# Notes Weekly Meeting PA ContMigration

|  |
| --- |
| **Weekly meeting 20.12.2023 12:00 – 13:00**  Attendance: Guerkan, Wissem, Rinchen, Tony |

|  |  |
| --- | --- |
| Admin | * \*\*Meeting 20.12.23\*\* * What we did since last meeting: * - * --- * Questions: * - OT: SWS1 joker? * --- * Notes * - keep data close lmao * - draw histogram distribution * - cpu load * - reason for fluctuation? * - bias? * - docker pull? * - download of image? * - explain them * - have images locally? * - what kind of delay? * - * - rather focus on how to analize the given data * - finish paper * - chapter * - xmention criu in background/discussion * - explain tmp, networkfilesystem before giving steps * - xmove steps to methods * - maybe use computersymbols * - export figures as pdf! * - quality * - also for plots! * - xrename title "containers" * - benchmarking applications * - "this script" * - highlevel description whats this app first * - then explain * - mention NetworkIO in discussion * - killfirst simulate criu * - put the shell scripts to appendix/annex/or GITHUB!!! * --- * Next steps: |

|  |  |  |
| --- | --- | --- |
| **Weekly meeting 13.12.2023 12:00 – 13:00**  Attendance: Guerkan, Wissem, Rinchen, Tony | | |
| **Topic** | **Discussion** | **Deadline** |

|  |  |
| --- | --- |
| Admin | * \*\*Meeting 13.12.23\*\* * What we did since last meeting: * - found our way of migrating the statefulsets * - for redis * - mongodb * - had a meeting on monday with Wissem * - discussion about our way of migrating * - setting up a real nfs as storageclass * - meeting yesterday again with Wissem * - where we followed the procedure on complesis to migrate together * - turns out that memory state is not copied * - conclusion * - -> the statefulset migration is not copying a state of a container! * - * --- * Questions: * - * - Offtopic BA * - if we decide to do it, how should we proceed? * --- * Notes: * - WISSEM * - external client in podman * - script that read/writes * - make migration while script is running * - check integrity * - pod down time * - script downtime * - with this method * - we can measure 2 types of migration * - simulate comparison * - between * - inverted shell * - simulate proxy/loadbalancer * - signal approach * - sh script * --- * Next steps: * - finish the statefulset migrations * - focus on redis by tmr * - mongodb by sat * - cpu if there is time left * - find metrics to measure * - monitor migrations * - at least for the migrations we have * - finish with paper * - literature * - write texts * --- * --- |

|  |  |  |
| --- | --- | --- |
| **Weekly meeting 6.12.2023 12:00 – 13:00**  Attendance: Guerkan, Wissem, Rinchen, Tony | | |
| **Topic** | **Discussion** | **Deadline** |

|  |  |
| --- | --- |
| Admin | * \*\*Meeting 6.12.23\*\* * What we did since last meeting: * - started paper * - first version of structure, layout * - some figures * - see appendix * - PA\_HS\_2023\_kolodrin\_maant001-4.pdf * - note that the title page and declaration of originality need to be updated * - started on the statefulset migrations * - cpu * - need to write the state out so script knows where to continue after migration * - mongodb * - redis * - we are able to dump the redis db state in cluster1 and load it again from cluster2 after migrating * - as for both dbs * - after migrating to other cluster * - it connects again to/loads from the storage and can access the data from before * - but still need to check if the scripts continue from where they left off * - as in, we say "write 10000 values into the db", when we stop the container at value x, so that it starts at x+1 * - created some scripts to automate container * - setup * - teardown * - migration * --- * Questions: * - as for the formal presentation of our report, would the week of january 29th be ok for you? * - can you check the current version of the layout? * - Offtopic BA: will you inform us when you upload your BA description? * - @Wissem: we are struggling to setup the CRIU migrations/snapshotting thing. Would you be able to find some time to sit down with us? * --- * Next steps: * - continue with paper * - check literature again * - write texts * - have a first version ready until next meeting * - finish the statefulset migrations * - cpu * - mongodb * - redis * - (networkIO?) * - might not be able to * - find metrics to measure * - monitor migrations * - at least for the migrations we have * - try CRIU migrations * --- |

|  |  |  |
| --- | --- | --- |
| **Weekly meeting 29.11.2023 12:00 – 13:00**  Attendance: Guerkan, Wissem, Rinchen, Tony | | |
| **Topic** | **Discussion** | **Deadline** |

