Norwegian University of Science and Technology
Department of Telematics

Contact during exam [Faglig kontakt under eksamen]:
Bjarne E. Helvik (92667)

EXAM IN COURSE [EKSAMEN I EMNE]
TTM4120 Dependable Systems [Pålitelige systemer]

Wednesday [Onsdag] 2012-05-30
09:00 – 13:00

The English version starts on page 2.

Den norske bokmålsutgaven starter på side 11.

Hjelpemidler:
D - No printed or handwritten material is allowed. Predefined simple calculator [Ingen trykte eller håndskrevne hjelpemidler tillatt. Forhåndsbestemt enkel kalkulator]

Sensur 2012-06-20

English version[1]

A computing centre providing cloud services is organised as illustrated in Figure 1. It is constituted by N identical racks, each containing n identical processors. Each of the racks are dually homed to an internal network. Communication within the rack may be regarded as fault free. M gateways interface the computer centre to the Internet. The gateways are dually homed to both the internal network and the Internet.

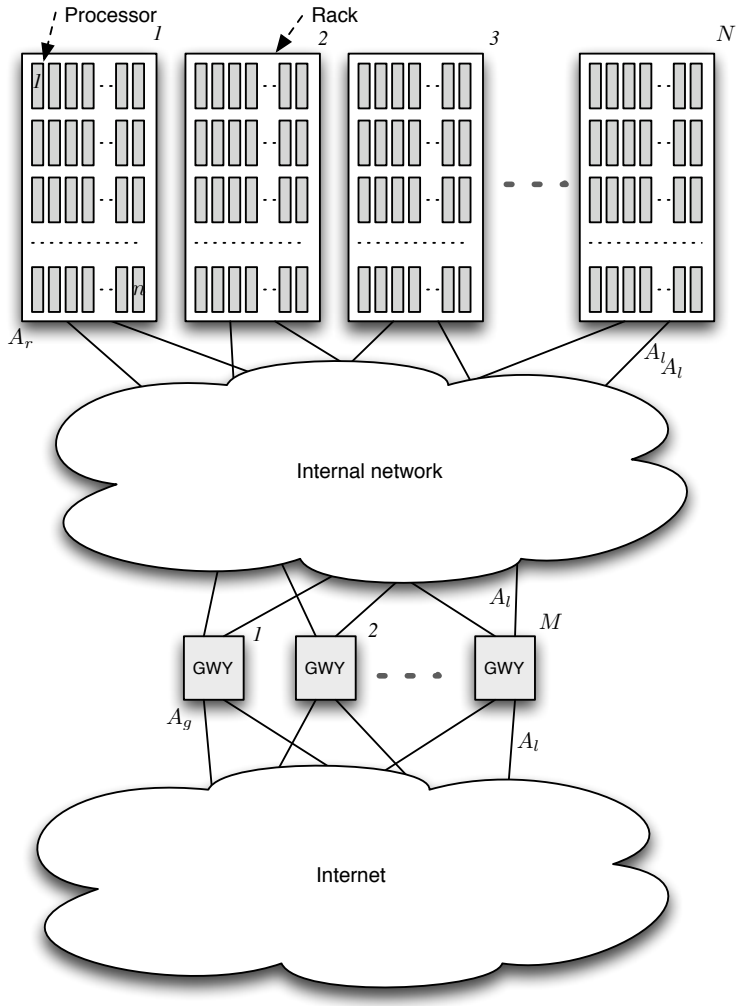


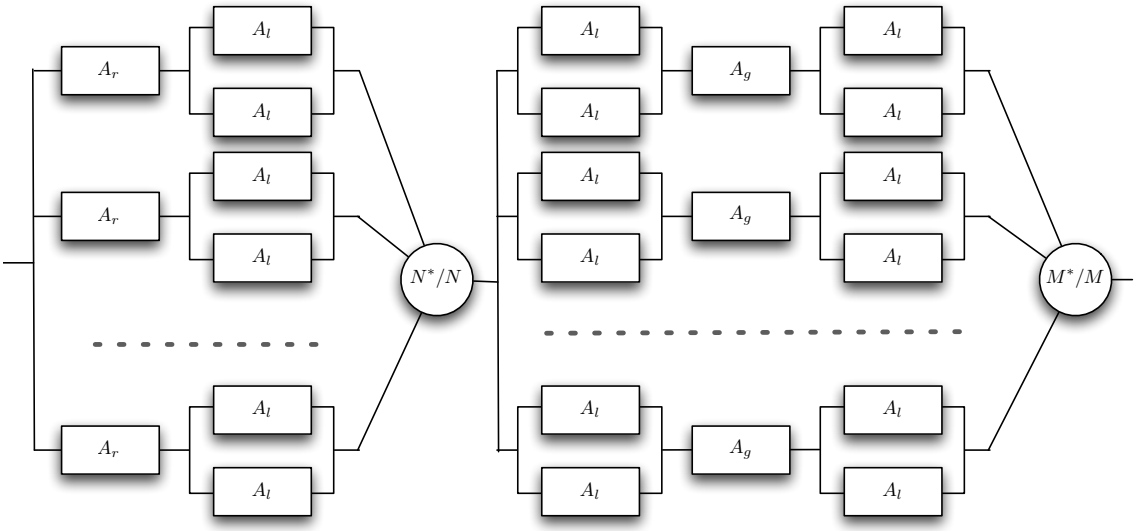
Figure 1: Sketch of computer centre.[Skisse av databehandlingssenter]

