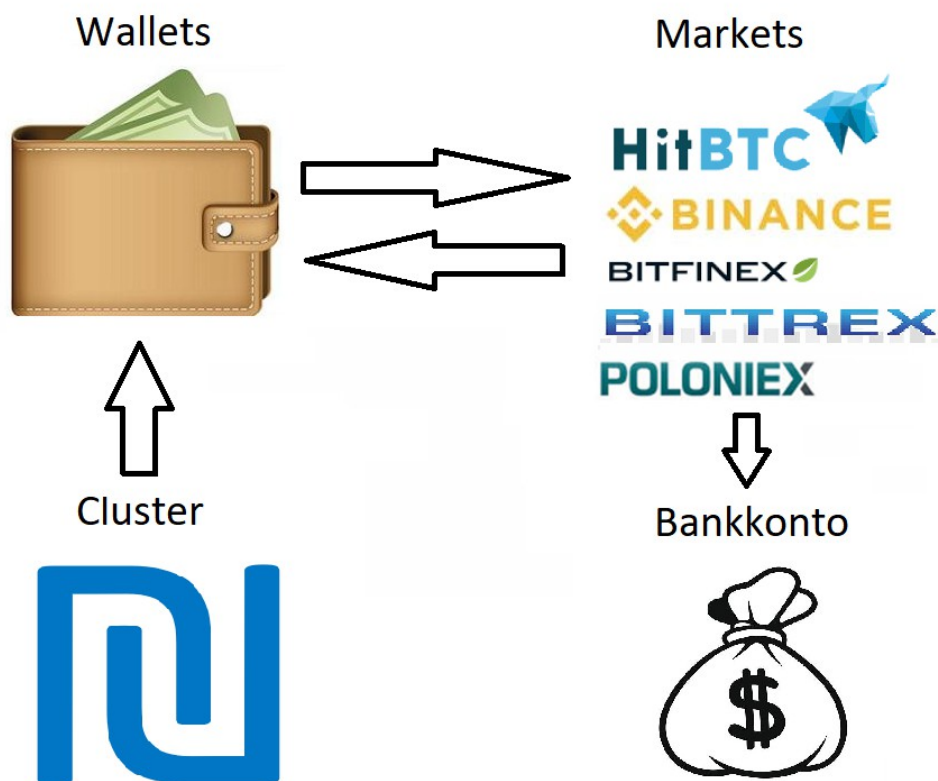


Nebula: Mining Cluster

Christoph Amrein



Management Summary

Inhaltsverzeichnis

1	Initialisierung	1
1.1	Ausgangslage	1
1.1.1	Weshalb soll das Projekt realisiert werden?	1
1.1.2	Für wen ist das Projekt gedacht?	1
1.2	Situationsanalyse	1
1.2.1	Stärken	2
1.2.2	Schwächen	2
1.3	Ziele	2
1.3.1	Vorgehensziele	2
1.3.2	Projektziele	3
1.3.3	Lieferobjekte	4
1.3.4	Rahmenbedingungen	4
1.3.5	Abgrenzungen	4
1.4	Lösungsbeschreibung	5
1.5	Kosten	5
1.5.1	Einmalige Kosten	5
1.5.2	Betriebskosten (Repetitiv)	6
1.5.3	Gesamtkosten	6
1.6	Wirtschaftlichkeit	6
1.6.1	Spekulation	7
1.6.2	Infrastruktur	7
1.7	Planung	8
1.7.1	Grober Projektplan	8
1.7.2	Termine	9
1.8	Ressourcen	9
1.8.1	Budget	9
1.8.2	Sachmittel	9
1.9	Organisation	10
1.9.1	Projektorganisation	10
1.9.2	Projektablage	10
1.10	Cluster-Software evaluation	11
1.10.1	Cluster Software Kriterien	11
1.10.2	Informationsbeschaffung	11
1.11	Lösungsvarianten	12
1.11.1	Variantenübersicht	12
1.11.2	Variante V1 «OpenHPC»	12

1.11.3	Variante V2 «TinyTitan»	12
1.11.4	Variante V3 «Minimale Lösung»	13
1.11.5	Anforderungsabdeckung der Varianten	13
1.12	Bewertung der Varianten	14
1.13	Variantenentscheid	14
1.14	Risiken	14
2	Konzept	16
2.1	Physikalischer Überblick	16
2.2	Physikalische Verbindungen	17
2.2.1	Stromversorgung Managementnode	17
2.2.2	Computenodes	17
2.2.3	übrige Geräte	17
2.2.4	Netzwerkverbindungen	17
2.3	Technischer Überblick	18
2.3.1	Verwendete Protokolle	19
2.4	Technische Verbindungen & Kommunikation	19
2.5	Komponentenbeschreibung	20
2.5.1	Router	20
2.5.2	PC	20
2.5.3	Managementnode	20
2.5.4	Netzteil Managementnode	20
2.5.5	NAS	20
2.5.6	Switch	20
2.5.7	Computenodes	21
2.5.8	Schaltnetzteil Computenodes	21
2.6	Tests	21
2.6.1	Testobjekte	21
2.6.2	Testarten	22
2.6.3	Testvoraussetzungen	22
2.6.4	Fehlerklassen	23
2.6.5	Testhilfsmittel	23
2.6.6	Testfälle	23
2.7	Monitoring	28
2.7.1	Service Monitoring - Nagios	28
2.7.2	Performance Monitoring - Ganglia	28
2.8	Mining	28
2.8.1	Kryptowährungen	29
2.9	Hostnamen	29
2.9.1	Managementnode Name	29
2.9.2	Reservenode Name	29

2.9.3	Computenode Namen	30
3	Realisierung	31
3.1	Physischer Aufbau	31
3.1.1	Komponenten Platzierung	31
3.1.2	Kühlung	32
3.1.3	Stromversorgung	32
3.1.4	Kommunikation	32
3.2	Technischer Aufbau	32
3.2.1	Betriebssystem	32
3.2.2	Vorbereitungen	33
3.2.3	Installation	33
4	Schlussbetrachtung	33
4.1	Arbeiten nach dem Projekt	34
4.2	Persönliche Betrachtung	34
4.3	Danksagung	34
5	Authentizität	34
A	Anhang	35
B	Vorbereitung RPI's	35
B.1	Betriebssystem installieren	35
B.2	RPI für den Netzwerkboot vorbereiten	38
B.3	Hostname und IP der MAC Adresse zuweisen	38
C	Installation Managementnode	40
C.1	Variablen Definition	40
C.2	Basiskonfiguration	44
C.3	OpenHPC Komponenten installieren	44
C.4	Netzwerkboot einrichten	45
C.5	Registration Computenodes	49
C.6	Monitoring installieren	50
C.7	Miner installieren	51
C.8	Quellenverzeichnis	52
D	Diplomeingabe	53
E	Testprotokoll	56
E.1	Testobjekte	58
E.2	Arbeitsjournal	59
E.3	Protkolle	59
E.4	Datenblätter	60

Inhaltsverzeichnis

E.5	Produktinformationen	60
E.6	Benutzerdokumentation	60
Abkürzungsverzeichnis		60
Abbildungsverzeichnis		61
Tabellenverzeichnis		62
Listings		64

1 Initialisierung

1.1 Ausgangslage

Die aktuelle Umgebung ist nicht auf das Schürfen von Kryptowährungen ausgelegt. Das System hat eine Uptime von maximal 40%. Auch der Standort, der Lärm und die Hitze des Systems und Raumes werden als störend empfunden. Dadurch wurden hauptsächlich Tokens auf Börsen gekauft, welche nicht geschürft werden können. Die Sicherung der Daten ist ebenfalls nicht gewährleistet. Zudem existieren keine Monitoring Tools, welche den Status des Schürfens und der Hardware zu erkennen geben. Dabei wird das Analysieren von Problemen als schwierig erachtet, da keine Logdaten existieren, oder diese mit viel Aufwand zusammengesucht werden müssen.

1.1.1 Weshalb soll das Projekt realisiert werden?

Es soll eine stabile Lösung zum Schürfen von Kryptowährungen auf CPU Basis erschaffen werden, welche permanent in Betrieb sein kann und Profit generiert.

1.1.2 Für wen ist das Projekt gedacht?

Das Projekt wird in eigenem Interesse aufgebaut. Es existieren demnach keine Kunden und Abhängigkeiten zu anderen Personen oder Unternehmen.

1.2 Situationsanalyse

Für das Schürfen von Kryptowährungen werden folgende Komponenten eingesetzt:

Nr.	Typ	Komponente	Modell Version
1	HW	Prozessor (CPU)	Intel Core i7-4700, 3.40 GHz Quad Core
2	HW	Grafikkarte (GPU)	NVIDIA GeForce GTX 1070 Ti
3	HW	Festplatte (HDD)	TOSHIBA DT01ACA200
4	SW	Schürf-Software	Minergate, Version 7.2
5	SW	Betriebssystem (OS)	Windows 10 EDU, Version 1709

Tabelle 1: Situationsanalyse Komponenten

Legende: HW = Hardware, SW = Software

1 Initialisierung

1.2.1 Stärken

Nr.	Kategorie	Beschreibung
1	Bedienbarkeit	Das Schürfen der Währungen kann über ein GUI gestartet werden.
2	Wartung	Es existieren keine Umsysteme

Tabelle 2: Situationsanalyse Stärken

1.2.2 Schwächen

Nr.	Kategorie	Beschreibung
1	Flexibilität	Während des Schürfens, ist der Computer für andere Tätigkeiten blockiert.
2	Kosten	die Betriebskosten sind höher als der Ertrag
3	Betriebszeit	Es kann nicht durchgehend Kryptowährungen geschürft werden.

Tabelle 3: Situationsanalyse Stärken

1.3 Ziele

1.3.1 Vorgehensziele

Zeitplan

Während der Initialisierungsphase wurde eine Projektplanung mit den Aufgaben und den vorgesehenen Aufwänden während des Projektes erstellt. Die definierten Soll-Aufwände sollen mit den stetig nachgeführten IST-Aufwänden verglichen werden. Die Abweichungen werden im Projektplan direkt errechnet.

Meilensteine

Die Meilensteine wurden in der Zeitplanung des Projektes berücksichtigt und definiert. Die Aufwände werden jeweils im Projektplan nachgeführt, dies ermöglicht einen Ist- und Soll-Aufwand Vergleich

Arbeitsjournal

Das Arbeitsjournal wird alle 2 Wochen an die Experten versendet. Diese haben die Möglichkeit die Aufwände und investierte Zeit zu prüfen.

Beweiserbringung

Alle geleisteten Arbeiten sollen in dokumentarischer Form, Präsentation oder einem Gespräch bewiesen werden können.

1.3.2 Projektziele

Nr.	Ziel	Messgrösse	Kat.	Prio.
01	Die CPU des Clusters soll zu 90% zum Schürfen von Kryptowährung beansprucht werden	Log und Monitoring Auswertungen nach dem Testlauf	LZ	M
02	Die Daten werden auf einem NAS mit RAID I gesichert	Die Festplatten werden einzeln überprüft, der Datenbestand muss identisch sein	BZ TZ	M
03	Der Cluster soll eine Verfügbarkeit von 98% aufweisen	Dies kann erst nach dem Testlauf durch ein Monitoring der Laufzeit gemessen werden	LZ BZ TZ	M
04	Es können während des Betriebs neue Computenodes hinzugefügt werden & ausfallende Computenodes verursachen keinen Unterbruch des Betriebs	Während der Testphase werden neue Computenodes hinzugefügt und Computenodes vom Cluster getrennt	LZ BZ TZ	M
05	Der Cluster kann für verschiedene Anwendungsgebiete eingesetzt werden	Während der Testphase werden andere Applikationen welche die Cluster Ressourcen verwenden sollen installiert	BZ	M
06	Das Betriebssystem soll über das Netzwerk an die Computenodes verteilt werden um SD-Karten zu sparen und ein Betriebssystem zentral verwalten zu können	Wird während der Installation über Systemlogdateien ausgelesen und mit SSH-Zugriffen getestet.	WZ BZ TZ	M
07	Das Schürfprogramm soll automatisiert die gewinnbringendste Währung abbauen	Nach der Testphase werden die Logdateien und Wallets ausgewertet und mit Daten der Währungskurse abgeglichen	LZ WZ	K
08	Mit der geschürften Währung soll auf Börsen gehandelt werden können	Kann nach der Realisierung durch Transaktionslogdaten gemessen werden	LZ WZ	K
09	Die Wartungsarbeiten sollen pro Monat nicht mehr als 3 Stunden betragen	Wird durch ein Eingriffsprotokoll nach der Realisierungsphase festgehalten	BZ	K
10	Der Cluster soll einfach transportierbar und wiederaufbaubar sein	Der Cluster wird nach der Testphase physisch verschoben und neu aufgebaut, dabei wird die Zeit des Wiederaufbaus gemessen	TZ	K

Tabelle 4: Projektziele

Legende: LZ = Leistungsziel, WZ = Wirtschaftsziel, BZ = Betriebsziel, TZ = Technisches Ziel, M = Muss-Kriterium, K = Kann-Kriterium

1.3.3 Lieferobjekte

Folgende Dokumente werden während des Projektes erstellt und geliefert.

Nr.	Dokument	Phase	Termin
1	Projektplan	Initialisierung	15.02.2018
2	Projektlogo	Initialisierung	20.02.2018
3	Projektauftrag	Initialisierung	20.02.2018
4	Studie	Initialisierung	25.02.2018
5	Detaillkonzept	Konzept	17.03.2018
6	Testkonzept	Konzept	22.03.2018
7	Installationshandbuch	Realisierung	10.05.2018
8	Testprotokoll	Realisierung	03.05.2018
9	Betriebshandbuch	Realisierung	13.05.2018
10	Diplombericht	Einführung	22.05.2018

Tabelle 5: Lieferobjekte

1.3.4 Rahmenbedingungen

Folgende Bedingungen gelten für die Durchführung des Projektes:

- Dem Projekt stehen 294 Stunden Arbeitszeit zur Verfügung.
- Es wird nach der Projektmethode HERMES gearbeitet.
- Der Stundenansatz der involvierten Personen ist auf 120.00 CHF angesetzt.
- Neue Anforderungen werden erst nach dem Projektabschluss berücksichtigt.