|  |  |
| --- | --- |
| Admin | * What we did since last meeting: * - setup podman vm with * - kubectl * - as nfs storage * - focus on diskrw * - as it is only one which really has a state * - have shell/python scripts * - are working on the stateful migration * - cpu container doesnt just calculate primes now * - it writes them into a file with timestamps * - started thinking about the reports structure * - table of content * --- * Questions: * - (PA)/BA deadline? * - as long as provide some results and a nice paper * - we pass! * - if we dont pass * - are we to repeat the same PA? * - * Notes: * - change hostname file? * - update hostnames based on ip addresses of new instance * - when doing migration * - /etc/hosts/ * - put ip addr of mongodb service * - and name wed like to use * - for migration * - main focus on * - is state of service same * - during migration? * - migration controller? * - options we chose we should mention and discuss in report * - use rsync * - for moving pv around * --- * Next steps: * - fix mongodb setup * - ip addr thingy * - script reads? * - Wissem * - if redirex takes too much time * - check if new instance has state of old one * - check measuring latency * - and downtime of service unavailable * - perform sts migration on mongodb * - we are not moving the data * - -> check if script starts from where it left off * - Wissem idea * - simulate distance new location * - > latency * - > scenario 1 * - db doesnt move * - > scenario 2 * - db also moves * - > compare both scenarios? * - >> interesting result for report! * - refine redis * - check if ram state is same * - criu migration * - do both migrations on cpu/redis * - start report * - structure * - references * - figures * - tables * - mail until 6th |

|  |  |  |
| --- | --- | --- |
| **Weekly meeting 22.11.2023 12:00 – 13:00**  Attendance: Wissem Rinchen, Tony | | |
| **Topic** | **Discussion** | **Deadline** |
| Admin | What we did since last meeting:    * - deployed redis and mongodb as statefulset * - setup cluster 2 * - almost had a heartattack * - wrote the mem script to write thousands key value pairs and keep changing them, but seems to not be so intensive? * - doesnt need to stress out the ram * - check rather the data in redis after migration * - if its still there, the state * - check, script shouldnt be in the service * - we are talking about the monitoring "script" here * - checks ram during migration * - should be on podman! * - all the monitoring stuff should be here * - wrote some scripts to automate the the setups and teardowns of the containers * - cannot get the mem script to reach critical levels? * - since a week been trying to figure out migrations, stuck * - deployed a first prototype of ms app: the working containers as a single pod instead of several pods * --- * Questions: * - ms app * - single pod? * - wait for now * - first have migrations running * - will probably not be able to reach all the goals set on complesis * - because we should start writing in 2 weeks * - and we still are stuck in the migrations * - tool velero? * - <https://velero.io> * - but maybe not * - storage node? * - Rini idea * - at least for diskrw container * Notes: * - daemonsets? * - maybe not, because time * - put in report * - work will be very important :) * --- * Next steps: * - podman vm * - kubectl * - access both clusters from here * - monitoring tools * - depends on service * - locust * - script data redis if changed or not * - cpu, what is last prime * - needs to continue from last prime * - queries the last * - storageclass, pv, pvc * - then we can figure out the diskrw container/migration! * - we need to figure out the migrations * - otherwise no data to compare/analyse * - start the report asap! * - at least layout * - ULTIMATE GOAL IS TO HAVE AT LEAST MIGRATION! |  |

|  |  |  |
| --- | --- | --- |
| **Weekly meeting 8.11.2023 12:00 – 13:00**  Attendance: Guerkan, Rinchen, Tony | | |
| **Topic** | **Discussion** | **Deadline** |
| Admin | * \*\*Meeting Notes\*\* * - we did * - managed to deploy to cluster * - flask * - still problems though to connect, need service? * - new approach? workaround? * - mongodb * - redis * - mongo and redis are still * - missing scripts to put pressure on system * - added timestamps to cpu intensive * - reach out to Wissem! * - wait for feedback * - more resources for second cluster? * - until second cluster * - we keep looking into node to node migrations * - but we need to setup ourselves * - prepare questions! * - Github? * - actual migrations? * - Questions * - how do we go about both migrations node to node? * - because everything we find is cluster to cluster * - need a second cluster? * - istio? * - next steps * - statefulset migration * - services flask app * - save timestamps * - ultimate goal is a migration |  |