A rough prediction of the availability of this design should be made. To make this prediction it is assumed that entire racks, gateways, and interconnections (lines in the figure) between these and the

[1]In case of divergence between the English and the Norwegian version, the English version prevails.

networks, may fail. The asymptotic availability of these components are A_r, A_g and A_l respectively. All the failure processes are Poisson with intensities $\lambda_{r,}\lambda_g$ and λ_l for these components and λ_p for the processors. Each of the interconnections has sufficient capacity to carry the entire load to and from the connected devices. For the computer centre to deliver service with the required capacity, at least N^* of the racks and M^* of the gateways must work. Processors, and unspecified parts of the network are not considered in item a).

a) Based on the assumptions and parameters above, make a model of the computing centre in order to predict its availability, and establish an expression that determines this availability. If you make additional assumptions, state these.



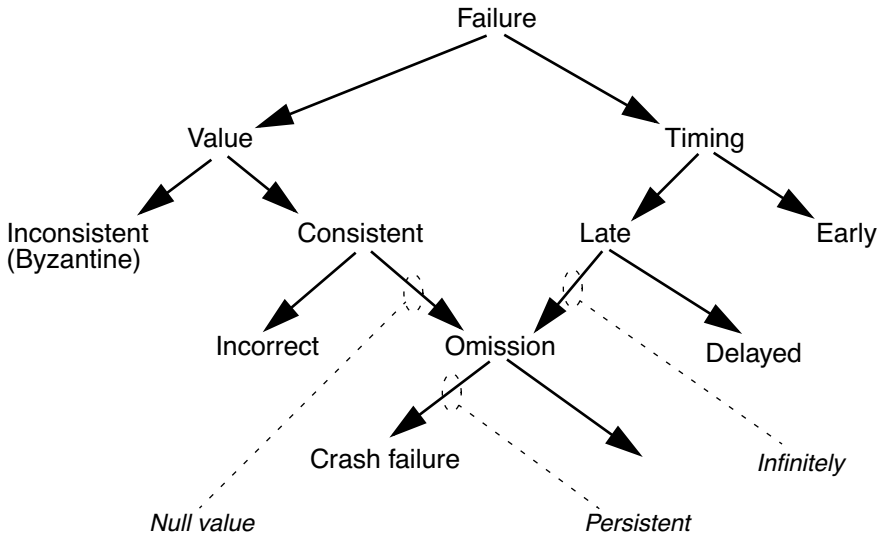
$$
\begin{aligned}
A_R &= A_r(1-(1-A_l)^2) \\
A_G &= A_g(1-(1-A_l)^2)^2 \\
A_S &= \left(\sum_{k=N^*}^{N}\binom{N}{k}A_R^k(1-A_R)^{N-k}\right)\left(\sum_{k=M^*}^{M}\binom{M}{k}A_G^k(1-A_G)^{M-k}\right)
\end{aligned}
$$

Failure and repairs of all network components included in the model are independent.

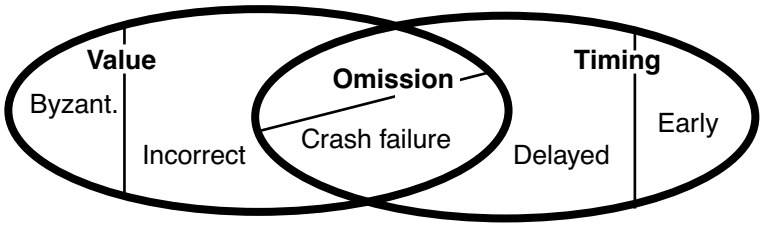
The computing centre hosts virtual machines (VM) for the cloud customers. For the sake of simplicity, we assume that one processor supports one and only one VM. A VM may be placed, i.e., executed, on any working processor in the system. When the processor or the rack containing the processor fails, the VM, i.e., the service executing on this processor, will fail. The VM fails according to a specific failure semantics.

b) What is meant by failure semantics? Briefly describe the different types of failure semantics. Assume that the VM's failure semantics is in the value domain. A (virtual) fault tolerant system providing an uninterrupted service shall be built from such VMs. What is the number of replicas of the VM that is needed, dependent on each of the the VM's value domain failure semantics?