1.3.5 Abgrenzungen

- Das Projekt wird für den privaten Nutzen durchgeführt.
- Die Experten sind in den Kostentrechnungen nicht berücksichtigt.
- Der Cluster wird aus Kosten- und Leistungsgründen nicht redundant aufgebaut.
- Neue Anforderungen können während des Projektes nicht berücksichtigt werden.
- Das Projektbudget kann aus finanziellen Gründen nicht erhöht werden.
- Defekte Computenodes werden während des Projektes nicht ersetzt.

1.4 Lösungsbeschreibung

Es wird physisch ein Cluster aus mindestens 40 aktiven RPI's aufgebaut. Der Cluster soll aus finanziellen Gründen mit möglichst wenigen Komponenten wie, Netzteile, USB Kabel & Speicherkarten in Betrieb genommen werden. Dabei wird das Betriebssystem zentral verwaltet und über das Netzwerk an die einzelnen Raspberry PI's verteilt. Zugleich wird zur Datensicherheit ein Netzwerkshare (NAS) mit RAID I installiert. Zusätzlich soll die rentabelste Währung automatisch für eine Woche geschürft werden, bevor eine erneute Prüfung auf die rentabelste Währung geschieht. Die geschürften Kryptowährungen werden jeweils in die entsprechenden verschlüsselten Wallets transferiert. Durch die geschürften Währungen soll an Börsen gehandelt werden können, welche es ermöglichen sollen Profit zu erzielen. Durch eine Monitoring, Alarming und Logdaten Lösung soll auf Missstände des Clusters aufmerksam gemacht werden. Die Tools bieten sich sogleich für eine Analyse der Probleme an und sind über einen Webbrowser aufrufbar.

1.5 Kosten

1.5.1 Einmalige Kosten

Beschaffungskosten

Anzahl	Komponente	Stückpreis(CHF)	Gesamtwert(CHF)
40	Raspberry PI Model B+	33.00	1320.00
1	Schaltnetzteil	229.00	229.00
1	TTL Serial Kabel	30.00	30.00
40	Patchkabel Cat. 5e	1.00	40.00
1	TP-Link Switch	220.00	220.00
1	Synology NAS DS216	600.00	600.00
1	Diverse Kabel, Schrauben	50.00	50.00
-	Total	-	2'489.00

Tabelle 6: Beschaffungskosten

Aufwandskosten

Stunden	Phase	Stundenansatz(CHF)	Gesamtkosten(CHF)
30	Initialisierung	120.00	3'600.00
50	Konzept	120.00	6'000.00
142	Realisierung	120.00	17'040.00
22	Einführung	120.00	2'640
25	Periodische Arbeiten	120.00	3'000.00
25	Reserve	120.00	3'000.00
294	Total	120.00	35'280.00

Tabelle 7: Aufwandskosten

1.5.2 Betriebskosten (Repetitiv)

Folgende Voraussetzungen sind für die folgenden Berechnungen definiert.

- 1000 Watt die Stunde kosten 0.2894 CHF.
- Ein Monat hat 30 Tage
- Ein Jahr hat 360 Tage

Wartungskosten

Pro Monat sind 3 Stunden Wartungsaufwand einzuberechnen, dadurch ergeben sich mit dem definierten Stundenansatz jährliche Wartungskosten von **4'320.00 CHF**.

Stromkosten Der Strom wird durch die BKW über den Vertrag Energy Blue bezogen.

Anzahl	Leistung	Kosten in CHF			
RPI	kW	Stunde	Tag	Monat	Jahr
40	0.4	0.11576	2.78	83.35	1000.17

Tabelle 8: Stromkostenrechnung

1.5.3 Gesamtkosten

Jährliche Kosten

Das Projekt ist Anfangs Juni abgeschlossen. Deshalb belaufen sich die Wartungs- und Stromkosten im 1. Jahr auf die Hälfte gegenüber den Folgejahren. Der Cluster soll innerhalb von 3 Jahren gewinnbringend wirken. Die jährlichen sowie täglichen Kosten sind unten zu entnehmen.

Kostengrund	Kosten 1.Jahr	Kosten 2.Jahr	Kosten 3.Jahr
Beschaffung	2'489.00	-	-
Aufwand	35'280.00	-	-
Wartungskosten	2'160.00	4'320.00	4'320.00
Stromkosten	500.00	1000.00	1000.00
Total	40'429.00	45'749.00	51'069.00

Tabelle 9: Gesamtkosten

Tägliche Kosten Auf 3 Jahre ausgerechnet muss täglich ein Ertrag von 47.29 CHF erwirtschaftet werden um die investierten Aufwände und Kosten zu decken.

1.6 Wirtschaftlichkeit

Das Projekt wird für den privaten Nutzen und aus eigenem Interesse aufgebaut, aus diesem Grunde ist die Wirtschaftlichkeit kein Kernpunkt des Clusters.

1.6.1 Spekulation

Das Ziel des Clusters ist es täglich **30 CHF** zu erwirtschaften. Dieser Wert ist nicht deckungsgleich mit den täglichen Kosten, soll aber über Marktspekulationen gedeckt werden. durch die volatilen Märkte sind Kursschwankungen in beide Richtungen möglich. Es ist jedoch davon auszugehen, dass die Währungen in Zukunft noch an Wert zulegen werden, sobald diese einmal als geltende Zahlungsmittel aufgenommen werden. Durch reine Betriebskosten ohne Spekulation ergibt sich ein tägliches Defizit von **17.29 CHF** welches einem Verlust von **36.56%** entspricht.

1.6.2 Infrastruktur

Beim Projekt wird der Fokus der Wirtschaftlichkeit hauptsächlich auf den Aufbau gelegt, hier gilt es möglichst wenige überflüssige Komponenten zu benutzen. Es wurde darauf geachtet, dass die Komponenten durch eine zentrale Stelle versorgt werden. Dabei werden die Raspberry PI's mit nur einem Netzteil versorgt und das Betriebssystem wird über das Netzwerk verteilt welches Speicherkarten einspart.

Anzahl	Komponente	Stückpreis in CHF	Gesamtwert in CHF
Standardlösung			700.00
4	USB-HUB 10 Ports	35.00	140.00
40	Mini-USB Kabel	6.00	240.00
40	MicroSD Karte	8.00	320.00
Projektlösung			268.00
1	Netzteil	230.00	230.00
1	MicroSD Karte	8.00	8.00
-	Diverse Stromkabel	30.00	30.00
Differenz der Lösungen			432.00

Tabelle 10: Wirtschaftlichkeit Hardware

Durch die vorgesehene Hardwarelösung können 432.00 CHF eingespart werden. Dies entspricht einer Einsparung von **261%**.

1.7 Planung

1.7.1 Grober Projektplan

Die folgende Tabelle zeigt die Arbeiten an, welche während des Projektes erledigt werden sollen. Ein grafischer und detaillierter Projektplan ist dem Anhang zu entnehmen.

Aufgabe	Start	Ende	Dauer in Stunden		
			Soll	Ist	Abw.
0.0 Initialisierung			30		
0.1 Projektplan erstellen	06.02.2018	15.02.2018	4		
0.2 Projektlogo erstellen	06.02.2018	20.02.2018	2		
0.3 Studie: durchführen	06.02.2018	20.02.2018	18		
0.4 Projektauftrag erstellen	20.02.2018	25.02.2018	5		
0.5 Diplombericht erstellen	25.02.2018	28.02.2018	4		
1.0 Konzept			50		
1.1 Zwischen-Meeting	01.03.2018	01.03.2018	6		
1.2 Detailkonzept erstellen	05.03.2018	17.03.2018	12		
1.3 Testkonzept erstellen	18.03.2018	22.03.2018	11		
1.4 Dokumenten Review	24.03.2018	26.03.2018	12		
2.0 Realisierung			142		
2.1 Physischer Aufbau	03.04.2018	07.04.2018	20		
2.2 Stromversorgung einrichten	08.04.2018	09.04.2018	8		
2.3 Raspberry PI's vorbereiten	17.04.2018	17.04.2018	4		
2.4 Netzwerkboot einrichten	21.04.2018	23.04.2018	8		
2.5 Cluster Software installieren	24.04.2018	25.04.2018	20		
2.6 Schürf Software installieren	26.04.2018	26.04.2018	12		
2.7 Entwickeln von Tools und Automatismen	02.04.2018	28.04.2018	30		
2.8 Monitoring einrichten	01.05.2018	10.05.2018	14		
2.9 Periodische Systemtests	10.04.2018	13.05.2018	7		
3.0 Installationshandbuch erstellen	02.04.2018	10.05.2018	8		
3.1 Testprotokoll erstellen	02.05.2018	03.05.2018	3		
3.2 Produktdokumentation	02.04.2018	10.05.2018	3		
3.3 Betriebshandbuch	01.05.2018	13.05.2018	5		
3.4 Freigabe zur Einführung	07.05.2018	15.05.2018	0		
4.0 Einführung			22		
4.1 Abschlussbericht	17.05.2018	22.05.2018	6		
4.2 Management Summary	20.05.2018	24.05.2018	8		
4.3 Vorbereitung Abschluss Meeting	22.05.2018	27.05.2018	3		
4.4 Drucken und Binden	24.05.2018	01.06.2018	2		
4.5 Abschluss-Meeting	02.06.2018	02.06.2018	2		
4.6 Projektabschluss	03.06.2018	03.06.2018	1		

Tabelle 11: Grober Projektplan

1 Initialisierung

1.7.2 Termine

Ereignis	Datum	Teilnehmer	Standort
Einmalige Ereignisse			
Kick-Off Meeting	05.02.2018	Projektleiter & Experten	Post IT, Zollikofen
Zwischenmeeting	01.03.2018	Projektleiter & Experten	GIBB (TSBE), Bern
Abgabe des Diplomberichts	01.06.2018	Projektleiter	-
Abschlussmeeting	07.06.2018	Projektleiter & Experten	GIBB (TSBE), Bern
Periodische Ereignisse			
Statusbericht	Monatlich	Projektleiter	-
Arbeitsjournal	2-Wöchentlich	Projektleiter	-

Tabelle 12: Termine

1.8 Ressourcen

1.8.1 Budget

Dem privaten Projekt steht ein Budget von 3'250 CHF zu. Die Aufwände werden hierbei nicht berücksichtigt, da keine Löhne bezahlt werden müssen.

Nr.	Verwendungszweck	Budget in CHF
1	Beschaffungen	3'000.00
2	Apéro	150.00
3	Drucken & Binden	100.00
	Total	3'250.00

Tabelle 13: Projektbudget

1.8.2 Sachmittel

Die aufgelisteten Komponenten werden für die Lösung benötigt.

Nr.	Anzahl	Komponenten	Modell / Spezifikationen
1	40	Mini Computer	Raspberry PI 3 Model B+
2	1	Schaltnetzteil	RSP-750-5, Mean Well
3	1	USB zu TTL Serial-Kabel	Adafruit USB zu TTL Serial Kabel, 75cm
4	40	Ethernetkabel	FTP Cat.5e Patchkabel
5	1	Switch	TL-SL3452 48-Port 10/100, TP-Link
6	1	Datenspeicher	Synology NAS DS218
7	*	Diverse Kabel & Befestigungsmaterialien	*

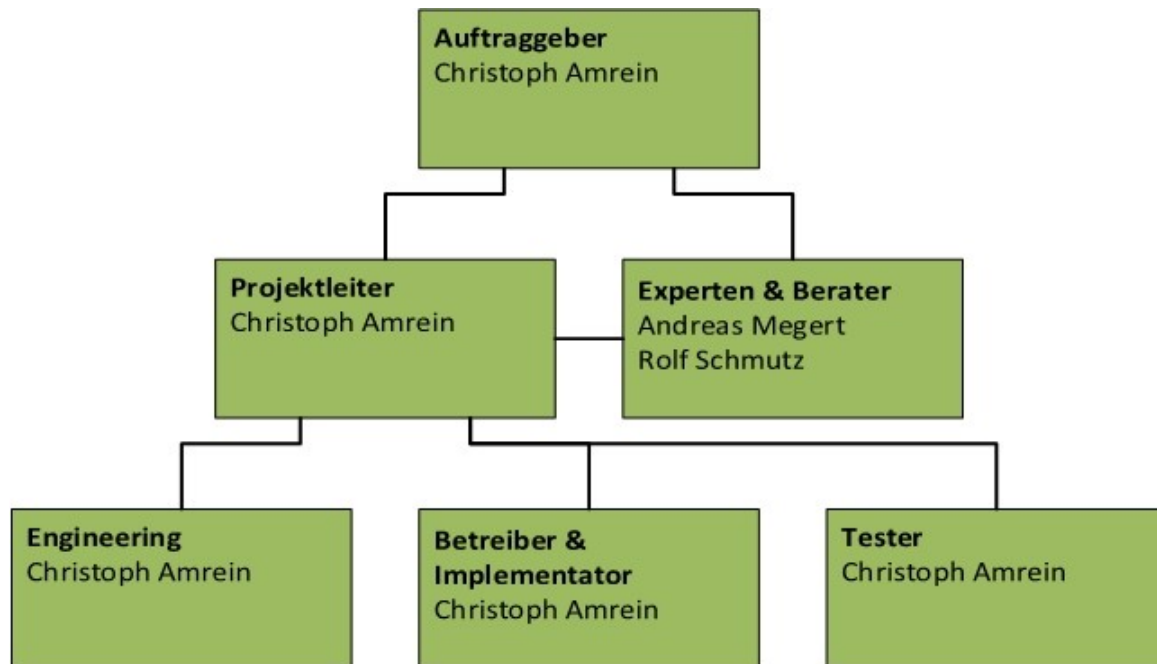
Tabelle 14: Sachmittel

* Anzahl und Hersteller unbekannt. Die Artikel wurden in lokalen Baumärkten eingekauft

1.9 Organisation

1.9.1 Projektorganisation

Die Projektorganisation ist wie folgt aufgebaut.



