PoC: Opzetten van Pydev in IBM Developer for z/OS om Python programma's uit te voeren op de mainframe.

Alif Monnoye.

Scriptie voorgedragen tot het bekomen van de graad van
Professionele bachelor in de toegepaste informatica

Promotor: Dhr. L. Blondeel
Co-promotor: Dhr. N. Sletten
Instelling: Den Norske Bank (DNB)
Academiejaar: 2023–2024
Eerste examenperiode

Departement IT en Digitale Innovatie .

HO
GENT

Woord vooraf

Dit is een proof of concept die ik wil uitwerken omdat dit hulp kan bieden in de modernisatie van de mainframe systemen van IBM. Op het moment van schrijven heb ik al stage gedaan bij 2 bedrijven die gebruik maken van een mainframe en deze gebruiken allebei IBM Developer for zOS om applicaties te schrijven in verschillende soorten programmeertalen zoals COBOL, PL/1, Python, Java, ... Python is één van de meest gebruikte programmeertalen op de dag van vandaag en hierin wordt dus veel ingezet door IBM om deze programmeertaal zo efficiënt mogelijk te laten werken op hun systemen. De skills in de 'oudere' programmeertalen zoals COBOL en PL/1 is sterk aan het afnemen dus is de inbreng van Python bijna een must om nog goede programmeurs te vinden. Een goede editor is ook nodig wat momenteel niet het geval is in IDz dus maak ik in dit onderzoek een stappenplan om dit zo goed mogelijk op te stellen.

Ik zou Njaal Sletten en Stian Botnevik van DNB, Noorwegen willen bedanken voor de hulp en inspiratie bij het schrijven en uitwerken van dit onderzoek. Ze hielpen allebei met het technische gedeelte en Stian kwam met de probleemstelling.

Samenvatting

IBM heeft voor zijn IBM Z Systems de deur opengezet naar vernieuwing door Java en Python toegankelijk te maken op de Mainframe. Hoewel deze talen ondersteund worden, is het niet altijd even makkelijk om programma's hierin te schrijven. IBM Developer for z/OS is een stap in de goede richting door zijn functionaliteit om in een bekend programma scripts te openen en aan te passen die opgeslagen zijn op mainframe. Aangezien dit Eclipse based is, is er geen Python interpreter waardoor Python scripts aangepast moeten worden in een text editor.

Deze paper zal een proof of concept zijn voor het opstellen van Pydev en een Python IDE in IBM Developer for z/OS. Hierdoor kunnen Python developers efficiënter code schrijven, aanpassen en submitten in de Unix System Services omgeving van de mainframe. Verder zal er ook nog onderzocht worden om de output van de Python programma's te zien in de console van IBM Developer for z/OS. Dit is belangrijk voor de programmeurs van DNB die werken in dit programma die nu hun Python code moeten testen in een ssh connectie.

In dit onderzoek zal er gewerkt worden met de Pydev plug-in voor Eclipse en zal er een stap voor stap instructie gegeven worden over hoe dit ingesteld moet worden. Ook zal er onderzocht worden of deze scripts uitgevoerd kunnen worden door IBM Developer for z/OS in de Mainframe omgeving.

Het resultaat zal een volledige Proof of concept zijn voor een Python interpreter op te zetten in dit programma. Dit is vooral relevant voor Python developers in bedrijven die IBM Developer for z/OS gebruiken in combinatie met een IBM Z Mainframe.

Inhoudsopgave

Lijst van figuren

1

Inleiding

1.1. Probleemstelling

IBM heeft voor zijn IBM Z Systems de deur opengezet naar vernieuwing door Java en Python toegankelijk te maken op de Mainframe. Hoewel deze talen ondersteund worden, is het niet altijd even makkelijk om programma's hierin te schrijven. IBM Developer for z/OS is een stap in de goede richting door zijn functionaliteit om in een bekend programma scripts te openen en aan te passen die opgeslagen zijn op een mainframe. Aangezien dit Eclipse based is, is er geen Python interpreter die gebruikt kan worden waardoor Python scripts enkel geopend en aangepast kunnen worden in een text editor. Dit komt vanzelfsprekend niet met een syntaxcheck, code completion, etc

Dit is een probleem voor developers zoals Stian Botnevik van DNB om Python code te schrijven in dit programma. Hoewel scripts vaak ook lokaal worden geschreven in een programma zoals Visual Studio Code en dan worden overgezet naar de mainframe, moeten er nogsteeds aanpassingen gebeuren door de verschillen in runtime environment. In een klein Python programma lukt dit wel nog in een text editor maar als ze wat groter zijn, wordt dit veel moeilijker.

1.2. Onderzoeksvraag

In dit onderzoek zal er gewerkt worden met de Pydev plug-in voor Eclipse en zullen er stap voor stap instructies gegeven worden over hoe dit opgezet wordt. Aangezien IBM Developer for z/OS toch nog verschillend is van Eclipse kunnen er verschillende problemen opkomen die niet gebeuren in het standaard Eclipse programma.

