1 Algemeen

Algemeen gaan we ons systeem voorzien van een `Dynamische Cloud Booter` die ervoor gaat zorgen dat het systeem VM's gaat booten wanneer de eigen load te hoog zou worden. Hierdoor vereenvoudigen en versnellen we het proces om short jobs over te zetten naar de publieke cloud.

Verder gaan we ook nog rekening moeten houden met encryptie. Indien de gebruiker expliciet vraagt om encryptie moeten we zien dat het platform waarop we dit deployen ook encryptie kunnen voorzien.

2 Short Jobs

Als de geschatte responstijd van de berekening op minder dan 2 minuten zou bedragen, classificeren deze berekening onder de `Short Jobs`. Deze jobs zullen we meteen uitvoeren. Bij de short jobs zal ook geen rekening gehouden worden met de vereisten om het zo rap mogelijk, zo goedkoop mogelijk, .. te doen.

1. Als de gebruiker een berekening request met een uitvoeringstijd van minder dan 10 seconden dan gaan we deze in de eigen private cloud uitvoeren tenzij de load hoger is dan 90%.
   Indien de private load te hoog is:
   (a) Zal de job in de lokale wachtrij gezet worden.
   (b) Als het aantal jobs in de lokale wachtrij groter is dan vijf zal de berekening naar de publieke cloud overgezet worden. Indien er meer dan x hoeveelheid data mee overgezet moet worden, voegen we de berekening toch toe aan de private cloud.

3 Long Jobs

`Long jobs` zijn jobs waarvan de geschatte uitvoeringstijd meer dan 2 minuten vraagt.

3.1 Prioriteit : Zo snel mogelijk

De prioriteit van deze berekening is zo snel mogelijk. De gebruiker wil dat de berekening zo snel mogelijk gedaan is, ongeacht de kost. De focus zal dus voornamelijk liggen op het overzetten naar de publieke cloud.

3.1.1 Beperkte hoeveelheid benodigde data

De hoeveelheid benodigde data is kleiner dan 50MB.

- De private load is $< 20\%$
  Indien de private load $< 20\%$ kiezen we ervoor om de berekening lokaal uit te voeren. Door de hoge vereisten die we stellen kunnen we er vanuit gaan dat de berekening lokaal vrij rap zou gaan.

- De private load is $> 20\%$ en `Bootingstijd VM` $< 20\%$ van de geschatte uitvoeringstijd
  Indien aan deze vereisten voldaan is, kiezen we ervoor om de berekeningen naar de publieke cloud te verplaatsen. Hierdoor zal de vereiste rekentijd zeer beperkt blijven. We houden ook rekening met de geschatte `Bootingstijd` van de VM. Als de `bootingstijd` groter is dan een vooraf bepaald percentage is het vaak de moeite niet om de berekeningen naar de publieke cloud te forwarden.

- De private load is $> 20\% < 70\%$ en `Bootingstijd VM` $> 20\%$ van de geschatte uitvoeringstijd
  Nu kiezen we ervoor om de berekening lokaal uit te voeren.

- De private load is $> 70\%$ De berekening zal sowieso in de publieke cloud uitgevoerd worden.

3.1.2 Aanzienlijke hoeveelheid benodigde data

De hoeveelheid benodigde data is groter dan 50MB.

- De private load is $< 20\%$ en de hoeveelheid extra data $< 150MB$
  Indien de private load $< 20\%$ kiezen we ervoor om de berekening lokaal uit te voeren. Door de hoge vereisten die we stellen kunnen we er vanuit gaan dat de berekening lokaal vrij rap zou gaan.

- De private load is $< 20\%$ en de hoeveelheid extra data $> 150MB$
  Door de grote hoeveelheid data die we op onze server zouden moeten bijhouden, verkiezen we om deze berekening ook sowieso in de cloud te doen. We zetten dan ook de benodigde data over.

- De private load is $> 20\%$ De berekening zal sowieso in de publieke cloud uitgevoerd worden.