Rolle	Verantwortlichkeit
Auftraggeber	Erstellt den Auftrag und übergibt diesen an den Projektleiter
Projektleiter	Organisiert die Planung, Durchführung und präsentiert das Projekt.
Experte	Stehen in Kontakt mit dem Projektleiter und beraten ihn bei Schwierigkeiten
Engineering	Stellt dem Betreiber zu implementierende Applikationen zur Verfügung
Betreiber	Setzt die Lösung technisch um
Tester	Testet die Lösung auf Fehler

Tabelle 15: Organisation

1.9.2 Projektablage

Nr.	Was	Wo
1	Allgemeine Ablage	wiki.influ.ch
2	Dokumentation	https://github.com/amreinch/Nebula_AMC
3	Snapshots	Lokal, D:\Diplomarbeit\CentOS_works
4	Skripte, Entwürfe	https://github.com/amreinch/OpenHPC_Install_Nebula

Tabelle 16: Projektablage

1.10 Cluster-Software evaluation

1.10.1 Cluster Software Kriterien

Es wurden drei Cluster Software Produkte evaluiert, dabei mussten die Muss Kriterien erfüllt werden um in die Auswahl zu kommen. Diese Kriterien werden für den Entscheid der Software nicht berücksichtigt, grenzt aber die Auswahlmöglichkeit der Produkte ein.

Nr.	Anforderung	Prio.
01	Ist die Software HPC tauglich?	M
02	Kann das Produkt innerhalb des vorgesehenen Zeitraumes installiert werden?	M
03	Ist die Lösung skalierbar?	M
04	Existieren Dokumentationen?	S
05	Kann Support beansprucht und bezogen werden?	S
06	Ist die Lösung benutzerfreundlich?	S
07	existieren Verwaltungstools?	S
08	Fallen zusätzliche Kosten an?	S

Tabelle 17: Software Kriterien

1.10.2 Informationsbeschaffung

Es wurde nach einer Lösung gemäss der oben definierten Kriterien gesucht. Dabei bin ich auf den Wikipedia Eintrag https://en.wikipedia.org/wiki/Comparison_of_cluster_software gestossen, welcher die verschiedenen Cluster Software Angebote auflistet. Dabei wurde nur nach einer HPC Lösung gefiltert. Durch diese Analyse hat sich die OpenHPC Lösung der Linux Foundation herauskristalisiert. Weiterhin wurden Suchbegriffe wie "HPC Raspberry PI" über Suchmaschinen eingegeben da die Computenodes des Cluster Raspberry PI's sein sollen. Folgender Artikel habe ich als interessant erachtet und wurde genauer betrachtet. <http://www.hpctoday.com/best-practices/tinytitan-a-raspberry-pi-computing-based-cluster/> Mit einer weiteren Suche(hpc cluster software raspberry) bin ich auf einen Guide gestossen, der relativ simpel aussieht und einfach umzusetzen ist. <http://thundaxsoftware.blogspot.ch/2016/07/creating-raspberry-pi-3-cluster.html>. Es gab durchaus noch weitere Guides und Softwarelösungen, welche ich aber nach einer genaueren Analyse der Installationsanleitung verworfen habe, da diese mir zum Teil zu wenig Informationen lieferten. Während der Informationsbeschaffung wurden alle Installationsskripte und Anleitungen sorgfältig durchgelesen um diese als mögliche Variante zu empfehlen. Während der Informationsbeschaffung bin ich auf zwei Fachbegriffe (MPI & SLURM) welche meistens in Zusammenhang mit HPC stehen gestossen. Diese musste ich ebenfalls noch in Erfahrung bringen.

1.11 Lösungsvarianten

1.11.1 Variantenübersicht

Die Informationen wurden über die folgenden Produkte gesammelt und zusammengestellt:

Nr.	Variante	Bezeichnung
01	OpenHPC	HPC Lösung entwickelt von der Linux Foundation
02	TinyTitan	Open Source Lösung entwickelt von Oak Ridge Leadership Computing Facility
03	Minimale Lösung	Simple 32-Bit Architekturlösung

Tabelle 18: Variantenübersicht

1.11.2 Variante V1 «OpenHPC»

Beschreibung

OpenHPC gilt als vorangeschrittenes OpenSource Projekt der Linux Foundation. Das Produkt steht in direkter Verbindung mit diversen grossen IT Unternehmen weltweit. Das Ziel der Linux Foundation ist es, durch OpenHPC eine kostengünstige sowie schnell zu installierende HPC-Umgebung aufzubauen. Durch viele zusätzliche OpenSource Tools rundet sich das Produkt ab und gilt als ernstzunehmender Konkurrent gegenüber kostenpflichtiger Software.

Installation und Betrieb

Es existieren diverse Guides, Foren und Chats sowie eine E-Mail-Liste zu OpenHPC. Dadurch scheint die Unterstützung bei allfälligen Problemen vorhanden zu sein. Die Installationsanleitung, welche von der Linux Foundation geschrieben wurde, liest sich sehr gut und ist absolut ausreichend für die Installation. Der Betriebsaufwand wird als gering eingeschätzt, da es ein sehr ausgereiftes Produkt, welches stetig weiterentwickelt wird, ist.

Voraussetzungen, Abhängigkeiten

Für die Cluster Software werden mindestens ein Masternode und 4 Computenodes vorausgesetzt. Das Betriebssystem bezieht sich hierbei auf ein CentOS7x. Jeder Computenode benötigt 2 Netzwerkschnittstellen. Das eine Interface wird für den Standard Ethernet Zugriff verwendet und das zweite Interface wird für die Kommunikation zu jedem BMC Host verwendet. Es werden zusätzliche Intel Bibliotheken benötigt. Dazu müssen Lizenzen für Parallel Studio XE von Intel besorgt werden. Die Lizenzen können mit einer offiziellen E-Mail-Adresse der Schule gratis bezogen werden. Die Linux Foundation erwähnt in ihrem Guide, dass sie die «Bring your own Licence» Strategie verfolgt.

1.11.3 Variante V2 «TinyTitan»

Beschreibung

Das Produkt wurde von der Firma «Oak Ridge Facility» entwickelt. Die Software ist unter anderem

1 Initialisierung

für RPI's entwickelt worden. TinyTitan wurde für das Durchführen wissenschaftlicher Berechnungen entworfen. Jedoch wurde seit geraumer Zeit an dem Produkt nicht mehr weitergearbeitet, wie dem offiziellen GitHub Repository zu entnehmen ist. Die Community selbst erweist sich ebenfalls als sehr klein.

Installation und Betrieb

Für die Installation des Produktes wird ein XServer vorausgesetzt, da empfohlen wird Tiny-Titan über ein GUI zu installieren. Der Installationsanleitung ist ebenfalls zu entnehmen, dass sich die Entwickler viele Gedanken über das Look a Like des Clusters gemacht haben, zum Beispiel wird ein Thema dem Einbinden von LED's gewidmet. Die Installation findet ausschliesslich durch vordefinierte Scripts statt. Durch die kleine Community und nicht mehr gepflegte Software kann nichts über den Betriebsaufwand in Erfahrung gebracht werden.

Voraussetzungen, Abhängigkeiten

Laut Guides werden lediglich 2 RPI's benötigt.

1.11.4 Variante V3 «Minimale Lösung»

Beschreibung

Die Minimale Installation ist eine zum Teil Eigenbau Lösung, welche sich nahe an diverse Guides aus dem Internet bezieht. Jedoch wird diese auf eigene Bedürfnisse angepasst.

Installation und Betrieb

Da es bei dieser Lösung selber zu entscheiden gibt, was und wie die Lösung installiert und umgesetzt werden soll, kann während der Installation darauf geachtet werden, was den grössten Vorteil für den Betrieb danach mit sich bringt. Während dem Projekt soll die Installation aber klein gehalten werden und nur das nötigste wird umgesetzt.

Voraussetzungen, Abhängigkeiten

Es werden 2 RPI's benötigt.

1.11.5 Anforderungsabdeckung der Varianten

Nr.	Kriterium	Gewichtung	Begründung
1	Installation	40%	Die Installation soll keine Hürden aufweisen, da der Zeitplan ansonsten nicht eingehalten werden kann.
2	Partner	10%	Je mehr Partner vorhanden sind, desto grösser und innovativer ist die Software. Die Software hat dadurch einen fixen Standpunkt auf dem Markt und wird weiterentwickelt.
3	Aktualität	20%	Fördert den LifeCycle und die Sicherheit der Cluster-Software.
4	Tools	30%	Mitgelieferte Tools

Tabelle 19: Anforderungsabdeckung

1 Initialisierung

Nr.	Kriterium	Note	Begründung
1	Installation	0/3/5	5 = Kann gemäss Anleitung direkt installiert werden 3 = Veraltete Anleitung, es wird mit Kompatibilitätsproblemen gerechnet 0 = Keine Anleitung vorhanden
2	Partner	0/2	2 = Viele Partner vorhanden 0 = keine Partner vorhanden
3	Aktualität	0/3/5	5 = Releases in den letzten 2 Monaten 3 = Releases in den letzten 6 Monaten 0 = Keine Releases seit einem Jahr
4	Tools	0/5	5 = Es werden Tools angeboten 0 = Es werden keine angeboten

Tabelle 20: Bewertung der Varianten

1.12 Bewertung der Varianten

Kriterium	Gewicht	Variante 1	Variante 2	Variante 3
Installation	40%	5x40 = 200	4x40 = 160	5x40 = 200
Partner	10%	2x10 = 20	0x10 = 0	0x10 = 0
Aktualität	20%	5x20 = 100	0x20 = 0	5x20 = 100
Tools	30%	5x30 = 150	0x30 = 0	0x30 = 0
Total	100%	470	160	260

Tabelle 21: Bewertung der Varianten

1.13 Variantenentscheid

Anhand der Bewertung wird empfohlen, die OpenHPC Lösung der Linux Foundation zu verwenden. Die Installation kann gemäss Anleitung in kürzester Zeit umgesetzt werden. Die Releases können mit kleinerem Aufwand installiert werden. Zudem runden die Möglichkeiten der Schnittstellen und Komponenten den Entscheid ab. Es ist möglich, Administrations- sowie Performance Monitoring Tools einzusetzen, welche mit der Lösung harmonieren. Als Hürde sehe ich die möglichen anfallenden Lizenzen und das zweite Netzwerk Interface, welches man für die Kommunikation unter den RPI's benötigt.

1.14 Risiken

In der untenstehenden Abbildung kann entnommen werden, welche Risiken während des Projekts existieren. Dabei sind Eintrittswahrscheinlichkeit und Auswirkungen tabularisch aufgelistet.

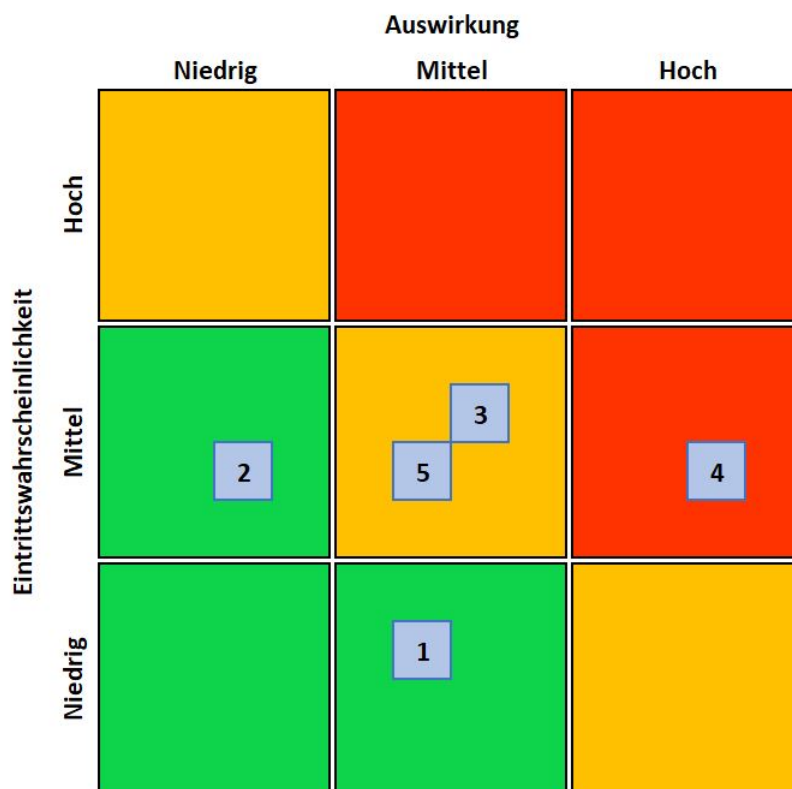


Abbildung 1: Risiken

Nr.	Beschreibung	Massnahmen zur Problemlösung
1	Der Terminplan kann nicht eingehalten werden	- Zeitplan anpassen - Experten informieren und nach einer Lösung suchen
2	Ausfall durch Unfall	Experten informieren und nach einer Lösung suchen
3	Technische Umsetzungsprobleme	- Informieren der Experten - Hilfe der Experten einholen - Alternative Lösung umsetzen
4	Defekte Hardware (Switch, Netzteil)	Hardware muss umgehend neu beschafft werden
5	Softwarefehler	Patches einspielen, Kontakt mit Lieferanten aufnehmen

Tabelle 22: Risiken

KDE! (KDE!)