|  |  |  |
| --- | --- | --- |
| **Weekly meeting 1.11.2023 12:00 – 13:00**  Attendance: Rinchen, Tony, Wissem | | |
| **Topic** | **Discussion** | **Deadline** |
| Admin | * \*\*Meeting Notes\*\* * - \*\*migrations\*\* * - start * - finish FIRST FLASK and do the migration on this one! * - or on CPU Intensive! * - highest priority! * - were not able to continue \*\*flask\*\* * - external service component? * - we need to properly setup the service component * - as loadbalancer type * - internal * - vs external * - external service is easier * - we need to specify clearly * - we need THIS one * - check the long video!!! * - for flask * - also for mongodb * - specify port for external service * - 30xxx * - video! * - \*\*disk rw container\*\* * - (kinda working) * - (will continue tweaking) * - check long video for mongodb * - <https://www.youtube.com/watch?v=X48VuDVv0do&t=4576s> * - first have mongodb container running * - and in this have a/our script running!!! * - check if size limit can be increased in k8s * - \*\*redis/memory intensive container\*\* * - similar to mongodb * - have alot IO ops * - but in memory * - maybe use same script as mongodb |  |

|  |  |  |
| --- | --- | --- |
| **Weekly meeting 25.10.2023 12:00 – 13:00**  Attendance: Gürkan, Rinchen, Tony, Wissem | | |
| **Topic** | **Discussion** | **Deadline** |
| Admin | * - while being stuck * - > another approach via docker * - > that works * - > is that ok? * - >> yes is fine * - cpu intensive * - > prime number calculator * - > ok? * - >> still discussable, but good start :) * - physical meetings not must * - not two clusters for migration * - > for moment just one |  |
| Project scope | Goals for next meeting:  - 1. locust?  - > setup service component in yaml file, port 30903, ip for k8s service    - 2. MAIN FOCUS: migration?  - > statefulset first  - > then crio    - 3. document/record parameters and resulting data  - > measurement framework  - > prometheus? servicemesh? istio?  - > istio more comfortable!!!  - install into k8s environment  - > dont reinvent wheel  - > use data to show graph etc  - > timestamps  - > which aspects should we consider  - > latency or response?    - Notes  - > networking intensive  - 0.0.0.0 can be any ip addr?  - > traffic, service in k8s?  - > CLEARLY MENTION DOCKERHUBAPPROACH THIS ON DOCU  - > easier for public  - > cpu intensive  - > maybe change to configure cpu load  - > try different levels  - > locust to generate cpu load?  - > stress tests? |  |

|  |  |  |
| --- | --- | --- |
| **Weekly meeting 18.10.2023 12:00 – 13:00**  Attendance: Gürkan, Rinchen, Tony, Wissem | | |
| **Topic** | **Discussion** | **Deadline** |
| Admin | * Flask/Locust done (on podman vm) * Are able now to send files via scp from podman vm to control plane * Finished 4h video K8s introduction |  |
| Project scope | * Next steps: * Figure out how to connect * Create new modified container images * Find k8s location where images usually are * bring images/container to cluster in order to start from there * Get container running * Test traffic locust * Try with different params * Record/print/timestamp results, so we can go back and compare * Start CRIO migration * Then StatefulSet * Maybe copy first yaml and then modify accordingly * Notes: * Maybe change yaml file (with rest api etc) so that image not from dockerhub or so but from OUR image * Service in yaml file? * Try with current yaml * As next step, change to statefulset, then run it as such * Service gonna be another container * Goal is to map port of container/service? * 30903, not port 80 for locust test * Delete dead images * If and only if service is enough, no ingress controller needed * IMPORTANT * We should note stuff * Remember we put this stuff public * Versions, config params |  |

|  |  |  |
| --- | --- | --- |
| **Weekly meeting 11.10.2023 12:00 – 13:00**  Attendance: Gürkan, Rinchen, Tony, Wissem | | |
| **Topic** | **Discussion** | **Deadline** |
| Admin | * Wissem infrastructure meetings happened * redis didnt happen * most of the stuff discussed done |  |
| Project scope | * flask app * locust app which sends http requests/traffic * 200 response to client should be there * migration (criu/stateful -> both stateful) * first go with stateful migration! * and then criu * no regular Wissem techmeeting * more ad hoc * http traffic for the app * map port of container to podman host port * send frames to ingress for flask app * still need to connect master with podman * convert docker to yaml file for k8s * directly put flask on k8s and test locust there |  |