The failure semantics of a system is its likely failure behaviour. For a more extensive description, see the lecture notes section 1.2.3. The types of failure semantics is summarized in the figure below, fetched from this section.



a) Graph depicting subdivision of failure modes



b) Venn diagram

Figure 1.12 Classification of failures

- Crash failure -> 2 replica
- (Omission failure in general -> 2 replica, if sporadic null value replica are recognized.)

- Incorrect result, consistent behaviour -> 3 replicas
- Byzantine (arbitrary) failures -> 4 replicas (for single fault tolerance)

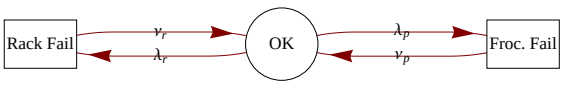
To improve the availability of the VM service, each VM is made fault tolerant by passive replication. (Passive replication is also denoted stand-by replication). In the computing centre, each active VM has a passive replica residing on a processor in another rack. When a processor fails, the standby continues the execution after a negative exponentially distributed time with expectation ν_p^{-1}. When a rack fails, all processors in the rack, and the VM executing on these, fail. The standbys of the VMs continue execution after a negative exponentially distributed time with expectation ν_r^{-1}. These recovery times may be assumed to be independent. At the same time the passive replica becomes active, a new passive replica is created.

c) Assume that one fault handling takes place at a time. What are the main factors contributing to the VM down time after a failure? Under what condition will the VMs have their maximum availability? Establish an expression for the maximum asymptotic availability a VM may have, $A_{VM,\max}$. What is the pdf (probability density function) of the VM-service down time in this case?

The main factors:

1. Time needed to restore the processing on a spare,
2. the time until a spare is provided (zero if spares are available, otherwise dependent on repair times and other VMs waiting for a spare.)

The main argument is that the maximum availability is achieved when the centre does not run out of spare processors, i.e. only the restoration time contributes to the down time. This yields the model below



which solved (simple balance equations) gives the maximum asymptotic availability

$\frac{\nu_p\nu_r}{\nu_p\lambda_r+\nu_r(\lambda_p+\nu_p)}$.

The ratio between down times and their respective yields the down time distribution. (See lecture notes)

$\frac{\lambda_p e^{-t\nu_p}}{\lambda_p+\lambda_r} + \frac{\lambda_r e^{-t\nu_r}}{\lambda_p+\lambda_r}$

At the computing centre, there are *one* repairman repairing rack failures, and a large number of repairmen dealing with processor failures. As an approximation, we may assume that there are one

repairman per processor. All repairmen are working independently of each other. The repair times are independently negative exponentially distributed with expectations μ_r^{-1} and μ_p^{-1} respectively. To simplify the analysis, assume that processors in failed racks may fail and be repaired independently of the state of the rack. (But processors in failed racks can not provide any service.)

The probability of having i failed units in a system with m units, each having a Poisson failure process with intensity λ is given for the following two cases:

1. One repairman with negative exponentially i.i.d. repair times with rate μ: $\tilde{p}_i(\lambda, \mu, m) = a_i / (\sum_{j=0}^{m} a_j)$ where $a_i = (\lambda/\mu)^i \, m!/(m-i)!$.
2. m independent repairmen, each with negative exponentially i.i.d. repair times with rate μ: $\hat{p}_i(\lambda, \mu, m) = \binom{m}{i} a^i (1-a)^{(m-i)}$ where $a = \lambda/(\lambda+\mu)^i$.

d) Expressed with $\tilde{p}_i(\ldots)$ and $\hat{p}_i(\ldots)$ given above, what is the probability of having i_r failed racks and i_p failed processors in the working racks? Motivate the answer. Let the centre host k VMs. For what range of i_r, the number of failed racks, can never all VMs be served? When i_r racks have failed, for what range of i_p, the number of failed processors, will not all VMs be served? Under the above assumptions, establish an expression for the probability U_S that the system runs out of spare processors, i.e. one or more VMs becomes unavailable due to exhaustion of spare processors.

With the assumptions above, the failure and repairs of processors and racks may be regarded as independent. Hence, the probability is $\tilde{p}_{i_r}(\lambda_r, \mu_r, N) \cdot \hat{p}_{i_p}(\lambda_p, \mu_p, n(N-i_r))$.

All VMs can never be served if: $(N-i_r)n < k$ and hence $i_r > N - k/n$ bounded by the number of racks in the system $i_r \leq N$.

If $(N-i_r)n \geq k$, all VMs cannot be served if $(N-i_r)n - i_p < k$ and hence $i_p > n(N-i_r) - k$ bounded by the number of processor in working racks $i_p \leq n(N-i_r)$.