2.1 Physikalischer Überblick

```
graph TD
    Internet((Internet)) --- Router[Router]
    Router --- PC[PC]
    Router --- Switch[Switch]
    NAS[NAS FileShare] --- Switch
    Switch --- Managementnode[Managementnode]
    Switch --- Computenodes[Computenodes<br/>40 aktive<br/>5 passive]
    Managementnode --- 1((1)) --- NetzteilManagement[Netzteil Managementnode]
    Computenodes --- 2((2)) --- Schaltnetzteil[Schaltnetzteil Computenodes]
```

Beschreibung

Verbindung 1

Verbindung 2

Die Computenodes werden über ein Schaltnetzteil über die GPIO Pins mit Strom betrieben.

2.2 Physikalische Verbindungen

2.2.1 Stromversorgung Managementnode

Der Managementnode wird über den Micro USB Anschluss mit Strom versorgt. Dabei muss darauf geachtet werden, dass ein mindest Strom von 2 Ampere fließt. Zudem wird eine konstante Spannung von 5 Volt benötigt. Deshalb wird ein Netzteil mit einer Leistung von 10 Watt verwendet. Das Netzteil wird über eine Stromschiene an das Stromnetz angeschlossen.

2.2.2 Computenodes

Die Managementnodes werden über die GPIO Pins via Jumperkabel über ein gemeinsames Netzteil mit Strom versorgt. Da es sich hierbei um eine Anzahl von mindestens 45 Raspberry's handelt ist ein Netzteil mit einer Leistung von 500W vorgesehen. Das Netzteil wird über die Stromschiene an das Stromnetz angeschlossen.

2.2.3 übrige Geräte

Die übrigen Geräte werden über den herkömmlichen Weg mit Strom über eine Stromschiene versorgt.

2.2.4 Netzwerkverbindungen

Die folgenden Komponenten sind über den Switch in das lokale Netzwerk eingebunden, die nicht aufgelisteten Geräte werden direkt über Powerline oder WLAN mit dem Router verbunden.

- Managementnode
- Computenodes
- NAS

2.3 Technischer Überblick

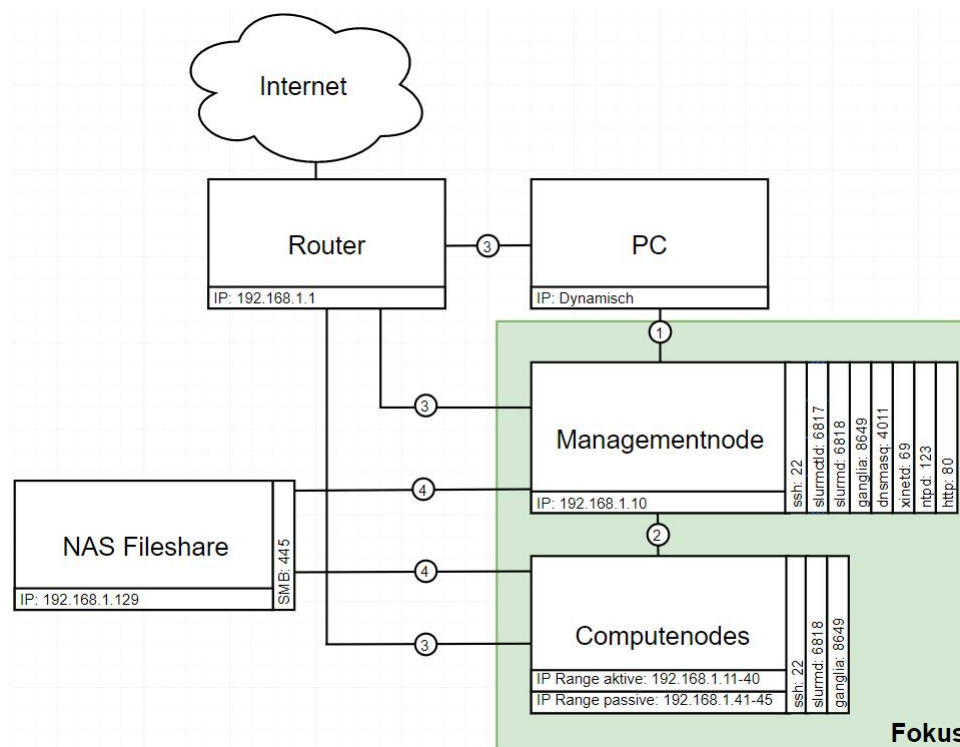


Abbildung 3: Technischer Überblick

Beschreibung

Der grün markierte Teil beinhaltet das Vorhaben. Diese Komponenten werden neu in das Netzwerk eingebunden und aufgebaut. Die Komponenten ausserhalb des grünen Bereiches existieren bereits und es müssen für die Umsetzung konfigurationen vorgenommen werden.

Verbindung 1

Der PC kann mit dem **SSH Protokoll** auf den Managementnode zugreifen. Dadurch kann die Installation vorgenommen werden. Zugleich wird über **HTTP** via Webbrowser der Zugriff auf diverse Applikationen wie z.B. Nagios & Ganglia ermöglicht.

Verbindung 2

Der Managementnode verteilt via **dnsmasq** und **TFTP** das Betriebssystem an die Computenodes über das Netzwerk. Sogleich ist auch der **Slurm Controller** für die Jobsteuerung auf dem Managementnode installiert, welcher mit den **Slurm Daemons** auf den Computenodes kommuniziert. Weiterhin sind die Monitoring Komponenten **Ganglia** und **Nagios** auf dem Managementnode installiert, welche Monitoringdaten der Computenodes sammeln und zur Auswertung verarbeiten.

Verbindung 3

Der Router verteilt via **DHCP** statische IP Adressen und Hostnamen welche über die MAC Adressen definiert sind.

Verbindung 4

Der NAS Share wird über **SMB** auf den Computenodes und dem Masternode angehängt.

2.3.1 Verwendete Protokolle

Verbindung	Protokoll	Protokollfamilie	Ports
1	SSH	TCP	22
2	SMTP	TCP	25
3	DHCP	UDP	67 / 78
4	TFTP	UDP	69
5	HTTP	TCP	80
6	SMB	TCP	445

Tabelle 23: Protokolle

2.4 Technische Verbindungen & Kommunikation

Nr.	Quelle	Ziel	Betrifft	Beschreibung
1	NAS	Mgmt	Datenablage	Der NAS Share wird über das SMB Protokoll angehängt.
2	NAS	Compute	Datenablage	Der NAS Share wird über das SMB Protokoll angehängt.
3	Router	Mgmt	IP Adresse	Anhand der MAC Adresse wird eine statische IP Adresse zugewiesen.
4	Router	Compute	IP Adressen	Anhand der MAC Adressen werden statische IP Adressen zugewiesen.
5	Router	Mgmt	Hostname	Es wird über den Router ein definierter Hostname verteilt.
6	Router	Compute	Hostnamen	Es werden über den Router definierte Hostnamen verteilt.
7	Mgmt	Compute	Netzwerkboot	Der Managementnode beliefert die Computenodes über das TFTP Protokoll mit dem Betriebssystem
8	Internet	Mgmt	Zeitserver	Die aktuelle Zeit wird mit NTP über das Internet synchronisiert.
8	Mgmt	Compute	Zeitserver	Die Computenodes beziehen die aktuelle Zeit über NTP.
9	Internet	Compute	Internetzugriff	Die Computenodes können über ein routing über den Mgmt auf das Internet zugreifen.
10	PC	Mgmt	Zugriff	Verbindungen über den PC können mit dem SSH Protokoll aufgebaut werden.
11	PC	Compute	Zugriff	Verbindungen über den PC können mit dem SSH Protokoll aufgebaut werden.

Tabelle 24: Verbindungen & Kommunikation

Legende: Mgmt = Managementnode, Compute = Computenode, PC = Home Computer

2.5 Komponentenbeschreibung

2.5.1 Router

Bei dem Router handelt es sich um eine Internet-Box Plus von Swisscom. Das Admin Interface ist über <http://internetbox> aufrufbar.

2.5.2 PC

Der PC ist selbst zusammengestellt und wird für den SSH Zugriff auf den Managementnode und für Zugriffe auf die Webanwendungen des Clusters benötigt.

2.5.3 Managementnode

Der Managementnode dient der Jobsteuerung sowie Clusterverwaltung. Alle zentralen Programme sind auf diesem Node installiert.

Hostname	nebula
Modell	Raspberry PI 3 B
Betriebssystem	Centos 7.4
CPU	Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
RAM	1GB

Tabelle 25: Komponente Managementnode

2.5.4 Netzteil Managementnode

Das Netzteil liefert eine konstante Spannung von 5V und Strom von mindestens 2 Ampere. Dabei handelt es sich um ein Noname Netzteil welches eine Mindestleistug von 10 Watt aufbringen muss.

2.5.5 NAS

Das NAS ist von der Firma Synology, das Modell lautet DS216 und wird als redundanter Datenspeicher benutzt.

2.5.6 Switch

Der Managed Switch TL-SL3428 von TP-Link wird für die Kommunikation zwischen NAS, Router, Managementnode und den Computenodes benötigt. Auf die Managed Funktion wird allerdings während des Aufbaus und Betriebs verzichtet.

2.5.7 Computenodes

Die Computenodes erhalten über das Netzwerk das Betriebssystem durch den Managementnode zugestellt. Dabei sind alle Hostnamen der Computenodes mit dem Prefix "c" versehen und werden aufnummeriert. Dabei sind die Computenodes in aktiv und passiv (Fallback, Reserve) aufgeteilt, die passiven Computenodes sollen ausgefallene aktive Computenodes ersetzen und deren Arbeiten übernehmen und die Leistung des Clusters konstant halten.

Aktiv

Hostname	c[1-40]
Modell	Raspberry PI 3 B
Betriebssystem	Centos 7.4
CPU	Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
RAM	1GB

Tabelle 26: Komponente aktive Copmputenodes

Passiv

Hostname	c[41-45]
Modell	Raspberry PI 3 B
Betriebssystem	Centos 7.4
CPU	Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
RAM	1GB

Tabelle 27: Komponente passive Copmputenodes

2.5.8 Schaltnetzteil Computenodes

Das Schaltnetzteil RSP-750-5 von Mean Well liefert konstante 5 Volt aus Ausgangsspannung und kann eine Leistung bis zu 500 Watt aufbringen, daraus können 100 Ampere auf die Nodes verteilt werden.

2.6 Tests

2.6.1 Testobjekte

Die folgende Hardware ist für die Tests der Funktionsfähigkeit des Clusters im Scope vorgesehen.

2 Konzept

Nr.	Objekt	Beschreibung
1	Managementnodes	Raspberry PI 3
2	Computenodes	Raspberry PI 3
3	NAS	Synology NAS DS216+
4	Switch	TP-Link TL-SL3428

Tabelle 28: Testobjekte

2.6.2 Testarten

Die Tests werden in folgende Kategorien eingestuft:

Nr.	Testart	Beschreibung
1	Komponententest	Die Lauffähigkeit und Erreichbarkeit der einzelnen Hardware Komponenten wird überprüft.
2	Integrationstest	Es wird die Zusammenarbeit der aktiven und neu Integrierten abhängigen Komponenten überprüft.
3	Systemtest	Das System wird als Komplettlösung getestet. Hierbei soll geprüft werden ob die Lösung den Anforderungen der Anwendbarkeit und Nutzbarkeit dem Auftrag entspricht.

Tabelle 29: Testarten

2.6.3 Testvoraussetzungen

Startbedingungen

Für den Start der Tests muss der Cluster aufgebaut sein und die einzelnen Komponenten müssen mit Strom versorgt sein.

Abbruchbedingungen

Die Tests werden abgebrochen sobald Fehler auftauchen welche Folgetests verhindern.

2.6.4 Fehlerklassen

Nr.	Fehlerklassen	Beschreibung
1	Fehlerfrei	Die Erwartungen sind erfüllt.
2	Harmloser Mangel	Es sind keine Betriebsverhinderungen zu erkennen. Die Erwartungen sind erfüllt.
3	Kleiner Mangel	Der Betrieb kann aufgenommen werden, das Problem sollte aber über einen Zeitraum von 6 Monaten behoben werden.
4	Schwerer Mangel	Der Cluster kann nur teilweise in Betrieb genommen werden, der Mangel muss innerhalb zwei Wochen behoben werden.
5	Kritischer Mangel	Der Cluster kann nicht in Betrieb genommen werden. Die Mängel müssen umgehend behoben werden.

Tabelle 30: Fehlerklassen

2.6.5 Testhilfsmittel

Die Dokumentation der Tests wird im Testprotokoll nachgeführt. Damit die Tests durchgeführt werden können wird ein PC oder Notebook welches auf mit Linux betrieben wird benötigt. Dieser Client muss sich im selben Netzwerk wie der Cluster befinden.

2.6.6 Testfälle

Bezeichnung	K-001	Start Managementnode	Systemstart
Beschreibung	Der Managementnode wird auf die Erreichbarkeit nach einem Systemstart der Komponenten überprüft		
Testvoraussetzung	Der Managementnode und der Test PC befinden sich im selben Netzwerk		
Testschritte	<ul style="list-style-type: none"> • Managementnode starten (Strom anschliessen) • 30 Sekunden warten • auf Test PC mit dem Befehl “nmap -sn 192.168.1.10 grep Nmap scan report“ auf erreichbare Geräte mit der IP 192.168.1.10 prüfen. 		
Erwartetes Ergebnis	Nmap scan report for nebula (192.168.1.10)		

Tabelle 31: Testfall K-001

2 Konzept

Bezeichnung	K-002	Start Computenodes	Systemstart
Beschreibung	Die Computenodes werden auf die Erreichbarkeit nach einem Systemstart überprüft		
Testvoraussetzung	Die Nodes und der Test PC befinden sich im selben Netzwerk, der Managementnode muss bereits in Betrieb sein		
Testschritte	<ul style="list-style-type: none"> • Computenodes an Strom anschliessen • 5 Minuten warten • auf Test PC mit dem Befehl <i>nmap -sn 192.168.1.11-55 / grep Nmap scan report</i> auf erreichbare Geräte mit der IP Range 192.168.1.11 - 192.168.1.55 prüfen 		
Erwartetes Ergebnis	Nmap scan report for c1 (192.168.1.11) ...		