3.2 Prioriteit : Zo goedkoop mogelijk

De enigste vereiste die de gebruiker ons hier stelt is dat de berekeningen gedaan moeten worden tegen een zo goedkoop mogelijke prijs. We gaan dus (bijna) nooit gebruik maken van de publieke cloud.

3.2.1 Beperkte hoeveelheid benodigde data

- De private load is $< 30\%$
  Indien de private load $< 30\%$ kiezen we ervoor om de berekening lokaal uit te voeren. Als de huidige berekening de oorzaak is dat andere berekeningen met hogere prioriteit (Zie Prioriteit: Zo snel mogelijk) naar de cloud moeten, pauzeren we deze berekening.

- De private load is $> 30\%$ en er zijn andere processen bezig. Dan stoppen we dit proces totdat de private load terug kleiner is dan 30.

3.2.2 Aanzienlijke hoeveelheid benodigde data

??

3.3 Prioriteit : Balanceerde uitvoering

3.3.1 Beperkte hoeveelheid benodigde data

De hoeveelheid benodigde data is kleiner dan 50MB.

- De private load is $< 50\%$
  Indien de private load $< 50\%$ kiezen we ervoor om de berekening lokaal uit te voeren.
- De private load is $> 50\%$ en $< 70\%$ Nu kiezen we ervoor om de berekening lokaal uit te voeren. Indien we merken dat de lokale load niet meer daalt, kiezen we ervoor om dit process uiteindelijk naar de publieke cloud over te dragen. In tussentijd zal er hier een VM geboot worden zodat we tijdig alle resources kunnen overzetten. (Indien de het type berekening ons dit toelaat).
- De private load is $> 70\%$ De berekening zal sowieso in de publieke cloud plaatsvinden.

3.3.2 Aanzienlijke hoeveelheid benodigde data

Hierbij treffen we dezelfde regeling als bij `Prioriteit:zo snel mogelijk`

4 Verschillende aspecten

- Het type prioriteit
- encryptie
- benodigde data

5 Voorgestelde Policy Language

We gaan gebruik maken van de volgende configuratie:

$$< Priority > Fast < /Priority >$$
$$< Encryption > Required < /Encryption >$$
$$< Data > Small < /Data >$$

Mogelijke items voor prioriteit zouden de volgende zijn :

- Fast

- Cheap
- Balanced

Encryption :

- Required
- Prefered
- None

Data :

- Small (< 50) MB
- Medium (> 50) en (< 150) MB
- Large (> 150) MB

Intern definiëren we we de volgende eigenschappen voor de verschillende open source platformen: (alvast een voorbeeld).

$$
\begin{aligned}
&< Platform > Heroku < /Platform > \\
&< Encryption > Basic < /Encryption > \\
&< Log > Extended < /Log > \\
&< DataModel > Basic < /DataModel > \\
&< Processing > Basic < /Processing > \\
&< Deployment > Average < /Deployment > \\
&< Cost > Low < /Cost >
\end{aligned}
$$

De verschillende eigenschappen kunnen het volgende bevatten:

- Platform :
  Platform duidt de naam van het platform aan. (Verdere specificaties kunnen dan nog intern opgezocht worden)
- Encryption :
  Indien er de mogelijkheid is tot encryption duiden we dit ook aan.
- Log :
  Sommige cloudplatformen voorzien een zeer gedetailleerde log waarin alle kosten duidelijk te zien zijn. Voor sommige applicaties kan het handig zijn dat dit ter beschikking staat.
- DataModel :
  Dit aspect heeft oog naar hoe het cloudplatform omgaat met data en hoeveel data het kan verwerken.
- Processing :
  Sommige platformen hebben een grotere beschikbare (of schaalbare) rekenkracht. Naar mate de vereisten van de applicatie kunnen we dit dus optimaal kiezen.
- Deployment :
  Hoe snel kan de code gedeployed worden naar de specifieke cloud.
- Cost :
  Hoe veel kost het gebruik van de specifieke cloud.