The system fails when there is less "usable" processors in the system than k. hence, for i_r failed racks, there are less than k working corresponding to more than $n(N-i_r) - k$ failed.

$$U_S = \sum_{i_r=0}^{N} \tilde{p}_{i_r}(\lambda_r, \mu_r, N) \sum_{i_p=\max(0, n(N-i_r)-k+1)}^{n(N-i_r)} \hat{p}_{i_p}(\lambda_p, \mu_p, n(N-i_r))$$

Alternately the sum over i_r may be split into the two ranges found above.

e) The centre host k VMs. When i_r and i_p are such that the system run out of spares, what is the probability (expressed by i_r , i_p, n, N and k) that a random VM is affected by the lack of spares? What is the unavailability of a VM due to lack of spare processors, $U_{VM,s}$? (Hint: Use the same line of reasoning as in item d).) Since simplifying assumptions are made, this value will be an approximation. Is the obtained approximation for $U_{VM,s}$ conservative (not smaller than the true value) or not, compared to the more realistic case where we have limited number of repairmen for processor repair? Motivate the answer; a formal proof is not required. Use $U_{VM,s}$ and the result in item c) to find a simple and rough approximation to the availability of a VM.

The probability of a VM being affected, is proportional to the number of processors short to meet the requirement. Denote this number ℓ. Regard the case with i_r failed racks. When $i_p = n(N-i_r)-k+\ell$ processors has failed, there are $\ell = i_p + k - n(N - i_r)$ out of k VMs out of service.

Alternative line of reasoning: To deal with all the VMs, there should have been i_p+k+ni_r processors. Hence, we have $\ell = i_p + k + ni_r - nN$ too few. the probability is ℓ/k.

Using the same sum over the failure space as above

$$U_{VM,s} = \sum_{i_r=0}^{N} \tilde{p}_{i_r}(\lambda_r, \mu_r, N) \sum_{i_p=\max(0,n(N-i_r)-k+1)}^{n(N-i_r)} \hat{p}_{i_p}(\lambda_p, \mu_p, n(N-i_r)) \frac{i_p + k - n(N-i_r)}{k}$$

With a limited number of repairmen, the expected downtime after a failure will increase due to waiting for a repairman. Hence, down-times will be longer and the unavailability larger. The approximation is not conservative.

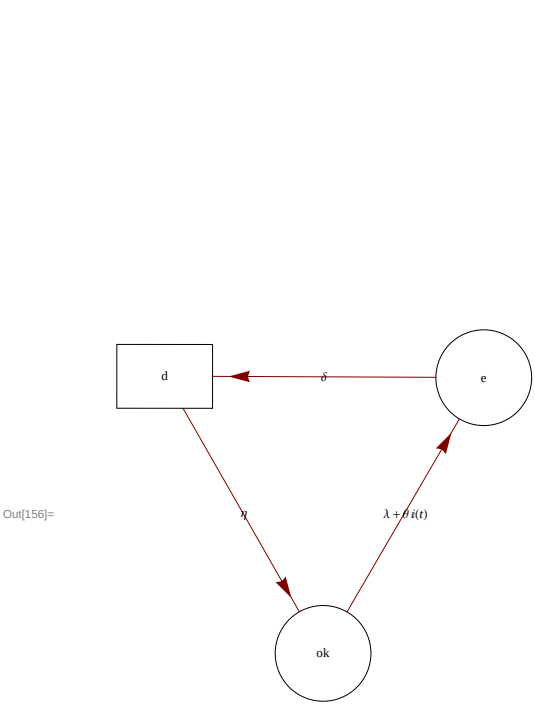
One of the effects, will dominate. $A_{VM} = \min(A_{VM,\max}, 1 - U_{VM,s})$ An alternative more conservative answer is $A_{VM} = A_{VM,\max} - U_{VM,s}$ or $A_{VM} = \max(A_{VM,\max} - U_{VM,s}, 0)$