|  |  |  |
| --- | --- | --- |
| **Weekly meeting 04.10.2023 12:00 – 13:00**  Attendance: Rinchen, Tony, Wissem | | |
| **Topic** | **Discussion** | **Deadline** |
| Setup | 1) Setup infrastructure   1. xSetup 1 K8s cluster with kubeadm (test with redis -> memory intensive)-> redis we leave for the moment being 2. xSetup Podman VM -> should be done 3. xStart with making the network-intensive container (simple Flask API) 4. xSetup the Locust in the Podman VM (maybe later on local laptop) -> locust is installed   2) Cycle to follow for each container (net-intensive, mem-intensive, disk-intensive)   1. xTest network-intensive in Podman with Locust (just check that the API runs) 2. Generate yaml file for the net-intensive container and run it on K8s 3. Test network-intensive in K8s with Locust 4. Start with one of the migration methods (CRIU or StatefulSet) 5. Test the migration method 6. Start the other one 7. Test again   3) Next phase is to put the containers together in one pod to migrate |  |

|  |  |  |
| --- | --- | --- |
| **Weekly meeting 27.09.2023 12:00 – 13:00**  Attendance: Gürkan, Rinchen, Tony, Wissem | | |
| **Topic** | **Discussion** | **Deadline** |
| Admin | * Rinchen and Tony continued research Papers * Trying to setup K8 |  |
| Setup | **Openstack, K8**   * Q: fixed requirement? Standard ubuntu image, may need some config -> check with Wissem * For mem container: instance with high memory! * -> but high memory should be relative to available memory! |  |
| **Overleaf, paper**   * Created Overleaf acc and have access, get used to Latex * Check PA paper structure * German AND English abstract * Check paper requirements * Appendix |  |
| GitHub | * Markdown for GitHub project * GitHub instead of GitLab * Should be public repo at the end and NOT ZHAW GitHub * Automation? Auxiliary files for script based experiments? * Use Private accounts, but insert “was implemented as part of ZHAW PA” * Existing containers -> to be extended? What should we look for? * -> Follow literature for this!, install other people’s GitHub projects/databases * Write programs/dummy apps for experiments to make stress/performance tests in the containers * End product should be like a program/application * -> runs some scripts * -> one big executable in python (?), python script * -> config file from command line * -> .csv files to store results, for plots, analysis, collection of data to persist |  |
| Etc. | * Jira should be used more for ourselves (in background) |  |
| Goals until next meeting | * Continue with paper research (until Sa, continuously) * Playing around with K8 tutorials to get to know better (Fr) * Finish setting up test/working environment on Openstack (Fr, Sa) * Setup Mendeley repo (Sa) * Setup GitHub repo (Sa) * once done -> start with first container until next meeting (Sa, So, Mo) |  |

|  |  |  |
| --- | --- | --- |
| **Weekly meeting 20.09.2023 12:00 – 13:00**  Attendance: Gürkan, Rinchen, Tony, Wissem | | |
| **Topic** | **Discussion** | **Deadline** |
| Admin | Project plan is ok, some minor changes:   * Milestone meetings around once per month: * M1: After first 2 MS apps (disk & memory intensive) * M2: After all MS apps * M3: After migration * M4: After submission   Start container migration after M1 |  |
| Setup | Wissem creates an OpenStack project for Tony & Rini to work on | 22.09.23 |
| Gürkan creates an Overleaf project for Tony & Rini to write the report on |  |
| Tony & Rini do research, read papers and try to set up a working environment |  |

|  |  |  |
| --- | --- | --- |
| **Kick-Off meeting 15.09.2023 11:00 – 12:00**  Attendance: Gürkan, Rinchen, Tony, Wissem | | |
| **Topic** | **Discussion** | **Deadline** |
| Admin | Weekly meetings take place online via Teams or at the office in Winterthur. Milestone meetings take place in person in Winterthur |  |
| A "Vereinbarung" needs to be signed that the PA is being done in English |  |
| At the end of the PA Tony + Rini need to do a presentation |  |
| Paper is to be written with Overleaf (Latex), Gürkan provides his account for access |  |
| Use [Mendeley](https://www.mendeley.com/search/) for "paper tracking" |  |
| One of the students has to take notes and write a short protocol after every meeting |  |
| Tony and Rini provide a rough plan as to how this project will be done including milestones | 20.09.23 |
| Project scope | * Build MS apps from preexisting containers * Perform migration on those apps in different settings   + i.e. Network heavy, memory heavy, cpu heavy * Analyze performance (based on best practice on how to evaluate performance) * (possible) Migration of VMs |  |
| Project has to be reproducable   * One must be able to simply download, configure and run * Everything has to be documented |