Tabelle 32: Testfall K-002

Bezeichnung	K-003	Hostnamen & IP Nodes	DHCP
Beschreibung	Es wird geprüft ob alle Nodes die richtige IP Adresse und den richtigen Hostnamen zugewiesen erhalten haben		
Testvoraussetzung	Die Nodes und der Test PC befinden sich im selben Netzwerk, alle Nodes müssen gestartet sein. Dieser Test ist abhängig von K-001 und K-002		
Testschritte	<ul style="list-style-type: none"> • Auf dem Test PC mit dem Befehl <i>nmap -sn 192.168.1.10-55 / 'grep Nmap scan report/MAC Address'</i> die IP, den Hostnamen und die MAC Adressen auslesen. • Die Ausgabe mit der Hostnamenliste im Anhang vergleichen, diese muss identisch sein 		
Erwartetes Ergebnis	Nmap scan report for c1.home (192.168.1.11) MAC Address: B8:27:EB:32:39:A7 (Raspberry Pi Foundation) ... Es müssen 46 Einträge vorhanden sein		

Tabelle 33: Testfall K-003

Bezeichnung	K-004	NAS-Share	SMB
Beschreibung	Es wird getestet ob das NAS erreichbar ist und der Samba Dienst läuft		
Testvoraussetzung	Das NAS und der Managementnode sind gestartet		
Testschritte	Auf dem Managementnode mit dem Befehl <i>nc -v nasbox 445</i> in der Shell die Verbindung prüfen.		
Erwartetes Ergebnis	Ncat: Version 6.40 (http://nmap.org nccat) Ncat: Connected to 2a02:1205:5012:90f0:211:32ff:fe54:1e69:445.		

Tabelle 34: Testfall K-004

2 Konzept

Bezeichnung	I-001	Installation	Skript
Beschreibung	Das Installationsskript soll automatisiert durchlaufen. Der Cluster soll danach direkt einsetzbar sein ohne zusätzliche Konfigurationen vornehmen zu müssen		
Testvoraussetzung	Alle Komponententests müssen fehlerfrei durchgelaufen sein		
Testschritte	<ul style="list-style-type: none"> • Das Installationsskript aus dem Git Repository herunterladen • Das Skript ausführen • Installationsdurchlauf abwarten, ca. 1 Stunde • Auf den Managementnode per SSH über den PC verbinden • Den Befehl <code>sinfo</code> eingeben 		
Erwartetes Ergebnis	Die Ausgabe muss die Computenodes im idle Status ausgeben. PARTITION AVAIL TIMELIMIT NODES STATE NODELIST normal* up infinite 1 idle c[1-45]		

Tabelle 35: Testfall I-001

Bezeichnung	I-002	Netzwerkboot	Betriebssystem
Beschreibung	Der Managementnode verteilt das Betriebssystem an alle Computenodes		
Testvoraussetzung	Dnsmasq und die Netzwerkboot Verzeichnisse sind angelegt. Der Master-node ist erreichbar		
Testschritte	<ul style="list-style-type: none"> • Die Computenodes mit Strom versorgen • 5 Minuten warten • Auf dem Managementnode in der Konsole folgendes eingeben: <code>for ((i=1; i<=45; i++)); do ssh c\$i hostname; done</code> eingeben • Abwarten bis alle Computenodes durchgegangen wurden 		
Erwartetes Ergebnis	Für jeden Computenode wird der Hostname ausgegeben. c1, c2, c3, ... c45.		

Tabelle 36: Testfall I-002

2 Konzept

Bezeichnung	I-003	NAS Share	Verbindung
Beschreibung	Der NAS-Share ist auf dem Masternode und den Computenodes angehängt		
Testvoraussetzung	Es ist ein NAS-Share eingerichtet		
Testschritte	<ul style="list-style-type: none"> • Auf den Managementnode per SSH verbinden • Den Befehl mount eingeben • Den Mountpoint auf das NAS auslesen • mit cd Dir in das Verzeichnis wechseln 		
Erwartetes Ergebnis	Der Mountpoint wird bei der Eingabe von mount angezeigt Es kann in das Verzeichnis gewechselt werden Falls bereits Daten auf dem Share vorhanden sind werden diese mit dem Befehl ls -lrtha angezeigt.		

Tabelle 37: Testfall I-003

Bezeichnung	I-004	Computenodes Internet	Konnektivität
Beschreibung	Es wird überprüft ob die Computenodes eine Verbindung zum Internet aufbauen können.		
Testvoraussetzung	Managementnode und Computenodes sind gestartet		
Testschritte	Auf jedem Computenode muss in der Kommandozeile ping google.com eingeben werden		
Erwartetes Ergebnis	PING google.com (216.58.205.46) 56(84) bytes of data. 64 bytes from mil04s24-in-f46.1e100.net (216.58.205.46) 64 bytes from mil04s24-in-f46.1e100.net (216.58.205.46)		

Tabelle 38: Testfall I-004

Bezeichnung	S-001	NAS Share	Existenz
Beschreibung	Es wird überprüft ob der NAS-Share nebula vorhanden ist		
Testvoraussetzung	Das NAS ist erreichbar und der Share ist eingerichtet		
Testschritte	<ul style="list-style-type: none"> • Über den Browser auf dem PC auf http://nasbox:5000/ anmelden • Logindaten eingeben • Dem Pfad folgen Desktop/File Station/nebula 		
Erwartetes Ergebnis	Das Verzeichnis ist vorhanden und kann gelesen und beschrieben werden		

Tabelle 39: Testfall S-001

2 Konzept

Bezeichnung	S-002	Ausfall Computenode	Ausfallsicherheit
Beschreibung	Es soll getestet werden ob der Cluster weiterhin stabil läuft, nachdem ein Computenode vom Strom entfernt wird		
Testvoraussetzung	Der Cluster ist in Betrieb		
Testschritte	<ul style="list-style-type: none"> • Job in Auftrag geben (srun) • Die Stromzufuhr eines aktiven Computenodes unterbrechen • Den Job mit sinfo oder squeue abrufen 		
Erwartetes Ergebnis	Der Job läuft weiter und ist auf einen passiven Node ausgelagert worden		

Tabelle 40: Testfall S-002

Bezeichnung	S-003	Slurm	Jobverwaltung
Beschreibung	Es soll getestet werden, ob Slurm richtig konfiguriert wurde und Jobs auf dem Cluster gestartet werden können		
Testvoraussetzung	Managementnode und Computenodes sind gestartet		
Testschritte	<ul style="list-style-type: none"> • srun -nodes=40-40 -ntasks=40 -cpus-per-task=4 testslurm.sh in der Kommandozeile eingeben • squeue in der Kommandozeile eingeben 		
Erwartetes Ergebnis	Das testslurm.sh Script wird in der Queue angezeigt und der Status ist auf Running		

Tabelle 41: Testfall S-003

Bezeichnung	S-004	Nagios	Monitoring
Beschreibung	Die Installation und Konfiguration von Nagios soll überprüft werden		
Testvoraussetzung	Nagios ist installiert und Konfiguriert		
Testschritte	<ul style="list-style-type: none"> • http://nebula/nagios aufrufen • Anmeldedaten eingeben • 		
Erwartetes Ergebnis	Das testslurm.sh Script wird in der Queue angezeigt und der Status ist auf Running		

Tabelle 42: Testfall S-004

2.7 Monitoring

Als Monitoring Lösungen werden die Applikationen Nagios und Ganglia eingesetzt. Die Einsatzgebiete sind wie folgt definiert:

- Nagios = Service Monitoring
- Ganglia = Performance Monitoring

2.7.1 Service Monitoring - Nagios

Sämtliche Service Tests werden vom Managementnode aus, automatisiert in definierten Intervallen ausgeführt. Fehlgeschlagene Tests, sowie Statusänderungen der Überwachungsstatus generieren eine Benachrichtigungs E-Mail welche an den Systemadministrator versendet wird. Nagios ist über den Browser via <http://nebula/nagios> zu erreichen.

Die folgenden Überwachungen sollen während der Realisierungsphase implementiert werden. Weitere Überwachungen können nach dem Projektabschluss implementiert werden.

Nr.	Überwachung	Schweregrad	Intervall	Beschreibung
1	Erreichbarkeit	Kritisch	5	Es wird mittels Ping eine Statusüberprüfung der Nodes durchgeführt
2	SSH Zugriff	Mittel	60	Die Zugriffe auf die Computenodes sollen über den Managementnode stattfinden
3	CPU Last	Hoch	5	Die CPU's ständig ausgelastet sein

Tabelle 43: Service Monitoring

2.7.2 Performance Monitoring - Ganglia

Die Ganglia Applikation ist auf dem Managementnode installiert und kommuniziert mit den Ganglia Daemons auf den Computenodes. Dabei werden die übermittelten Daten als Grafen dargestellt. Ganglia ist über <http://nebula/ganglia> aufrufbar.

2.8 Mining

Die Kryptowährungen werden über die Miningpools von Minergate.com geschürft. Dafür wird die Cpuminer Version von [tkinjo1985](#) verwendet. Diese Version unterstützt die ARMv8 Prozessoren und bietet alle gängigen Algorithmen für das Schürfen der Währungen an. Zudem werden nur Währungen geschürft welche auf Börsen gehandelt werden können.

2.8.1 Kryptowährungen

Folgende Kryptowährungen werden über die Minergate Pools mit dem CryptoNight Algorithmus geschürft.

Nr.	Währung	Kürzel	Märkte
1	Bytecoin	BCN	HitBTC, Poloniex
2	Monero	XMR	HitBTC, Binance, Bitfinex, Poloniex
3	Monero Original	XMO	HitBTC
4	DigitalNote	XDN	HitBTC, Bittrex
5	Quazar Coin	QCN	HitBTC
6	DashCoin	DSH	HitBTC
7	FantomCoin	FCN	HitBTC

Tabelle 44: Kryptowährungen

2.9 Hostnamen

Die Computennamen wurden nach einem überschaulichen Konzept definiert. Jeder Computenode trägt den Prefix "c". Dies soll bei der Behebung von Problemen auf physischer Ebene, z.B. Austauschen eines Nodes dienen. Zudem werden alle Hostnamen immer in kleinen Buchstaben geschrieben.

2.9.1 Managementnode Name

Name	IP	MAC
nebula	192.168.1.10	B8:27:EB:32:A9:1C

Tabelle 45: Managementnode Name

2.9.2 Reservenode Name

Die Reservenodes sind als Fallback für ausgefallene Computenodes vorgesehen.

Nr.	Name	IP	MAC
1	c41	192.168.1.51	B8:27:EB:DE:C9:69
2	c42	192.168.1.52	B8:27:EB:7E:6F:48
3	c43	192.168.1.53	B8:27:EB:5D:DD:FE
4	c44	192.168.1.54	B8:27:EB:A6:6D:4D
5	c45	192.168.1.55	B8:27:EB:0C:63:10

Tabelle 46: Reservenode Name

2.9.3 Computenode Namen

Nr.	Name	IP	MAC
1	c1	192.168.1.11	B8:27:EB:32:39:A7
2	c2	192.168.1.12	B8:27:EB:2E:A3:D1
3	c3	192.168.1.13	B8:27:EB:50:45:3F
4	c4	192.168.1.14	B8:27:EB:0D:E6:25
5	c5	192.168.1.15	B8:27:EB:3E:96:B5
6	c6	192.168.1.16	B8:27:EB:EE:77:DA
7	c7	192.168.1.17	B8:27:EB:21:63:E6
8	c8	192.168.1.18	B8:27:EB:2E:2E:CC
9	c9	192.168.1.19	B8:27:EB:17:32:96
10	c10	192.168.1.20	B8:27:EB:B2:1C:A9
11	c11	192.168.1.21	B8:27:EB:AF:63:1F
12	c12	192.168.1.22	B8:27:EB:43:00:2C
13	c13	192.168.1.23	B8:27:EB:13:7B:18
14	c14	192.168.1.24	B8:27:EB:43:CD:29
15	c15	192.168.1.25	B8:27:EB:FF:C7:56
16	c16	192.168.1.26	B8:27:EB:CE:98:66
17	c17	192.168.1.27	B8:27:EB:5D:63:34
18	c18	192.168.1.28	B8:27:EB:91:3E:0F
19	c19	192.168.1.29	B8:27:EB:F4:65:EC
20	c20	192.168.1.30	B8:27:EB:3E:AB:DC
21	c21	192.168.1.31	B8:27:EB:66:60:F6
22	c22	192.168.1.32	B8:27:EB:37:3F:74
23	c23	192.168.1.33	B8:27:EB:18:5E:F0
24	c24	192.168.1.34	B8:27:EB:B0:23:B8
25	c25	192.168.1.35	B8:27:EB:BE:C4:94
26	c26	192.168.1.36	B8:27:EB:FB:FF:57
27	c27	192.168.1.37	B8:27:EB:4E:EC:CE
28	c28	192.168.1.38	B8:27:EB:43:1C:35
29	c29	192.168.1.39	B8:27:EB:DC:74:5F
30	c30	192.168.1.40	B8:27:EB:D1:DE:2F
31	c31	192.168.1.41	B8:27:EB:5E:90:34
32	c32	192.168.1.42	B8:27:EB:DE:80:24
33	c33	192.168.1.43	B8:27:EB:A4:79:6F
34	c34	192.168.1.44	B8:27:EB:0A:4D:C7
35	c35	192.168.1.45	B8:27:EB:5C:53:5F
36	c36	192.168.1.46	B8:27:EB:F7:AF:C2
37	c37	192.168.1.47	B8:27:EB:CE:BA:ED
38	c38	192.168.1.48	B8:27:EB:59:38:3C
39	c39	192.168.1.49	B8:27:EB:99:BB:8E
40	c40	192.168.1.50	B8:27:EB:8F:7A:0D

Tabelle 47: Computenode Namen

3 Realisierung

Dieses Kapitel beschreibt in welcher Reihenfolge der Cluster aufgebaut wurde. Einen tieferen Einblick in den Aufbau des Clusters kann dem Anhang entnommen werden.