Eens dit is opgezet, zal er onderzocht worden of deze scripts uitgevoerd kunnen

worden in de Mainframe omgeving via IBM Developer for z/OS.

1.3. Onderzoeksdoelstelling

Het resultaat zal een volledige proof of concept zijn voor een Python interpreter op te zetten in IBM Developer for z/OS. Dit is vooral relevant voor Python developers in bedrijven die dit programma gebruiken in combinatie met een IBM Z Mainframe.

1.4. Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 2 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeksdomein, op basis van een literatuurstudie.

In Hoofdstuk 3 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen.

In Hoofdstuk 4 wordt de gebruikte terminologie in detail besproken samen met andere onderwerpen relevant voor het onderzoek.

In Hoofdstuk 5 wordt de proof of concept duidelijk weergegeven en besproken.

In Hoofdstuk 7, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvragen. Daarbij wordt ook een aanzet gegeven voor toekomstig onderzoek binnen dit domein.

2

Stand van zaken

2.1. Een andere manier van werken

Veel bedrijven die werken met een mainframe hebben het probleem dat ze onvoldoende mensen vinden met de nodige kennis over deze technologie. De introductie van Unix systemen op de mainframe in het jaar 2000 (Mertic, 2020) had dit probleem wat verminderd maar zeker niet weggewerkt. De oorzaak van deze situatie is door de oude technieken die het systeem gebruikt. COBOL en PL/1 zijn niet de meest aantrekkelijke programmeertalen om te leren en mensen kiezen liever Bash als scriptingtaal in plaats van REXX. Hoewel IBM veel inzet op documentatie en online leerplatformen zoals IBM Z Xplore, blijft het tekort van ervaren mensen nog steeds te laag.

De mainframe wereld zal zich dus moeten aanpassen aan de vaardigheden van de mensen door over te schakelen naar beter gekende manieren van werken. Python bijvoorbeeld is een programmeertaal die steeds meer populariteit krijgt sinds zijn ontstaan in de vroege jaren 90. Dit is niet enkel bij reeds ervaren programmeurs, maar mensen die net beginnen programmeren kiezen hier steeds vaker voor. Dit is vooral door zijn begin vriendelijkheid, verschillende doeleinden en een actieve community. (Johnson, 2023) IBM heeft dit opgemerkt en een Python- compiler en interpreter ontwikkeld voor z/OS genaamd IBM Open Enterprise SDK for Python. Hierdoor kun je met Python interactie hebben met z/OS om bijvoorbeeld applicaties te ontwikkelen of de resources van het systeem beheren. Door de Unix omgeving heeft de programmeur ook geen geavanceerde z/OS kennis nodig. (Klaey, 2023)

2.2. Bedrijven in het werkveld

Een bank is een goed voorbeeld van een bedrijf die gebruik maakt van IBM hun Z Systems. Dit valt te concluderen door het feit dat maar liefst 92 van de top 100 banken wereldwijd een mainframe gebruikt. Dit is ook logisch aangezien er gemiddeld 12.6 miljard financiële transacties per dag zijn dat door deze systemen worden uitgevoerd. (Wagle, 2017)
Door dit aantal is de nood voor een mainframe toch niet te onderdrukken maar het is niet enkel het aantal transacties dat deze machine kan uitvoeren, het is ook de snelheid, schaalbaarheid en beveiliging van deze systemen dat een grote rol speelt. Het zijn niet enkel banken die van deze technologie gebruik maken, maar ook verzekeringsmaatschappijen bijvoorbeeld. De top 10 van deze soort bedrijven maken allemaal gebruik van een IBM Mainframe (Tozzi, 2022)

2.3. De Skill gap

Het is nog steeds moeilijk om nieuw talent aan te trekken in de mainframe wereld. Een onderzoek van Deloitte (2020) toont aan dat 79% van de projectleiders moeilijkheden heeft met het zoeken naar mensen met de juiste skillset. Hetzelfde onderzoek toont ook aan dat er in de teams zelf een groot verschil is van kennis en vaardigheden.
Hoewel dit systeem gebruikt wordt door 71% van de fortune 500 companies (Tozzi, 2022) , is de Skill gap nog steeds te groot waardoor veel bedrijven vrezen voor een groot tekort aan werknemers om deze Z Systems te onderhouden.
Volgens Petra Goude (2023) zijn er verschillende manieren om dit probleem aan te pakken. Zo kunnen bedrijven lessen geven over Mainframe en hoe ze dit gebruiken. Ze vertelt ook dat de vaardigheden die nodig zijn beter gecomplimenteerd moeten worden en kunnen leiden naar een belonende en lange termijn carrière.
Hoewel dit zou helpen, vind ze dit niet de kern van het probleem. Het zijn de oude technieken die mensen niet aantrekt. Ze stelt voor om meer te investeren in hedendaagse technologie en dit te installeren op de mainframe. Dit kan gaan over dezelfde test -en deployment technieken, maar ook over hedendaagse programmeertalen zoals Java of Python. Dit zou kunnen door middel van APIs en zou een nieuwe, jongere werkkracht aantrekken.