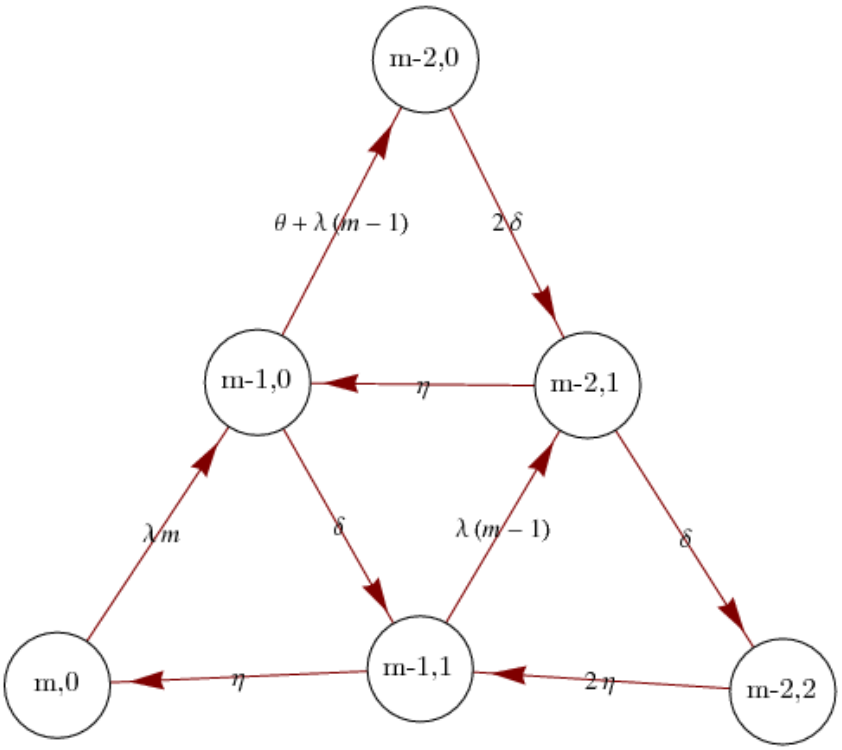
The VMs are managed by a distributed management system, which runs one identical management process on each processor. Each of these processes has a software fault activation rate λ_{SW}. When a fault is activated, it generates an error, which stays latent in the process until it causes a failure of the process. This latency time is negative exponentially distributed with expectation δ^{-1}. When it fails, a process is restarted in a negative exponentially distributed time with expectation η^{-1}. The process is error free after the restart. The processes co-operate tightly, and the erroneous state in a process will introduce an error in the state in any of the other working and error free processes in the system with an intensity θ.

f) Establish a state model illustrating the behaviour of one such process. If necessary introduce additional variable(s) to indicate the state of the rest of the software system. Make a partial diagram of the complete software system, which includes all states where two or less processes are affected by errors or failures. How many states would there be in the complete model?

If we denote the total number of working processors that are in the error state at time t by $i_e(t)$ the behaviour of a process is modeled to the right.

For the partial state model of the complete system, let x, y denote the state with x working and non erred processes, and y that has failed.





The total number of states becomes $((n \cdot N + 1)^2 + n \cdot N + 1)/2$

A simulation study of the behaviour of the distributed management system is performed. A sample result is shown in Figure 2, where $i_e(t)$ denotes the number of processors that is running with an erroneous internal state at time t and $i_f(t)$ denotes the number of processes that is not working at time t and due to software failure.

g) What is (approximately) the intensity that processes fails with due to software, when the system reaches steady state, $z_{\text{sw}}(\infty)$ (let say the system is in steady state at time $t = 2.5$ in Figure 2)? What is the initial intensity of processes failing, i.e., $z_{\text{sw}}(0)$? Are there "physical" arguments for this value? Do you know a technique that may be used to rectify the problem errors accumulating in a software system? Give a short (2-3 lines) description.

The intensity of processors failing is equal to the intensity of processes restarting, hence $z(\infty) = i_f(\infty)\eta \approx 7000$. It may also be argued as the intensity of processes where errors leads to failure, i.e. $z(\infty) = i_e(\infty)\delta \approx 7000$.

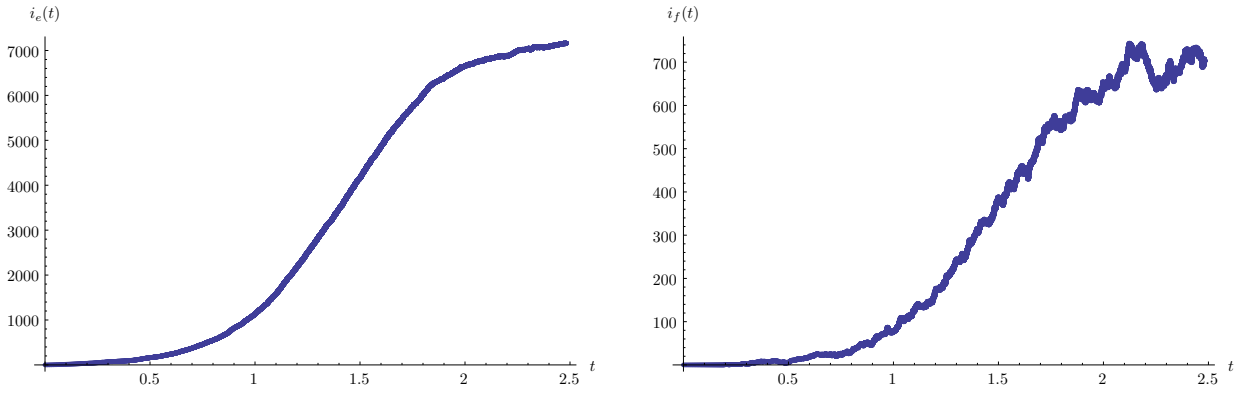


Figure 2: Simulation results from the distributed management system described above with parameter values $n \cdot N = 10000,\ \lambda = 0.01,\ \theta = 0.0005,\ \eta = 10,\ \delta = 1$.