3.1 Physischer Aufbau

3.1.1 Komponenten Platzierung

Der Cluster ist in einem Gestell welches 4 Ebenen hat implementiert, die Ebenen sind wie folgt aufgeteilt.

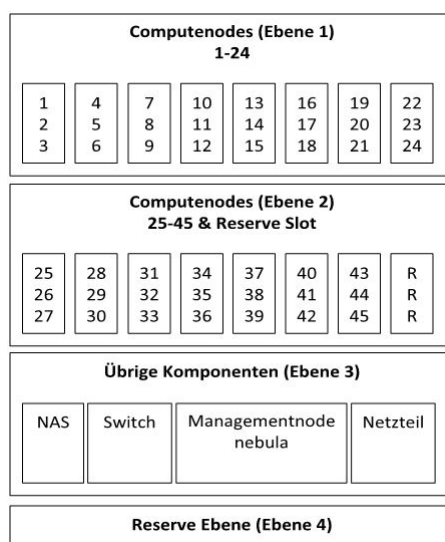


Abbildung 4: Physischer Aufbau

Ebene 1

24 RPI's sind auf dieser Ebene befestigt worden.

Ebene 2

Es befinden sich 21 RPI's auf dieser Ebene, es können noch 3 weitere platziert werden.

Ebene 3

Hier wurden alle übrigen Komponenten befestigt. Darunter ist das NAS, der Switch, der Managementnode und das Netzteil zu finden.

Ebene 4

Auf dieser Ebene wurde nichts installiert und kann als Reserve Ebene betrachtet werden.

3.1.2 Kühlung

Die CPU, RAM und GPU der RPI's wurden mit Aluminium Kühlkörpern ausgestattet. Die passive Kühlung soll die überhitzten CPU's der RPI's am laufen halten.

3.1.3 Stromversorgung

Computenodes

Die Computenodes wurden über die GPIO Pins 2 (5 Volt Anschluss) und 6 (GND Anschluss) über Jumperkabel und weiteren Leiterkabeln welche zur Verlängerung dienen mit dem Netzteil verbunden. Es wurde darauf geachtet das der Kabeldurchschnitt für eine Anzahl von mindestens 25 Ampere ausreicht, so dass diese nicht durchbrennen.

Netzteil

Am Netzteil wurden Kabelschuhe befestigt. Welche es ermöglichen eine Verbindung mit den Leitern in Richtung RPI's herzustellen. Das Netzteil ist an einer gewöhnlichen Stromschiene angeschlossen.

Generelle Verkabelung

Die Leiter wurden hauptsächlich mit Lüsterklemmen verlängert und auf das zusammenlöten der Komponenten wurde deswegen verzichtet. Dies bietet für neue Verkabelungen eine grössere Flexibilität.

3.1.4 Kommunikation

Alle Komponenten welche eine Netzwerkverbindung benötigen sind über den Switch mit Patchkabeln zusammengeschlossen worden. Es wurde dabei keine spezielle Slot zuweisung des Switches berücksichtigt.

3.2 Technischer Aufbau

3.2.1 Betriebssystem

Es existiert zur Zeit kein 64-Bit CentOS Kernel welcher mit den RPI's kompatibel ist. Deswegen wurde mit einer alternativen Lösung das Gentoo Image von <https://github.com/sakaki-/gentoo-on-rpi3-64bit> heruntergeladen und auf die SD-Karte geschrieben. Dabei wurden zwei Partitionen erstellt. Die boot Partition, welche den Kernel und die Bootbefehle beinhaltet. Die Dateisystem-Partition beinhaltet das Dateisystem und muss mit der eines CentOS Dateisystems überschrieben werden. Dabei wurde das Dateisystem von <http://mirror.centos.org/altarch/7/isos/aarch64/CentOS-7-aarch64-rootfs-7.4.1708.tar.xz> heruntergeladen und auf die betreffende Partition kopiert.

3.2.2 Vorbereitungen

Netzwerk

Die IP- und Hostnamenzuweisung wurde über die Internetbox von Swisscom eingerichtet. Dabei wurden alle RPI's an das Netzwerk und den Strom angeschlossen. Nach ca. 2 Minuten wurden alle angeschlossenen Geräte im Interface aufgelistet und konnten anhand dem Hostnamenkonzept eingerichtet werden.

Netzwerkboot

Der Managementnode dient als Provider und verteilt das Betriebssystem an alle angeschlossenen Computenodes. Diese mussten vorgängig bearbeitet werden und benötigen bei einem Power On einen OTP Eintrag, dadurch wird eine Anfrage von jedem Computenode (Client) an den Managementnode gestellt ob es das Betriebssystem erhalten darf. Zeitgleich wurden noch alle MAC-Adressen ausgelesen, diese werden später für die statische Zuweisung von IP Adressen und Hostnamen benötigt.

3.2.3 Installation

Während der Realisierung wurde nach Erfolgserlebnissen jeweils ein aktueller Snapshot der SD-Karte mit dem Programm Win32DiskImager erstellt. Dadurch war es möglich ein rasches vorantreiben der Installation zu gewährleisten. Falls zuviele Änderungen am System vorgenommen wurden, welche nicht mehr rasch rückgängig gemacht werden konnten und Probleme verursachten, wurde als Wiederherstellung eines funktionierenden Systems der letzte Snapshot wieder eingespielt.

Zudem wurde für die Installation des Clusters und deren Komponenten die Installationsanleitung (Install guide with Warewulf + Slurm) von OpenHPC verwendet. Dabei wurde der Warewulf Part zu einem grossen Teil übersprungen. Dieser hätte es ermöglicht einen vereinfachten Netzwerkboot der Computenodes einzurichten. Es ist aber leider nicht möglich die RPI's damit zu managen da Warewulf iPXE benutzt.

4 Schlussbetrachtung

Der Cluster kann parallele Berechnungen durchführen. Jedoch ist es nicht möglich Kryptowährungen mit nur einem Prozess über den Cluster hinweg zu schürfen. Jedoch ergibt es hierbei keinen unterschied ob jeweils neue Prozesse pro Node gestartet sind, oder ob diese in einen Prozess zusammengeführt sind. Dies würde erst eine Wichtigkeit erlangen, wenn es darum geht Währungen zu schürfen welche in den Blockchains diverse Bountys beinhalten, welche Belohnungen für das schürfen beinhalten.

4.1 Arbeiten nach dem Projekt

Der Cluster ist nicht rentabel und es ist vorgesehen das Anwendungsgebiet zu wechseln, dabei soll der Cluster als Entwicklerumgebung für Webanwendungen eingesetzt werden. Worauf gleichzeitig eine Virtualisierung stattfinden soll.

4.2 Persönliche Betrachtung

Generell ist es mir gelungen eine grössere Anzahl von verschiedenen Komponenten zu einem einheitlichen Produkt zu verbinden. Dadurch habe ich nun private CPU Ressourcen welche abgekoppelt von meinem PC sind. Das Produkt kann ich für meine nächsten persönlichen Vorhaben weiterhin benutzen und muss mir keine Webserver mieten.

4.3 Danksagung

An dieser Stelle möchte ich mich speziell den unten aufgeführten Personen für die Unterstützung meiner Diplomarbeit bedanken:

- Vielen Dank für die Überprüfung der Satzstellungen und das Korrigieren der Schreibfehler
- Vielen Dank für das anpassen der Schürfsoftware, welches mir ermöglicht Statistiken einzelner Nodes besser auszulesen.

5 Authentizität

Mit meiner Unterschrift bestätige ich, die vorliegende Diplomarbeit selbstständig, ohne Hilfe Dritter und nur unter Benutzung der angegebenen Quellen ohne Copyright-Verletzung, erstellt zu haben.

Schüpfen, 27.05.2018

Christoph Amrein

A Anhang

Die folgenden Kapitel gehören zum Anhang und ergänzen den Diplombericht.

B Vorbereitunge RPI's

B.1 Betriebssystem installieren

Für die Installation des CentOS 7.4 ist kein boot fähiger Kernel vorhanden. Das RPI kann aber dennoch mit einem Centos 7.4 betrieben werden. Dafür sind die folgenden Schritte vorzunehmen. Die Installation des Betriebssystems ist von einem Fedora Linux Client aus beschrieben.

1. Gentoo 64 Bit Image herunterladen (<https://github.com/sakaki-/gentoo-on-rpi3-64bit/releases/download/v1.2.1>)
2. Das Archiv wird mit dem Fedora Media Writer auf die SD Karte geschrieben. Dies ist anhand der Screenshots beschrieben.



Abbildung 5: FMW: Fedora Media Writer starten

B Vorbereitung RPI's

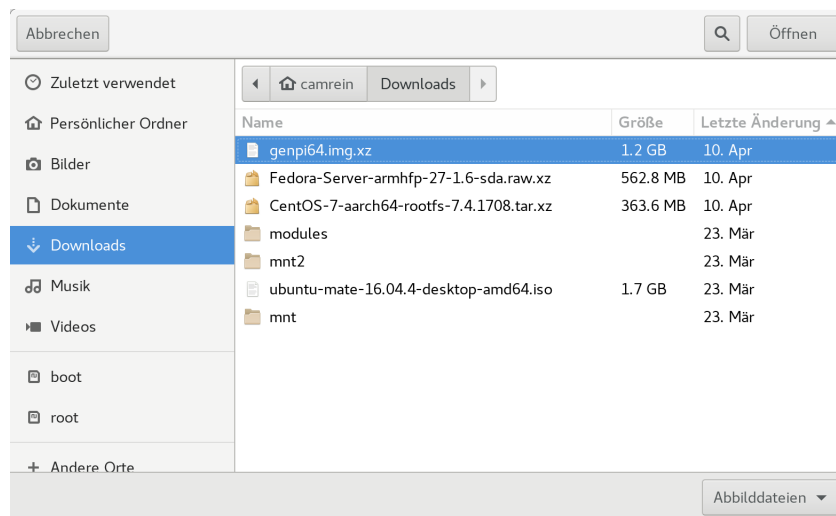


Abbildung 6: FMW: Archiv auswählen

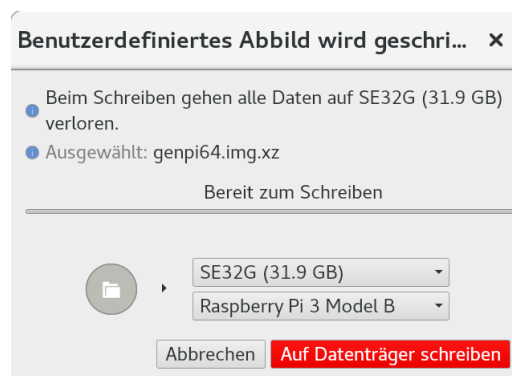


Abbildung 7: FMW: Abbild schreiben

3. Durch das Schreiben des Archivs wurden zwei Partitionen (boot & rootfs) auf der SD Karte erstellt. Diese werden wie folgt ausgelesen:

```
1 [camrein@wifibridge ~]\$ lsblk
2 NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
3 sda 8:0 0 238.5G 0 disk
4  └─sda1 8:1 0 200M 0 part /boot/efi
5  └─sda2 8:2 0 1G 0 part /boot
6  └─sda3 8:3 0 237.3G 0 part
7     └─fedora--root 253:0 0 50G 0 lvm /
8     └─fedora--swap 253:1 0 7.8G 0 lvm [SWAP]
9     └─fedora--home 253:2 0 179.5G 0 lvm /home
10 mmcblk0 179:0 0 29.7G 0 disk}
11  └─mmcblk0p1 179:1 0 43.1M 0 part /run/media/camrein/boot
12  └─mmcblk0p2 179:2 0 29.7G 0 part /run/media/camrein/rootfs
13 [camrein@wifibridge ~]\$
```

B Vorbereitung RPI's

4. Die Dateisystem Partition muss mit der von CentOS überschrieben werden. Dazu wird die Partition auf dem Linux Client angehängt. Bei Schritt 5-7 wird überprüft ob die Partition wirklich leer ist.

```
1 [root@wifibridge Downloads]# mkdir mnt
2 [root@wifibridge Downloads]# mount /dev/mmcblk0p2 mnt
3 [root@wifibridge Downloads]# cd mnt
4 [root@wifibridge mnt]# rm -rf *
5 [root@wifibridge mnt]# ls -lrtha
6 drwxr-xr-x. 5 camrein camrein 20K 15. Mai 17:39 ..
7 drwxr-xr-x. 2 root root 4.0K 16. Mai 17:58 .
```

Die Dateisystem Partition ist nun leer und kann mit der von CentOS überschrieben werden.

5. Das Dateisystem von Centos beziehen. <http://mirror.centos.org/altarch/7.4.1708/isos/aarch64/CentOS-7-aarch64-rootfs-7.4.1708.tar.xz>

6. Die geleerte Partition wird nun mit dem Dateisystem von Centos7.4 überschrieben, zugleich soll bei Schritt 2 überprüft werden, ob die Daten wirklich auf die Partition geschrieben wurden.

```
1 [root@wifibridge mnt]# tar --numeric-owner -xpJf ../CentOS-7-aarch64-rootfs-7.4.1708.tar.xz
2 [root@wifibridge mnt]# ls -lrtha
3 insgesamt 84K
4 drwxr-xr-x. 2 root root 4.0K 23. Nov 2016 srv
5 drwxr-xr-x. 2 root root 4.0K 23. Nov 2016 opt
6 drwxr-xr-x. 2 root root 4.0K 23. Nov 2016 mnt
7 drwxr-xr-x. 2 root root 4.0K 23. Nov 2016 media
8 drwxr-xr-x. 2 root root 4.0K 23. Nov 2016 home
9 drwxr-xr-x. 2 root root 4.0K 12. Sep 2017 dev
10 drwxr-xr-x. 2 root root 4.0K 12. Sep 2017 proc
11 drwxr-xr-x. 2 root root 4.0K 12. Sep 2017 run
12 drwxr-xr-x. 2 root root 4.0K 12. Sep 2017 sys
13 lrwxrwxrwx. 1 root root 7 12. Sep 2017 bin -> usr/bin
14 lrwxrwxrwx. 1 root root 8 12. Sep 2017 sbin -> usr/sbin
15 lrwxrwxrwx. 1 root root 9 12. Sep 2017 lib64 -> usr/lib64
16 lrwxrwxrwx. 1 root root 7 12. Sep 2017 lib -> usr/lib
17 drwxr-xr-x. 13 root root 4.0K 12. Sep 2017 usr
18 drwxr-xr-x. 19 root root 4.0K 12. Sep 2017 var
19 dr-xr-xr-x. 17 root root 4.0K 12. Sep 2017 .
20 drwxr-xr-x. 82 root root 4.0K 12. Sep 2017 etc
21 dr-xr-xr-x. 3 root root 4.0K 12. Sep 2017 boot
22 drwxrwxrwt. 7 root root 4.0K 12. Sep 2017 tmp
23 dr-xr-x---. 2 root root 4.0K 12. Sep 2017 root
24 drwxr-xr-x. 5 camrein camrein 20K 15. Mai 17:39 ..
25 [root@wifibridge mnt]#
```

Die SD Karte kann nun mit den RPI's verwendet werden. Diese starten jeweils mit dem Hostnamen centos.