2.4. Toch nog een kleurrijke toekomst

Ondanks deze probleemstelling, ziet de toekomst er toch nog goed uit voor deze systemen. Zo wordt er meer gemoderniseerd met bijvoorbeeld modernere programmeertalen. Er wordt ook voorspeld dat we een introductie van DevOps en self-service benaderingen gaan zien. (Pennaz, 2023)
Momenteel zijn Python en Java beschikbaar als programmeertaal op de mainframe naast PL/1 en COBOL. Sinds deze introductie zijn al bijna 2/3de van gebruikers op

de mainframe Java aan het toepassen op een bepaalde manier. (Watts, 2018)

3

Methodologie

3.1. Fase 1: Literatuurstudie

- **Doel:** Relevante informatie verzamelen van IDz, Pydev en de Unix System Services omgeving.
- **Aanpak:**
  - Opzoeken betrouwbare bronnen
  - Gelijke projecten onderzoeken
  - Overzicht van nodige software
  - Opmaken stappenplan voor installatie Pydev
- **Tijd:** 4 weken
- **Opbrengst:** Een volledig onderzoek naar vakliteratuur en nodige software die nodig is. Ook een volledige schets naar de omgeving waarin er wordt gewerkt.

3.2. Fase 2: Opstellen Pydev

- **Doel:** Opzetten van de Pydev plugin in IDz
- **Aanpak:**
  - De opgezochte literatuur bestuderen
  - Pydev opstellen met de juiste vereisten
  - Testen
- **Tijd:** 2 weken

- **Opbrengst:** Pydev voor functies zoals code completion, syntax check, code analysis, ...
  Bij het openen van een Python programma zal dit ook gebeuren via een Python editor

3.3. Fase 3: Python interpreter opzetten

- **Doel:** Python Interpreter opstellen om Python programma's uit te voeren in IDz.
- **Aanpak:**
- De opgezochte literatuur bestuderen
- Opzetten van de juiste interpreter
- Configuratie om output van de uitgevoerde code in de IDz console te kunnen zien
- Testen
- **Tijd:** 2 weken
- **Opbrengst:** Python programma's kunnen uitvoeren in IDz.

3.4. Fase 4: Analyse functies van Pydev

- **Doel:** Onderzoeken welke functies er beschikbaar zijn in Pydev
- **Aanpak:**
- Pydev documentatie bekijken
- Zelf onderzoeken in IDz
- **Tijd:** 4 week
- **Opbrengst:** Een overzicht van alle functies die beschikbaar zijn.

3.5. Fase 5: Evaluatie voorbereiding

- **Doel:** Eind evaluatie voorbereiden
- **Aanpak:**
- Resultaat bespreken
- Op orde stellen van nodige documenten
- **Tijd:** 2 weken
- **Opbrengst:** Een eind evaluatie waarin de opdracht besproken wordt

4

Literatuurstudie

4.1. De IBM Z Systems omgeving

Het is belangrijk om te weten wat een mainframe is en wat de hoofdpunten van de technologie zijn. Het is moeilijk om een goede definitie te plakken op deze term maar het is ontwikkelt door IBM en het wordt vooral gebruikt door grote bedrijven om belangrijke applicaties te hosten of veel transacties te kunnen doorvoeren. Hoewel dit ook mogelijk is op een kleinschalige server, zou het resultaat niet hetzelfde zijn omdat Mainframes miljarden transacties per dag zou kunnen uitvoeren zonder enige vertraging. (BasuMallick, 2023)

De Mainframe wordt vaak in vergelijking gebracht met een traditionele server die je vind in een datacenter. Deze vergelijking is niet onterecht omdat ze wel een gelijkaardige functie hebben. Een Mainframe zoals de hedendaagse IBM z16 heeft de mogelijkheid 19 miljard transacties per dag uit te voeren, wat ook verklaard waarom deze systemen gebruikt worden door 92 van de top 100 banken ter wereld. De z16 heeft ook 40 terabyte werkgeheugen wat 1200 keer het aantal is in hedendaagse high-performance computers. (Tozzi, 2022)

De z16 is ook 'Quantum safe': de bedreiging van Quantum computers in de toekomst blijft groeien en er is nog geen directe oplossing voor. IBM heeft hiervoor geïnvesteerd in Crypto Express 8S hardware security modules om data op de Mainframe te beschermen en Quantum safe te maken. De nieuwe modules bevatten nieuwe quantum safe encryptie algoritmes die geëvalueerd zijn door de US National Institute of Standards and Technology. (**Sayer2022**)