From the figures, we have that $z(0) = 0$. Physically, this is due to the fault activation -> error -> failure process, where no fault activation leads to an immediate failure since two events are needed.

Comment; Analytical solution

Note that a numerical solution may be obtained by establishing mean value balance equations for the entire system

eqs $= \{\text{iok}(\lambda + \theta\text{ie}) == \text{id}\eta, \delta\text{ie}==\text{id}\eta, \text{ie} + \text{iok} + \text{id} == m\}$

$\{(0.01\ + 0.0005\text{ie})\text{iok} == 10\text{id}, \text{ie} == 10\text{id}, \text{id} + \text{ie} + \text{iok} == 10000\}$

and solve these

Solve[eqs, {iok, ie, id}]

A closed form does not seem feasible. Solve was unable to solve the system with inexact coefficients. The answer was obtained by solving a corresponding exact system and numericizing the result.

$\{\{\text{iok} \to 1994.52, \text{ie} \to 7277.71, \text{id} \to 727.771\}, \{\text{iok} \to 10027.5, \text{ie} \to -24.9829, \text{id} \to -2.49829\}\}$

The technique is call software rejuvenation. The system is set to a default initial state (i.e., restarted) at regular intervals or when some contamination indicators" reaches a threshold.

In Figure 1 an internal network is indicated, handling the traffic in-between the racks and between the racks and the gateways. These are dual homed to the network, but it is not decided how the dual homing is utilized. This network must have a high availability. It shall be designed for uninterrupted operations, i.e., when a network element fails, the ongoing communication between the connected units shall not be interrupted or additionally delayed. It may be assumed that when a network element fails, the resulting packet transfer failures have omission semantics. Other issues that should be considered in the design of the network are; scalability, capacity, traffic interests, limitation of the effect of not tolerated faults.

h) Propose a network structure and mode of operation that meets the availability and continuity requirements above. Motivate the choices you make. Discuss/point out any strengths and weaknesses, that you are aware of, in your proposal, both with respect to the primary criteria, and the "other issues" listed above.

The answer to this question should not exceed two pages. NOTE: There is no single standard solution to this question. It is not required that you should come up with the perfect network in all respects. Make a reasonable proposal and discuss it; the quality of the discussion is important.

There is no standard solution. One feasible solution is:

- The network has two independent layers. Independence should be implemented wrt. as many factors that is techno-economic feasible, e.g. power supply, location (adjacent buildings). The extreme would be to have networks of different internal structure and make.
- To have uninterrupted service, active redundancy may be used. Since network elements has omission semantics, duplication is sufficient. (See also question b, which serve as a hint – and a reverse hint.) Every packet is duplicated and send across the two network layers. This may be handled on ISO layer 4, e.g. by using SCTP, or by special designed mechanisms on layer 3 or 2 if the connection overhead should be avoided. (which layer depends on technology chosen).
- The topology in each layer should be chosen so that the traffic flow (routing / switching) in the network may altered after a failure (e.g. rerouting or switch reconfiguration).
- The topology of the network layers should be chosen so that failure effect of each switch is limited. The topology may be designed to handle asymmetric traffic, where most go between racks and gateways. It should be able to be continuously extended in terms of N and M, and capacity. This indicates a medium dense mesh like structure. (A ring network would not do).

Norsk bokmål utgave[2]

Et datamaskinsenter som leverer cloud-tjenester er organisert som illustrert i figur 1. Det består av N identiske stativer, som hver inneholder n identiske prosessorer. Hvert av stativene er dobbelttilknyttet et internt nett. Kommunikasjon innenfor et stativ kan betraktes som feilfri. M "gatewayer" utgjør grensesnittet mellom datasenteret og Internett. "Gatewayene" er dobbelttilknyttet til både det interne nettet og Internett.

En grov prediksjon av tilgjengeligheten av denne systemløsningen skal gjennomføres. For å gjøre dette, antas det at hele stativ, "gatewayer", og sammenkoblingene (linjer i figuren) mellom disse og nettverkene, er de elementene som kan feile. Den asymptotiske tilgjengeligheten av disse komponentene er henholdsvis A_r, A_g og A_l. Alle feilprosessene er Poisson prosesser med intensitet λ_r, λ_g og λ_l for disse komponentene og λ_p for prosessorene. Hver av tilknytningene har tilstrekkelig kapasitet til å ta hele belastningen til og fra tilkoblede enheter. For at datamaskinsenteret skal kunne levere tjenester med den krevde kapasitet, må minst N^* av stativene og M^* av "gateway"ene være arbeidende (oppe). Prosessorer, og uspesifiserte deler av nettverket er ikke vurdert i punkt a).

a) Med utgangspunkt i forutsetningene og parametrene over, lag en modell av datamaskinsenteret for å prediktere dets tilgjengelighet. Basert på modellen, etabler et uttrykk som bestemmer denne tilgjengeligheten. Hvis du introduserer ytterligere antagelser, oppgi disse.