B.2 RPI für den Netzwerkboot vorbereiten

Für das Vorbereiten der Clients für den Netzwerkboot wurde der Guide NETWORK BOOT YOU RASPBERRY PI von raspberrypi.org verwendet. (<https://www.raspberrypi.org/documentation/hardware/raspber>)
Die RPI's werden wie folgt vorbereitet:

1. Die config.txt Datei im /boot Verzeichnis benötigt einen OTP Eintrag, dieser sagt aus, dass das RPI ohne SD Karte nach einem Betriebssystem anfragen soll.

```
1 echo program_usb_boot_mode=1 | sudo tee -a /boot/config.txt
```

2. RPI neustarten
3. Prüfen ob die Änderung aktiv ist

```
1 vcgencmd otp_dump | grep 17:
```

Erwartetes Ergebnis:

```
1 17:3020000a
```

4. Den Eintrag in der /boot/config.txt wieder entfernen

MAC-Adressen auslesen

Um Zeit zu sparen können alle MAC-Adressen der RPI's während der Vorbereitung der Clients auf den Netzwerkboot ausgelesen werden.

5. nmap Scan auf die IP Range 192.168.1.0-255 von einem Linux Client aus durchführen.

```
1 nmap -sP 192.168.1.0/24
```

Erwartetes Ergebnis für ein RPI:

```
1 Nmap scan report for centos.home (192.168.1.11)
2 Host is up (0.00055s latency).
3 MAC Address: B8:27:EB:32:39:A7 (Raspberry Pi Foundation)
```

Die Vorbereitung der RPI(Computenodes) ist somit abgeschlossen und es kann mit der Installation des Managementnode fortgefahren werden.

B.3 Hostname und IP der MAC Adresse zuweisen

über <http://internetbox/> wird die IP Adresse sowie der Hostname der MAC Adresse der RPI's zugewiesen. Alle jemals angeschlossenen Geräte werden unter der Geräteliste aufgelistet. Dort können ebenfalls die Hostnamen den Geräten zugewiesen werden.

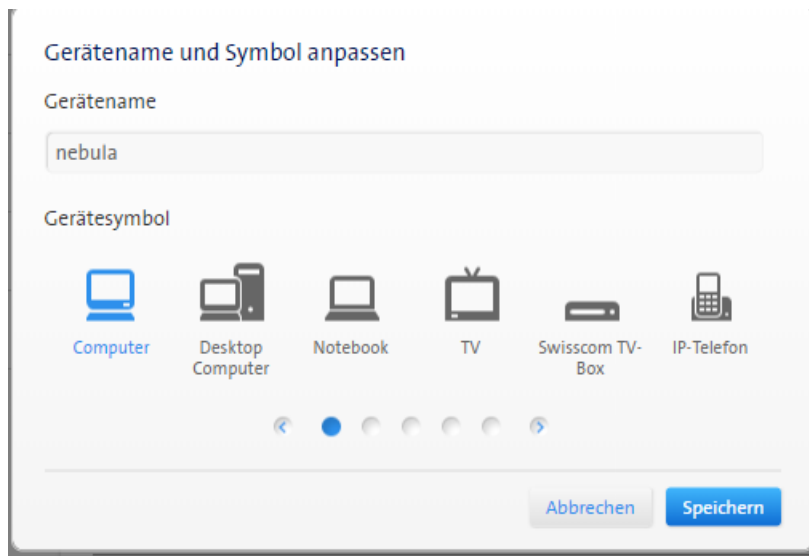


Abbildung 8: Hostnamen editieren



Abbildung 9: Übersicht der Hostnamen Zuweisung

Die fixe IP Adresse wird unter den Netzwerkeinstellungen des Routers direkt vorgenommen. Hierbei kann den bereits erkannten und definierten RPI's per Hostname eine IP zugewiesen werden.

Statische IP-Adresse automatisch zuweisen ⓘ

Gerätename	Statische IP-Adresse	MAC-Adresse	Aktionen
nebula	192.168.1.10	B8:27:EB:32:A9:1C	
c1	192.168.1.11	B8:27:EB:32:39:A7	
c2	192.168.1.12	B8:27:EB:2E:A3:D1	

Abbildung 10: Statische IP vergeben

C Installation Managementnode

Die Serverinstallation des Managementnodes wird nach dem offiziellen OpenHPC Guide mit einigen Abweichungen durchgeführt. Die Installation basiert auf der Anleitung CentOS 7.4 aarch64 Install guide with Warewulf + Slurm. Die Installation wird hauptsächlich mit dem ROOT-Benutzer durchgeführt, falls nichts anderes erwähnt wird.

C.1 Variablen Definition

Durch die Installation hinweg werden die folgenden Variablen verwendet.

Nr.	Variable	Beschreibung	Werte
1	sms_name	Hostname des Managementhosts	nebula
2	sms_ip	IP Adresse des Managementhosts	192.168.1.10
3	sms_eth_internal	Ethernet Interface	eth0
4	ntp_server[0] ntp_server[1] ntp_server[2] ntp_server[3]	Zeitserver (Array)	server 0.ch.pool.ntp.org server 1.ch.pool.ntp.org server 2.ch.pool.ntp.org server 3.ch.pool.ntp.org
5	num_computes	Anzahl Computenodes	45
6	c_ip[0] c_ip[1] c_ip[2] c_ip[3] c_ip[4] c_ip[5] c_ip[6] c_ip[7] c_ip[8] c_ip[9] c_ip[10] c_ip[11] c_ip[12] c_ip[13] c_ip[14] c_ip[15] c_ip[16] c_ip[17] c_ip[18] c_ip[19]	IP Adressen der Computenodes (Array)	192.168.1.11 192.168.1.12 192.168.1.13 192.168.1.14 192.168.1.15 192.168.1.16 192.168.1.17 192.168.1.18 192.168.1.19 192.168.1.20 192.168.1.21 192.168.1.22 192.168.1.23 192.168.1.24 192.168.1.25 192.168.1.26 192.168.1.27 192.168.1.28 192.168.1.29 192.168.1.30

Nr.	Variable	Beschreibung	Werte
6	c_ip[20]	IP Adressen der Computenodes (Array)	192.168.1.31
	c_ip[21]		192.168.1.32
	c_ip[22]		192.168.1.33
	c_ip[23]		192.168.1.34
	c_ip[24]		192.168.1.35
	c_ip[25]		192.168.1.36
	c_ip[26]		192.168.1.37
	c_ip[27]		192.168.1.38
	c_ip[28]		192.168.1.39
	c_ip[29]		192.168.1.40
	c_ip[30]		192.168.1.41
	c_ip[31]		192.168.1.42
	c_ip[32]		192.168.1.43
	c_ip[33]		192.168.1.44
	c_ip[34]		192.168.1.45
	c_ip[35]		192.168.1.46
	c_ip[36]		192.168.1.47
	c_ip[37]		192.168.1.48
	c_ip[38]		192.168.1.49
	c_ip[39]		192.168.1.50
	c_ip[40]		192.168.1.51
	c_ip[41]		192.168.1.52
	c_ip[42]		192.168.1.53
	c_ip[43]		192.168.1.54
	c_ip[44]		192.168.1.55
7	c_name[0]	Hostnamen der Computenodes (Array)	c1
	c_name[1]		c2
	c_name[2]		c3
	c_name[3]		c4
	c_name[4]		c5
	c_name[5]		c6
	c_name[6]		c7
	c_name[7]		c8
	c_name[8]		c9
	c_name[9]		c10
	c_name[10]		c11
	c_name[11]		c12
	c_name[12]		c13
	c_name[13]		c14

Nr.	Variable	Beschreibung	Werte
7	c_name[14]	Hostnamen der Computenodes (Array)	c15
	c_name[15]		c16
	c_name[16]		c17
	c_name[17]		c18
	c_name[18]		c19
	c_name[19]		c20
	c_name[20]		c21
	c_name[21]		c22
	c_name[22]		c23
	c_name[23]		c24
	c_name[24]		c25
	c_name[25]		c26
	c_name[26]		c27
	c_name[27]		c28
	c_name[28]		c29
	c_name[29]		c30
	c_name[30]		c31
	c_name[31]		c32
	c_name[32]		c33
	c_name[33]		c34
	c_name[34]		c35
	c_name[35]		c36
	c_name[36]		c37
	c_name[37]		c38
	c_name[38]		c39
	c_name[39]		c40
	c_name[40]		c41
	c_name[41]		c42
	c_name[42]		c43
	c_name[43]		c44
	c_name[44]		c45
8	c_mac[0]	MAC Adressen der Computenodes (Array)	B8:27:EB:32:39:A7
	c_mac[1]		B8:27:EB:2E:A3:D1
	c_mac[2]		B8:27:EB:50:45:3F
	c_mac[3]		B8:27:EB:0D:E6:25
	c_mac[4]		B8:27:EB:3E:96:B5
	c_mac[5]		B8:27:EB:EE:77:DA
	c_mac[6]		B8:27:EB:21:63:E6
	c_mac[7]		B8:27:EB:2E:2E:CC

Nr.	Variable	Beschreibung	Werte
8	c_mac[8]	MAC Adressen der Computenodes (Array)	B8:27:EB:17:32:96
	c_mac[9]		B8:27:EB:B2:1C:A9
	c_mac[10]		B8:27:EB:AF:63:1F
	c_mac[11]		B8:27:EB:43:00:2C
	c_mac[12]		B8:27:EB:13:7B:18
	c_mac[13]		B8:27:EB:43:CD:29
	c_mac[14]		B8:27:EB:FF:C7:56
	c_mac[15]		B8:27:EB:CE:98:66
	c_mac[16]		B8:27:EB:5D:63:34
	c_mac[17]		B8:27:EB:91:3E:0F
	c_mac[18]		B8:27:EB:F4:65:EC
	c_mac[19]		B8:27:EB:3E:AB:DC
	c_mac[20]		B8:27:EB:66:60:F6
	c_mac[21]		B8:27:EB:37:3F:74
	c_mac[22]		B8:27:EB:18:5E:F0
	c_mac[23]		B8:27:EB:B0:23:B8
	c_mac[24]		B8:27:EB:BE:C4:94
	c_mac[25]		B8:27:EB:FB:FF:57
	c_mac[26]		B8:27:EB:4E:EC:CE
	c_mac[27]		B8:27:EB:43:1C:35
	c_mac[28]		B8:27:EB:DC:74:5F
	c_mac[29]		B8:27:EB:D1:DE:2F
	c_mac[30]		B8:27:EB:5E:90:34
	c_mac[31]		B8:27:EB:DE:80:24
	c_mac[32]		B8:27:EB:A4:79:6F
	c_mac[33]		B8:27:EB:0A:4D:C7
	c_mac[34]		B8:27:EB:5C:53:5F
	c_mac[35]		B8:27:EB:F7:AF:C2
	c_mac[36]		B8:27:EB:CE:BA:ED
	c_mac[37]		B8:27:EB:59:38:3C
	c_mac[38]		B8:27:EB:99:BB:8E
	c_mac[39]		B8:27:EB:8F:7A:0D
	c_mac[40]		B8:27:EB:DE:C9:69
	c_mac[41]		B8:27:EB:7E:6F:48
	c_mac[42]		B8:27:EB:5D:DD:FE
	c_mac[43]		B8:27:EB:A6:6D:4D
	c_mac[44]		B8:27:EB:0C:63:10

Tabelle 48: Variablen Definition

C.2 Basiskonfiguration

Dem Managementnode muss ein eindeutiger Hostname zugewiesen werden. Hierbei werden die untenstehenden Kommandos abgesetzt.

```
1 echo ${sms_ip} ${sms_name} >> /etc/hosts
2 hostnamectl set-hostname ${sms_name}
```

Für die folgenden Arbeiten müssen die Firewall und SELinux deaktiviert werden. DHCP wird aufgrund der Verwendung von dnsmasq deaktiviert.

```
1 systemctl disable firewalld
2 systemctl stop firewalld
3 systemctl disable dhcpd
4 systemctl stop dhcpd
5 echo 0 > /selinux/enforce
```

C.3 OpenHPC Komponenten installieren

Es wird das OpenHPC Repository für die Installation der OpenHPC Komponenten benötigt. Dieses muss installiert werden.

```
1 yum install http://build.openhpc.community/OpenHPC:/1.3/CentOS_7/aarch64/ohpc-release-1.3-1.el7.aarch64.rpm
```

Von nun an können die benötigten Pakete über den RPM-Paketmanager installiert werden.