4.1.1. Het hoofdbesturingssysteem z/OS

Een IBM Mainframe ondersteund meerdere besturingssystemen maar de meest gebruikte is z/OS. Dit is, samen met de IBM Z Systems, ontwikkeld door IBM waardoor dit het meest compatibel is met de hardware componenten en het blijft ondersteund door IBM zelf. Deze heeft verschillende karakteristieken zoals workload manager (WLM) om het uitvoeren van jobs te plannen. Dit besturingssysteem kan gezien worden als hybride omdat het moderne taken van andere besturingssystemen neemt en combineerd met de architectuur van een IBM Mainframe. Het heeft ook de mogelijkheid om terug te draaien naar een vorige versie zonder dat dit problemen zal veroorzaken met het systeem. (Rupp, 2022)

4.1.2. Software in z/OS

Dit besturingssysteem bevat tools zoals TSO, ISPF en SDSF om er een paar op te noemen.

TSO staat voor Time Sharing Option en is in principe de command line interface op de mainframe. Dit laat meerdere gebruikers toe om een interactieve sessie op te starten met z/OS via hun eigen inloggegevens. Zoals een traditionele CLI bestaat dit uit een prompt waar je commando's kunt ingeven om acties uit te voeren op het systeem. In TSO wordt dit een 'READY' prompt genoemd omdat het een READY melding geeft als je commando's kunt invoeren. (IBM, z.d.-c)

Hoewel TSO vrij krachtig is, zal dit door de meeste eindgebruikers gebruikt worden in combinatie met ISPF of *Interactive System Productivity Facility*. Dit is een GUI dat bestaat uit menu's en panelen die allemaal verschillende functies kunnen uitvoeren (IBM, z.d.-c). Veel gebruikte functies zijn het aanmaken en schrijven van COBOL of PL/1 programma's. ISPF biedt nog meer verschillende functies zoals het aanmaken van datasets tot een connectie maken met de database op de mainframe. Hierin kun je dus eigenlijk alles doen wat het platform te bieden heeft maar wel in de lijnen van de rechten die je hebt als gebruiker.

System Display and Search Facility of SDSF is volgens de IBM (2023) documentatie een interface om jobs en hun output te kunnen zien. Zo is er ook de mogelijkheid om een job te doen stoppen, bijhouden of vrijgeven. Met bijhouden bedoelen we niet laten uitvoeren tot iemand een teken geef dat het uitgevoerd mag worden. Vrijgeven wordt gebruikt om de fysieke middelen zoals CPU percentage vrij te geven zodat een andere job deze kan gebruiken.
Het biedt ook informatie over het z/OS systeem zodat u dit kan monitoren, managen en controlleren. Hoewel het vooral gebruikt wordt om de status van een uitgevoerde job te zien, wordt deze tool ook gebruikt om fysieke toestellen te controleren (zoals een printer), de fysieke middelen zoals CPU of geheugen beheren

en de system log en messages bekijken.

4.1.3. Datasets

z/OS heeft ook een andere manier om data op te slaan door middel van datasets. Dit is volgens de documentatie van IBM (z.d.-c) een collectie van gerelateerde data records dat opgeslagen en opgehaald wordt door een toegewezen naam. Dit zou u kunnen zien als een bestand in andere besturingssystemen zoals in Windows of Linux.

Hier bestaan er verschillende soorten van:

- Sequentiële dataset (Seq)
- Partitionele dataset (Pds)
- Virtual storage access method (VSAM)

Sequentiële datasets bevatten eigenlijk records die achter elkaar opgeslagen zijn. Dit heeft als nadeel dat als je bijvoorbeeld record 20 wilt lezen, je eerst voorbij de voorgaande 19 records moet gaan. Dit is vooral nuttig om grote hoeveelheden data in op te slaan.

Een partitionele dataset is meer voor programma's in te schrijven. Deze dataset bevat allemaal 'members' die op hun beurt dan de effectieve data bevatten. Dit heeft als voordeel dat je in een pds de members kunt aanspreken in een willekeurige volgorde. Deze vorm van een dataset wordt ook wel een library genoemd.

Een VSAM biedt een complexere manier van toegang tot verschillende soorten data en is vooral bedoeld voor applicaties. Door hun complexiteit kunnen ze niet frequent bekeken of aangepast worden in ISPF in tegenstelling tot een Pds. Een VSAM kun je verdelen in 4 verschillende datasets:

- Key Sequence Data Set (KSDS):
  Dit is het meest voorkomend en slaat data op op basis van een key, value systeem
- Entry Sequence Data Set (ESDS):
  Dit houdt de records in een sequentiële volgorde bij en worden ook gelezen in deze volgorde. Dit is vooral gebruikt door IMS, DB2 en z/OS Unix.
- Relative Record Data Set (RRDS):
  Dit houdt records bij die je kunt ophalen op basis van een nummer. Zo heb je toegang tot records die opgeslagen zijn op plaats 100 zonder dat je door de eerste 99 records moet gaan. Dit is te vergelijken met een KSDS.