Datamaskinsenteret er vert for virtuelle maskiner (VM) for kunder av "cloud"-tjenester. For enkelhets skyld antar vi at en prosessor støtter én og bare én VM. En VM kan plasseres, dvs. bli eksekvert, på en vilkårlig arbeidende prosessor i systemet. Når prosessoren eller stativet som inneholder prosessoren feiler, feiler også den VMen (dvs. tjenesten som kjører på prosessoren) . VMen feiler i hht. en spesifikk feil-semantikk.

b) Hva menes med feil-semantikk? Beskriv kort forskjellige typer feil-semantikk. Anta at VMens feil-semantikk er i verdi-domenet. Et (virtuelt) feiltolerant system som skal gi en uavbrutt tjeneste bygges av slike VMer. Hva er antall replika som trengs, avhengig av hver av verdi-domene feil-semantikkene til VMene?

For å forbedre tilgjengeligheten av VM-tjenesten, er den gjort feiltolerant ved passiv replikering av VMene. (Passiv replikering er også kalt "stand-by" replikering.) I datamaskinsenteret har hver aktiv VM et passivt replika allokert til en prosessor i et annet stativ. Når en prosessor feiler, fortsetter det passive replikaet (reserven) utførelsen etter en negativ eksponensialfordelt tid med forventning ν_p^{-1}. Når et stativ feiler, vil alle prosessorene i stativet og VMene som eksekveres på disse feile. De passive replikaene av VMene fortsetter eksekveringen i løpet av en negativ eksponensialfordelt tid med forventning ν_r^{-1}. Disse gjennopprettelsestidene kan antas å være uavhengige. Samtidig som det passive replikaet blir aktivt, er et nytt passivt replika opprettet.

c) Anta at kun en feil må håndters ad gangen. Hva er hovedfaktorene som bidrar til en VMs nedetid etter en feil? Under hvilke betingelser vil VMene ha sin maksimale tilgjengelighet? Etabler

[2] I tilfelle uoverensstemmelse mellom den engelske og norske utgaven, er det den engelske som er gjeldende. Engelske betegnelser anvendes hvor ingen norsk oversettelse ble funnet.

et uttrykk for den maksimale asymptotiske tilgjengeligheten som en VM kan ha, $A_{VM,\max}$. Hva er pdf (sannsynlighetstetthetsfunksjonen) til nedetiden til VM-tjenesten i dette tilfellet?

I datamaskinsenteret er det én reparatør som reparerer stativfeil, og et stort antall reparatører som reparerer prosessorfeil. Som en tilnærming, kan vi anta at det finnes en reparatør for hver prosessor. Alle reparatørene arbeider uavhengig av hverandre. Reparasjonstidene er uavhengige og negative eksponentielt fordelt med forventning på henholdsvis μ_r^{-1} og μ_p^{-1}. For å forenkle analysen antas det at prosessorene i feilte stativ kan feile og bli reparert uavhengig av tilstanden til stativet (men de kan ikke levere noen tjeneste når stativet er feilet).

Sannsynligheten for å ha i feilte enheter i et system med m enheter, hvor hver enhet har en Poisson feilprosess med intensitet λ, er gitt for følgende to tilfeller:

1. Én reparatør med negativt eksponensialfordelt identiske og uavhengige reparasjonstider med rate μ: $\tilde{p}_i(\lambda, \mu, m) = a_i/(\sum_{j=0}^{m} a_j)$ hvor $a_i = (\lambda/\mu)^i m!/(m-i)!$.

2. m uavhengige reparatører, hver med negativt eksponensialfordelt identiske og uavhengige reparasjonstider med rate μ: $\hat{p}_i(\lambda, \mu, m) = \binom{m}{i} a^i (1-a)^{(m-i)}$ der $a = \lambda/(\lambda+\mu)^i$.

d) Uttrykt med $\tilde{p}_i(\ldots)$ and $\hat{p}_i(\ldots)$ gitt ovenfor, hva er sannsynligheten for å ha i_r feilte stativ og i_p feilte prosessorer i de arbeidene (ikke feilte) stativene? Begrunn svaret. La senteret være vert for k VMer. For hvilke verdier av i_r, antall feilte stativ, kan aldri alle VMene bli utført? Når i_r stativ har feilet, for hvilke verdier av i_p, antall feilte prosessorer, vil ikke alle VMene bli utført? Under ovennevnte forutsetninger, etabler et uttrykk for sannsynligheten U_S at systemet går tom for reserveprosessorer, dvs. én eller flere VM blir utilgjengelige på grunn av begrensinger i antall reserveprosessorer.

