 5 - 205 Users (Requests) Total: 1096 Users Satisfied: 848 (77%) Pending Requests: 248 Avg. Requests per Time Step: 30.0 Total Energy Consumption: 446600 Average Latency: 89ms Vmig: 5480 Dirty Pages: 1096 Memory: 204350

Results

	retry - migrate random	**retry - migrate random + SLA**
Run 1	PM is full after 14 steps, after round 334 migrations, 39 fails, 90 successful, Vmig = 10-210	no termination, Vmig = 5 - 1280
Run 2	PM is full after 17 steps, after round 445 migrations, 41 fails, 95 successful, Vmig = 10-90	Users (Requests) Total: 18875 Users Satisfied: 18534 (98%) Pending Requests: 341 Avg. Requests per Time Step: 30.0 Total Energy Consumption: 2678150 Average Latency: 40ms Vmig: 54595 Dirty Pages: 11014 Memory: 684200 
Run 3	PM is full after 30 steps, after round 654 migrations, 58 fails, 115 successful, Vmig = 10-90	

Result Analysis

- It was not possible to do all modes until one week regarding the termination
  - no retry - migrate random performs better than no retry - migrate random + SLA regarding performed steps BUT not Vmig
- SLA is important in combination with Retry → modus 4: best results
- retry - migrate random + SLA: only 1 Run, unbelievable values
- retry - migrate random: Run 3 → outlier, Relationship failures - successful better if the simulation time is higher

FINAL Analysis

- **Great improvement:** stop after five times, and find an other resource
- If no Retry was used, less migrations were performed
- SLA is important in combination with Retry → best results
- The use of SLAs greatly increased the number of fails
- Interesting observation: Failure rate = 100% → modus 4 is terminating

DEMO

Thank your for your attention