- Lineair Data Set (LDS):
  Dit houdt data bij in een byte stream en is de enige vorm van dit soort in een traditionele z/OS file

4.1.4. Unix op een mainframe

IBM heeft veel ingezet op modernisatie. Zo is er een Unix Systems Services (USS) omgeving bijgekomen die geïntegreerd is in het traditionele besturingssysteem z/OS. Er is ook een volledige Mainframe die enkel Linux heeft als besturingssysteem namelijk de LinuxOne. In dit onderzoek zal er enkel gekeken worden naar een z15 die een USS omgeving heeft.

Omdat de USS dus samenwerkt met z/OS heb je veel meer functies die je kunt gebruiken. Zo kun is er de mogelijkheid op XML parsing, OpenSSH, de IBM HTTP Server for z/OS, de z/OS SDK for Java en nog veel meer. (Dhawan, 2013)

Dit systeem biedt een hierarchische bestandssysteem (HFS) samen met een zSeries bestandssysteem (zFS). (Precisely, 2020) De HFS is wel bekend voor de meeste UNIX gebruikers: Dit is een hierarchie van directories met bestanden of subdirectories die grafisch weergegeven kunnen worden in een tree view (HCL Technologies, 2022).
zFS is iets minder bekend en kan gebruikt worden in pllaats van of als toevoeging op het traditionele HFS. Dit heeft vooral zijn waarde door zijn sterke performantie in bestanden die vaak worden gebruikt. Het verminderd ook het risico van het verlies in updates omdat het data asynchroon schrijft in plaats van te wachten op een sync interval. Bestanden in dit systeem kunnen aangepast worden door middel van een Application Programming Interface (API) en kunnen zelf in de HFS toegevoegd worden zonder enige problemen. (IBM, 2012)

Unix bestanden op de mainframe worden bijna op dezelfde manier gebruikt als op een traditioneel unix systeem. Het kan een Java, C++ of Python programma's bevatten. Deze programma's kunnen ook bestanden schrijven of schrijven in bijvoorbeeld een JSON of YAML formaat. Die kunnen op hun beurt dan gebruikt worden om analyses te doen op bepaalde data. Het hangt dus allemaal af van de use case om te zien op welke manier deze unix bestanden gebruikt worden. (Precisely, 2020)

4.1.5. Andere besturingssystemen

Zoals eerder vermeld is z/OS niet het enige besturingssysteem dat beschikbaar is op de mainframe. Hoewel dit door IBM aangeboden wordt, zijn er andere mogelijkheden die elk hun eigen sterktes en zwaktes hebben.

- z/Virtual Machine (z/VM)
  Dit is een hypervisor type 1 en kan gebruikt worden om meerdere besturingssystemen in te hosten. Dit bestaat uit een control program (cp) en een conversation monitoring system (CMS). De cp is verantwoordelijk voor het creëren van meerdere virtuele machines op basis van de fysieke hardware middelen. Het zorgt ook voor data en applicatie beveiliging voor alle systemen die in het systeem zitten. De CMS zit in een eigen virtuele machine biedt een interactieve sessie tussen de andere virtuele machines en de eindgebruikers. (IBM, z.d.-a)

- z/Virual Storage Extended (z/VSE)
  Dit besturingssysteem is vooral nuttig voor kleinere bedrijven die geen complexe batch -of transactie jobs moeten processen. Het is mogelijk dat naarmate ze groeien, ze overgaan naar z/OS als z/VSE niet genoeg blijkt te zijn. Het design van dit besturingssysteem maakt het perfect voor meer routine batch jobs parallel uit te voeren. Meestal wordt z/VM ook gebruikt als een teminal interface voor development en systeembeheer in z/VSE. (IBM, z.d.-a)

- Linux for System z

- z/Transaction Processing Facility (z/TPF)

4.2. IBM Developer for z/OS

De testomgeving waarin we zullen werken is IBM Developer for z/OS (IDz) versie 16.0.2. Dit programma is volgens de definitie van Spohn (2023) een toolset voor het ontwikkelen en opzetten van hybride cloud applicaties op z/OS.
Het is een eclipse based programma met de mogelijkheid hebt om een connectie te maken met verschillende omgevingen op de mainframe (bv de USS of z/OS omgeving). Zo kunt u alle bestanden zien, openen en aanpassen. Omdat de data nogsteeds op de mainframe staat, kunnen wijzigingen direct gezien worden ookal bekijkt u het in een ander programma. Als u bijvoorbeeld een bestand wijzigt via IDz, kunt u de wijzigingen direct zien in ISPF.
Dit programma is vooral ontwikkelt om een envoudigere IDE te bieden om in te programmeren aangezien niet iedereen direct bekend is met ISPF.