e) Senteret er vert for k VMer. Når i_r og i_p er slik at systemet går tomt for reserver, hva er sannsynligheten (uttrykt ved i_r , i_p, n, N og k) for at en tilfeldig VM blir berørt av mangelen på reserver? Hva er utilgjengeligheten for en VM på grunn av mangel på reserveprosessorer, $U_{VM,s}$? (Hint: Bruk samme grunnresonnement som i punkt d).) Siden det er gjort forenklede antakelser vil dette være en tilnærming. Er denne tilnærmingen til $U_{VM,s}$ konservativ (ikke mindre enn den sanne verdi) eller ikke, sammenlignet med det mer realistiske tilfellet der vi har et begrenset antall reparatører for prosessor reparasjon? Begrunn svaret; et formelt bevis er ikke nødvendig. Bruk $U_{VM,s}$ og resultatet i punkt c) for å finne en enkel og grov tilnærming av tilgjengeligheten for en VM.

VMene administreres av et distribuert system. Dette kjører en identisk prosess på hver prosessor. Hver av disse prosessene har en programvarefeilaktiveringsrate λ_{SW}. Når en feil blir aktivert, genererer det en feiltilstand, som forblir latent i prosessen før den fører til en feilhendelse av prosessen. Denne latenstiden er negativt eksponensialfordelt med forventning δ^{-1}. Når den feiler, vil en prosess bli restartet i løpet av en negativ eksponensialfordelt tid med forventning η^{-1}. Prosessen har ikke tilstandsfeil etter omstarten. Prosessene samarbeider tett, og en feiltilstand i en prosess vil introdusere en feiltilstand i enhver av de andre arbeidende og feilfrie prosessene i systemet med en intensitet θ.

f) Etabler en tilstandsmodell som illustrerer oppførselen til en slik prosess. Om nødvendig innføre ekstra variable for å indikere tilstanden til resten av programvaresystemet. Lag et partielt diagram av det komplette programvaresystemet, som inkluderer alle tilstander der to eller færre prosesser er berørt av feiltilstander eller feilytringer. Hvor mange tilstander ville det være i fullstendig modell?

En simuleringsstudie av oppførselen til det distribuerte systemet er gjennomført. Et eksempelresultat vises i figur 2, hvor $i_e(t)$ betegner antallet prosessorer som kjører med en feiltilstand ved tiden t og $i_f(t)$ betegner antallet prosesser som ikke er arbeidende ved tiden t på grunn av programvarefeil.

g) Hva er (tilnærmet) intensiteten som prosesser feiler med på grunn av programvaren, når systemet når stasjonærtilstand, $z_{\text{sw}}(\infty)$ (la si at systemet er stasjonært ved tiden $t = 2.5$ i figur 2)? Hva er den initielle intensiteten av prosesser som feiler, dvs. $z_{\text{sw}}(0)$? Er det "fysikalske" argumenter for denne verdien? Kjenner du en teknikk som kan brukes til å avhjelpe problemet med akkumulering av feiltilstander i et programvaresystem? Gi en kort (2-3 linjers) beskrivelse.

I figur 1 er det indikert et internt nett som håndterer trafikken i mellom stativene og mellom stativ og "gatewayer". Disse er dobbeltilkoblet til nettet, men det er ikke bestemt hvordan denne dobbeltilkoblingen skal benyttes. Dette nettet må ha en høy tilgjengelighet. Det skal være konstruert for uavbrutt drift, dvs. når et nettelement feiler, skal den pågående kommunikasjonen mellom de tilkoblede enhetene ikke bli avbrutt eller få tilleggsforsinkelser. Det kan antas at når et nettelement feiler, vil de pakkeoverføringsfeilene som intreffer pga. dette, ha unnlatelsessemantikk. Andre forhold som det skal tas hensyn til i konstruksjon av nettet er skalerbarhet, kapasitet, trafikkinteresser, begrensning av effekten av ikke-tolererte feil.

h) Foreslå en nettstruktur og -virkemåte som tilfredsstiller tilgjengelighets- og kontinuitetskravene ovenfor. Begrunn de valgene du gjør. Diskuter/påpek eventuelle styrker og svakheter du ser ved forslaget, både med hensyn til de primære kriteriene, og de andre forholdene som er nevnt ovenfor.

Svaret på dette spørsmålet bør ikke overstige to sider. MERK: Det er ingen enkelt standard løsning på dette spørsmålet. Det kreves ikke at du skal komme opp med det perfekte nett i alle henseender. Lag et rimelig forslag og diskutere det; kvaliteten på diskusjonen er viktig.