Zoals vermeld zal er gebruik worden gemaakt van IDz versie 16.0.2 . In het overzicht van IBM (2024b), ondersteund deze versie syntax veranderingen voor COBOL 6.4, PL/1 6.1 en REXX. Er is ook een ZUnit update wat vooral het gebruik van deze tool

makkelijker maakt.

ZUnit staat voor z/OS Automated Unit Testing en is een framework dat gebruikt wordt in IDz om COBOL en PL/1 programma's te testen. Hiervoor maakt het gebruik van verschillende 'samples' die door IBM zijn ontworpen (bv. Enterprise COBOL CALL02.cbl sample test case). Het gebruik van deze tool maakt het makkelijker voor developers om hun code (geschreven in COBOL of Pl/1) te testen op de mainframe. Dit maakt het schrijven van code in IDz veel efficiënter. (IBM, 2024a)

4.2.1. Wizards

4.2.2. Open source

4.3. IBM Z in een bankomgeving

Dit onderzoek wordt uitgevoerd in een bankomgeving dus is het wel interessant om te bekijken welke voordelen deze technologie heeft in zo een omgeving.
Volgens Turner (2022) gebruiken de meeste banken een IBM mainframe omdat ze de rekenkracht kunnen bieden die banken nodig hebben om efficiënt te kunnen werken. Kenmerken zoals robuustheid, betrouwbaarheid en snelle processing kracht spelen ook een grote rol aangezien het van groot belang is dat het systeem bijna altijd actief moeten zijn. De mainframe toont hier zijn sterkte door de 8 nines oftwel 99,999999% van de tijd beschikbaar per jaar. (IBM, z.d.-b)

5

Opzetten Pydev

Het programma waarin het onderzoek uitgevoerd zal worden is IBM Developer for z/OS (IDz) versie 16.0.2 met voorgeïnstalleerde programma's van DNB zelf. IDz is gebaseerd op Eclipse versie 4.23.0 .
In dit onderzoek zullen we versie 8.2.0 van PyDev installeren. Dit is een iets oudere versie maar het best compatibel met IDz, de andere versies kunnen problemen met zich meebrengen tijdens en na de installatie. Versie 8.2.0 bevat onder andere een debugger, een code formatter en code analyse. Het is belangrijk om te weten dat IDz door IBM beschikbaar wordt gesteld en ze dus support zullen bezorgen waar nodig moesten er problemen zijn met hun programma. Dit is niet het geval met PyDev omdat dit door een externe partij is ontwikkeld en dus niet ondersteund wordt door IBM.

5.1. Vereisten voor installatie

- Eclipse versie 4.6 based programma nodig - IDz
- Java 11 of hoger
- Python 2.6 of hoger

5.2. Installatie

Om Pydev te gebruiken moeten we het eerst installeren via hun Github repository https://github.com/fabioz/Pydev/releases.
Dit zal een .zip installeren die je moet uitpakken en ergens moet opslaan. Het uitpakken kan een enige tijd duren.

In IDz navigeer je naar *help -> install New Software*

Je zult een installatie wizard te zien krijgen en hierin klik je op *add -> local*

Selecteer de map waar Pydev in is uitgepakt. klik op *add*

Je krijgt weer een scherm te zien met de opties die je kunt installeren. Het kan zijn dat het eerst leeg is en dit verander je door in de checkbox de Group items by category uit te vinken. Hierna krijg je 3 opties.

- PyDev for Eclipse 8.2.0.202102211157
- PyDev for Eclipse Developer Resources 8.2.0.202102211157
- PyDev Mylyn Integration 0.6.0

De eerste is noodzakelijk en de tweede optioneel. De derde mag niet aangevinkt worden aangezien MyLyn niet is geïnstalleerd in IDz waardoor het opzetten van PyDev zal falen.

Klik op *next*, dit zal alles opzetten en kan even duren.

Als dit gedaan is krijg je een overzicht van de installatie details en hier klik je op *finish*.

In de balk rechtsonder kun je zien dat het bezig is met installeren.

Als het klaar is met installeren zal het een pop-up geven om IDz herop te starten. Als u nog open projecten heeft is het best om deze eerst op te slaan en dan zelf te herstarten. Anders klikt u op *Restart Now*

Om te zien of de installatie gelukt is navigeert u naar *help -> about -> Installation details*

Geef in de zoekbalk “Pydev”. Hier zou u de geïnstalleerde software moeten zien

Als je een Python script opent zal het nogsteeds gebeuren in de interne text editor. Om dit te wijzigen moeten we de file associations bekijken. Deze zullen bepalen welke editor er gebruikt wordt bij een bepaalde extensie. Hier zullen we dus de Python editor van Pydev linken aan de .py extensie.

Ga naar *Window -> Preferences -> General -> Editors -> File Associations*

Hier is een lijst met allemaal extensies en het programma waarmee ze geopend worden. IDz heeft niet standaard een python editor dus deze moeten we toevoegen. Klik bij “file types” op *add*

Geef hier .py in en klik op ok

In de lijst van extensies en ziet hier normaal .py tussenstaan

Selecteer de py extensie en bij “associated editors” komen er 3 opties:

- Python Editor
- Text Editor
- Generic Text Editor

Selecteer “Python Editor” en klik op *Default*.

Als je een python script opent zal dit automatisch gebeuren in de python editor en heb je alle functies die Pydev te bieden heeft.

6

Analyse Pydev

7

Conclusie

Curabitur nunc magna, posuere eget, venenatis eu, vehicula ac, velit. Aenean ornare, massa a accumsan pulvinar, quam lorem laoreet purus, eu sodales magna risus molestie lorem. Nunc erat velit, hendrerit quis, malesuada ut, aliquam vitae, wisi. Sed posuere. Suspendisse ipsum arcu, scelerisque nec, aliquam eu, molestie tincidunt, justo. Phasellus iaculis. Sed posuere lorem non ipsum. Pellentesque dapibus. Suspendisse quam libero, laoreet a, tincidunt eget, consequat at, est. Nullam ut lectus non enim consequat facilisis. Mauris leo. Quisque pede ligula, auctor vel, pellentesque vel, posuere id, turpis. Cras ipsum sem, cursus et, facilisis ut, tempus euismod, quam. Suspendisse tristique dolor eu orci. Mauris mattis. Aenean semper. Vivamus tortor magna, facilisis id, varius mattis, hendrerit in, justo. Integer purus.

Vivamus adipiscing. Curabitur imperdiet tempus turpis. Vivamus sapien dolor, congue venenatis, euismod eget, porta rhoncus, magna. Proin condimentum pretium enim. Fusce fringilla, libero et venenatis facilisis, eros enim cursus arcu, vitae facilisis odio augue vitae orci. Aliquam varius nibh ut odio. Sed condimentum condimentum nunc. Pellentesque eget massa. Pellentesque quis mauris. Donec ut ligula ac pede pulvinar lobortis. Pellentesque euismod. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent elit. Ut laoreet ornare est. Phasellus gravida vulputate nulla. Donec sit amet arcu ut sem tempor malesuada. Praesent hendrerit augue in urna. Proin enim ante, ornare vel, consequat ut, blandit in, justo. Donec felis elit, dignissim sed, sagittis ut, ullamcorper a, nulla. Aenean pharetra vulputate odio.

Quisque enim. Proin velit neque, tristique eu, eleifend eget, vestibulum nec, lacus. Vivamus odio. Duis odio urna, vehicula in, elementum aliquam, aliquet laoreet, tellus. Sed velit. Sed vel mi ac elit aliquet interdum. Etiam sapien neque, convallis et, aliquet vel, auctor non, arcu. Aliquam suscipit aliquam lectus. Proin tincidunt magna sed wisi. Integer blandit lacus ut lorem. Sed luctus justo sed enim.

Morbi malesuada hendrerit dui. Nunc mauris leo, dapibus sit amet, vestibulum et, commodo id, est. Pellentesque purus. Pellentesque tristique, nunc ac pulvinar adipiscing, justo eros consequat lectus, sit amet posuere lectus neque vel augue. Cras consectetuer libero ac eros. Ut eget massa. Fusce sit amet enim eleifend sem dictum auctor. In eget risus luctus wisi convallis pulvinar. Vivamus sapien risus, tempor in, viverra in, aliquet pellentesque, eros. Aliquam euismod libero a sem. Nunc velit augue, scelerisque dignissim, lobortis et, aliquam in, risus. In eu eros. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Curabitur vulputate elit viverra augue. Mauris fringilla, tortor sit amet malesuada mollis, sapien mi dapibus odio, ac imperdiet ligula enim eget nisl. Quisque vitae pede a pede aliquet suscipit. Phasellus tellus pede, viverra vestibulum, gravida id, laoreet in, justo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Integer commodo luctus lectus. Mauris justo. Duis varius eros. Sed quam. Cras lacus eros, rutrum eget, varius quis, convallis iaculis, velit. Mauris imperdiet, metus at tristique venenatis, purus neque pellentesque mauris, a ultrices elit lacus nec tortor. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent malesuada. Nam lacus lectus, auctor sit amet, malesuada vel, elementum eget, metus. Duis neque pede, facilisis eget, egestas elementum, nonummy id, neque.

A

Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage.

Bibliografie

BasuMallick, C. (2023). What Is a Mainframe? Features, Importance, and Examples. Verkregen maart 28, 2024, van https://www.spiceworks.com/tech/tech-101/articles/what-is-mainframe/

Deloitte. (2020, juni 1). *Mainframe talent drain* (Survey). https://www2.deloitte.com/content/dam/Deloitte/us/Documents/Alliances/us-alliances-deloitte-ibm-mainframe-market-pulse-survey.pdf

Dhawan, A. (2013). What is USS? Verkregen mei 3, 2024, van https://www.zmainframes.com/viewtopic.php?t=49

Goude, P. (2023). 5 ways to bridge the Mainframe skills gap. Verkregen december 13, 2023, van https://www.informationweek.com/it-infrastructure/5-ways-to-bridge-the-mainframe-skills-gap

HCL Technologies. (2022). HCL Z Data Tools User's Guide and Reference. Verkregen mei 3, 2024, van https://help.hcltechsw.com/zdt/1.1.0/en/base/hfs.html#:~:text=z/OS%C2%AE%20UNIX%E2%84%A2%20provides%20a%20Hierarchical%20File%20System,contain%20files%20or%20other%20subdirectories.

IBM. (z.d.-a). Mainframe Concepts. Verkregen mei 3, 2024, van https://www.ibm.com/docs/en/zosbasics/com.ibm.zos.zmainframe/zmainframe_book.pdf

IBM. (z.d.-b). Resilient server solutions with IBM Z. Verkregen mei 2, 2024, van https://www.ibm.com/z/resiliency

IBM. (z.d.-c). z/OS concepts. Verkregen april 26, 2024, van https://www.ibm.com/docs/en/zosbasics/com.ibm.zos.zconcepts/zconcepts_book.pdf

IBM. (2012). z/OS Distributed File Service zSeries File System Implementation z/OS V1R13. Verkregen mei 3, 2024, van https://www.redbooks.ibm.com/redbooks/pdfs/sg246580.pdf

IBM. (2023). Introduction to SDSF. Verkregen mei 2, 2024, van https://www.ibm.com/docs/en/zos/2.5.0?topic=guide-introduction-sdsf

IBM. (2024a). Unit testing Enterprise COBOL and PL/I applications. Verkregen mei 2, 2024, van https://www.ibm.com/docs/en/developer-for-zos/16.0?topic=eclipse-unit-testing-enterprise-cobol-pli-applications

IBM. (2024b). What's new in IBM Developer for z/OS and Developer for z/OS Enterprise Edition. Verkregen mei 2, 2024, van https://www.ibm.com/docs/en/developer-for-zos/16.0?topic=zos-whats-new-in-developer

Johnson, A. (2023). Python Popularity: The Rise of A Global Programming Language. Verkregen december 13, 2023, van https://flatironschool.com/blog/python-popularity-the-rise-of-a-global-programming-language/#:~:text=Python%20Popularity%20Over%20The%20Years&text=Stack%20Overflow's%20developer%20survey%20results,as%20the%20most%20popular%20language.

Klaey, W. (2023). Python Programming on z/OS. Verkregen december 13, 2023, van https://www.linkedin.com/pulse/python-programming-zos-walter-klaey/?trackingId=eYKhSv70Tc6dnZLPooMDYQ==

Mertic, J. (2020). Getting Jiggy With It: How a Linux Anniversary Led to the Creation of the Open Mainframe Project. Verkregen december 12, 2023, van https://newsroom.ibm.com/How-a-Linux-Anniversary-Led-to-the-Creation-of-the-Open-Mainframe-Project

Pennaz, M. (2023). The secret to attracting mainframe talent during a skills crisis. Verkregen december 13, 2023, van https://venturebeat.com/data-infrastructure/secret-to-attracting-mainframe-talent-during-skills-crisis/

Precisely. (2020). The Ultimate Guide to Mainframe Machine Data: SMF Data Beyond. Verkregen mei 3, 2024, van https://www.precisely.com/blog/mainframe/ultimate-guide-mainframe-machine-data-smf

Rupp, M. (2022). AN INTRODUCTION TO Z/OS AND THE IBM COMMON CRYPTOGRAPHIC ARCHITECTURE. Verkregen mei 2, 2024, van https://www.cryptomathic.com/news-events/blog/payment-banking-an-introduction-to-z/os-and-the-ibm-common-cryptographic-architecture

Spohn, J. (2023). IBM Developer for z/OSEnterprise Edition. Verkregen maart 28, 2024, van https://www.ibm.com/downloads/cas/DJ5P7W3N

Tozzi, C. (2022). 10 Mainframe Statistics That May Surprise You. Verkregen december 13, 2023, van https://www.precisely.com/blog/mainframe/9-mainframe-statistics

Turner, D. (2022). PAYMENT BANKING CRYPTOGRAPHY: THE BENEFITS OF Z/OS THE Z PLATFORM. Verkregen mei 2, 2024, van https://www.cryptomathic.com/news-events/blog/payment-banking-cryptography-an-overview-of-the-benefits-of-z/os-and-the-z-platform

Wagle, L. (2017). The modern mainframe. Verkregen december 12, 2023, van https://www.ibm.com/thought-leadership/institute-business-value/en-us/report/modernmainframe

Watts, S. (2018). Java on the Mainframe: z/OS vs Linux. Verkregen december 13, 2023, van https://www.bmc.com/blogs/java-mainframe-zos-linux/