Durch das neu angehängte Repository können nun die folgenden Pakete installiert werden.

```
1 yum -y install ohpc-base
2 yum -y install ohpc-warewulf
```

Weiterhin spielt die Zeit eine wichtige Rolle zwischen der Kommunikation des Managementhosts und der Computenodes. Dazu müssen die bereits vorhandenen Einträge der Zeitserver in der ntp.conf entfernt werden.

```
1 systemctl disable chronyd.service
2 systemctl stop chronyd.service
3 systemctl enable ntpd.service
4 sed -i '/^server/ d' /etc/ntp.conf
5 echo "server ${ntp_server[0]}" >> /etc/ntp.conf
6 echo "server ${ntp_server[1]}" >> /etc/ntp.conf
7 echo "server ${ntp_server[2]}" >> /etc/ntp.conf
8 echo "server ${ntp_server[3]}" >> /etc/ntp.conf
9 systemctl restart ntpd
```

C Installation Managementnode

Nun wird der Slurmcontroller installiert. Dieser dient dazu die Jobs auf die Computenodes zu verteilen. Dieser steuert die Jobverwaltung und muss installiert und eingerichtet werden. Hierbei wird die Konfiguration für 45 Computenodes vorgenommen.

```
1 yum -y install ohpc-slurm-server
2 perl -pi -e "s/ControlMachine=\S+/ControlMachine=${sms_name}/" /etc/slurm/slurm.conf
3 sed -i '/^NodeName/ d' /etc/slurm/slurm.conf
4 sed -i '/^Partition/ d' /etc/slurm/slurm.conf
5 echo "NodeName=c[1-45] CoresPerSocket=1 ThreadsPerCore=1 SocketsPerBoard=4 State=UNKNOWN" >> /etc/
  slurm/slurm.conf
6 echo "PartitionName=normal Nodes=c[1-45] Default=YES MaxTime=INFINITE State=UP" >> /etc/slurm/slurm.
  conf
```

C.4 Netzwerkboot einrichten

Das Dateisystem und die Boot Partition werden in zwei verschiedenen Verzeichnissen erstellt und abgelegt.

Definieren des Dateisystem Verzeichnisses und das Centos7.4 Basis Dateisystem erstellen.

```
1 export CHROOT=/opt/ohpc/admin/images/centos7.4
2 wwmkchroot centos-7 $CHROOT
```

Anstelle von DHCPD muss der networking Service verwendet werden. Deshalb wird dieser aktiviert und gestartet.

```
1 systemctl enable networking
2 systemctl start networking
```

Nun wird das Paket dnsmasq installiert. Dadurch wird zur Absicherung die Verteilung des Betriebssystems an eine gewisse IP Range und an ein Subnetz gewährleistet. Geräte ausserhalb der Range und des Subnetzes sollen das Betriebssystem nicht erhalten dürfen.

```
1 yum -y install dnsmasq
```

Nach der Installation kann die Konfiguration von dnsmasq vorgenommen werden. Dazu werden die Datei /etc/dnsmasq.conf angepasst und die Datei /etc/dnsmasq.d/host.allow erstellt.

In der dnsmasq.conf Datei wird das Konfigurationsverzeichnis für die erlaubten Hosts erstellt, zudem wird eine IP Range angegeben, alle Geräte innerhalb dieser Range dürfen den Kernel des Betriebssystems aus dem Verzeichnis /tftboot erhalten.

```
1 [root@nebula etc]# cat /etc/dnsmasq.conf
2 conf-dir=/etc/dnsmasq.d
3 port=0
4 dhcp-range=192.168.1.10,192.168.1.50,12h
5 log-dhcp
```

C Installation Managementnode

```
6 enable-tftp
7 tftp-root=/tftpboot
8 pxe-service=0,"Raspberry Pi Boot"
```

Falls sich andere Geräte in dieser IP Range befinden wird dies durch die `/etc/dnsmasq.d/host.allow` Datei abgefangen, dort sind alle Computenodes mit MAC Adresse, Hostnamen und IP Adresse hinterlegt.

Dies kann in einer for Schleife und den definierten Variablen einfach abgefüllt werden.

```
1 for ((i=0; i<45; i++)); do echo "dhcp-host=${c_mac[$i]};${c_name[$i]};${c_ip[$i]}" >> /etc/dnsmasq.d/host.
    allow; done
```

Das Ergebnis muss wiefolgt aussehen.

```
1 [root@nebula etc]# cat /etc/dnsmasq.d/host.allow
2 dhcp-host=B8:27:EB:32:39:A7,c1,192.168.1.11
3 dhcp-host=B8:27:EB:2E:A3:D1,c2,192.168.1.12
4 dhcp-host=B8:27:EB:50:45:3F,c3,192.168.1.13
5 dhcp-host=B8:27:EB:0D:E6:25,c4,192.168.1.14
6 dhcp-host=B8:27:EB:3E:96:B5,c5,192.168.1.15
7 dhcp-host=B8:27:EB:EE:77:DA,c6,192.168.1.16
8 dhcp-host=B8:27:EB:21:63:E6,c7,192.168.1.17
9 dhcp-host=B8:27:EB:2E:2E:CC,c8,192.168.1.18
10 dhcp-host=B8:27:EB:17:32:96,c9,192.168.1.19
11 dhcp-host=B8:27:EB:B2:1C:A9,c10,192.168.1.20
12 dhcp-host=B8:27:EB:AF:63:1F,c11,192.168.1.21
13 dhcp-host=B8:27:EB:43:00:2C,c12,192.168.1.22
14 dhcp-host=B8:27:EB:13:7B:18,c13,192.168.1.23
15 dhcp-host=B8:27:EB:43:CD:29,c14,192.168.1.24
16 dhcp-host=B8:27:EB:FF:C7:56,c15,192.168.1.25
17 dhcp-host=B8:27:EB:CE:98:66,c16,192.168.1.26
18 dhcp-host=B8:27:EB:5D:63:34,c17,192.168.1.27
19 dhcp-host=B8:27:EB:91:3E:0F,c18,192.168.1.28
20 dhcp-host=B8:27:EB:F4:65:EC,c19,192.168.1.29
21 dhcp-host=B8:27:EB:3E:AB:DC,c20,192.168.1.30
22 dhcp-host=B8:27:EB:66:60:F6,c21,192.168.1.31
23 dhcp-host=B8:27:EB:37:3F:74,c22,192.168.1.32
24 dhcp-host=B8:27:EB:18:5E:F0,c23,192.168.1.33
25 dhcp-host=B8:27:EB:B0:23:B8,c24,192.168.1.34
26 dhcp-host=B8:27:EB:BE:C4:94,c25,192.168.1.35
27 dhcp-host=B8:27:EB:FB:FF:57,c26,192.168.1.36
28 dhcp-host=B8:27:EB:4E:EC:CE,c27,192.168.1.37
29 dhcp-host=B8:27:EB:43:1C:35,c28,192.168.1.38
30 dhcp-host=B8:27:EB:DC:74:5F,c29,192.168.1.39
31 dhcp-host=B8:27:EB:D1:DE:2F,c30,192.168.1.40
32 dhcp-host=B8:27:EB:5E:90:34,c31,192.168.1.41
33 dhcp-host=B8:27:EB:DE:80:24,c32,192.168.1.42
34 dhcp-host=B8:27:EB:A4:79:6F,c33,192.168.1.43
35 dhcp-host=B8:27:EB:0A:4D:C7,c34,192.168.1.44
```

C Installation Managementnode

```
36 dhcp-host=B8:27:EB:5C:53:5F,c35,192.168.1.45
37 dhcp-host=B8:27:EB:F7:AF:C2,c36,192.168.1.46
38 dhcp-host=B8:27:EB:CE:BA:ED,c37,192.168.1.47
39 dhcp-host=B8:27:EB:59:38:3C,c38,192.168.1.48
40 dhcp-host=B8:27:EB:99:BB:8E,c39,192.168.1.49
41 dhcp-host=B8:27:EB:8F:7A:0D,c40,192.168.1.50
42 dhcp-host=B8:27:EB:DE:C9:69,c41,192.168.1.51
43 dhcp-host=B8:27:EB:7E:6F:48,c42,192.168.1.52
44 dhcp-host=B8:27:EB:5D:DD:FE,c43,192.168.1.53
45 dhcp-host=B8:27:EB:A6:6D:4D,c44,192.168.1.54
46 dhcp-host=B8:27:EB:0C:63:10,c45,192.168.1.55
```

Boot Verzeichnis erstellen, aus diesem Verzeichnis wird der Kernel an die Computenodes angeboten und übermittelt. Dabei wird das /boot Verzeichnis des Managementnodes in dieses Verzeichnis kopiert, da es sich um das selbe Betriebssystem handelt.

```
1 mkdir /tftpboot
2 chmod 777 /tftpboot
3 cp -r /boot /tftpboot
```

Dabei muss darauf geachtet werden, dass die cmdline.txt Datei im /tftpboot Verzeichnis folgenden Eintrag erhält. Dieser sagt aus, dass der Computenode sein /root Verzeichnis vom Managementnode unter dem Pfad /opt/ohpc/admin/images/centos7.4 beziehen soll.

```
1 [root@nebula etc]# cat /tftpboot/cmdline.txt
2 dwc_otg.lpm_enable=0 root=/dev/nfs nfsroot=192.168.1.10:/opt/ohpc/admin/images/centos7.4,vers=3 rw ip=dhcp
   rootwait elevator=deadline
```

Den dnsmasq Service aktivieren und starten

```
1 systemctl enable dnsmasq.service
2 systemctl restart dnsmasq.service
```

OpenHPC Pakete für die Computenodes installieren.

```
1 yum -y --installroot=$CHROOT install ohpc-base compute
```

Damit die Computenodes über den Hostnamen angesprochen werden können, wird eine DNS Konfiguration benötigt. Die Auflösung findet über folgende Einstellung statt. Dabei wird vom Managementnode die resolv.conf Datei übernommen und auf das Dateisystem der Computenodes kopiert.

```
1 cp -p /etc/resolv.conf $CHROOT/etc/resolv.conf
```

Die zusätzlichen Pakete werden für die Kommunikation mit dem Slurmcontroller und der Zeitsynchronisierung benötigt.

```
1 yum -y --installroot=$CHROOT install ohpc-slurm-client
2 yum -y --installroot=$CHROOT install ntp
```

Zudem müssen gewisse Verzeichnisse vom Managementnode aus exportiert werden.

Damit die Zeitsynchronisierung über den Managementnode läuft wird folgendes umgesetzt.

Die Computenodes können nun mit Strom versorgt werden, dabei wird auf dem Managementnode auf die `/var/log/messages` Logdatei geachtet, dort sind alle Einträge des Netzwerkboots zusammengeschlossen. Dabei müssen folgende Einträge erscheinen.

```

1 May 17 20:31:34 nebula dnsmasq-dhcp[382]: 653460281 next server: 192.168.1.10
2 May 17 20:31:34 nebula dnsmasq-dhcp[382]: 653460281 sent size: 1 option: 53 message-type 2
3 May 17 20:31:34 nebula dnsmasq-dhcp[382]: 653460281 sent size: 4 option: 54 server-identifier 192.168.1.10
4 May 17 20:31:34 nebula dnsmasq-dhcp[382]: 653460281 sent size: 4 option: 51 lease-time 12h
5 May 17 20:31:34 nebula dnsmasq-dhcp[382]: 653460281 sent size: 4 option: 58 T1 6h
6 May 17 20:31:34 nebula dnsmasq-dhcp[382]: 653460281 sent size: 4 option: 59 T2 10h30m
7 May 17 20:31:34 nebula dnsmasq-dhcp[382]: 653460281 sent size: 4 option: 1 netmask 255.255.255.0
8 May 17 20:31:34 nebula dnsmasq-dhcp[382]: 653460281 sent size: 4 option: 28 broadcast 192.168.1.255
9 May 17 20:31:34 nebula dnsmasq-dhcp[382]: 653460281 sent size: 9 option: 60 vendor-class 50:58:45:43:6c:69:65:6e
   :74
10 May 17 20:31:34 nebula dnsmasq-dhcp[382]: 653460281 sent size: 17 option: 97 client-machine-id
    00:44:44:44:44:44:44:44:44:44:44:44:44:44:44:44:44:44...
11 May 17 20:31:34 nebula dnsmasq-dhcp[382]: 653460281 sent size: 32 option: 43 vendor-encap 06:01:03:0a
    :04:00:50:58:45:09:14:00:00:11...
12 May 17 20:31:34 nebula dnsmasq-dhcp[382]: 653460281 available DHCP range: 192.168.1.10 -- 192.168.1.50
13 May 17 20:31:34 nebula dnsmasq-dhcp[382]: 653460281 vendor class: PXEClient:Arch:00000:UNDI:002001
14 ..
15 ..
16 ..
17 May 17 20:31:35 nebula dnsmasq-tftp[382]: sent /tftpboot/bootcode.bin to 192.168.1.46
18 May 17 20:31:35 nebula dnsmasq-tftp[382]: file /tftpboot/autoboot.txt not found
19 May 17 20:31:35 nebula dnsmasq-tftp[382]: sent /tftpboot/config.txt to 192.168.1.38
20 May 17 20:31:35 nebula dnsmasq-tftp[382]: file /tftpboot/recovery.elf not found
21 May 17 20:31:35 nebula dnsmasq-tftp[382]: file /tftpboot/beb21ca9/start.elf not found
22 May 17 20:31:35 nebula dnsmasq-tftp[382]: file /tftpboot/75a66d4d/start.elf not found
23 May 17 20:31:35 nebula dnsmasq-tftp[382]: file /tftpboot/d0185ef0/start.elf not found

```

C Installation Managementnode

```
24 May 17 20:31:35 nebula dnsmasq-tftp[382]: file /tftpboot/10dec969/start.elf not found
25 May 17 20:31:35 nebula dnsmasq-tftp[382]: file /tftpboot/d4373f74/start.elf not found
26 May 17 20:31:35 nebula dnsmasq-tftp[382]: file /tftpboot/0643cd29/start.elf not found
27 May 17 20:31:35 nebula dnsmasq-tftp[382]: file /tftpboot/bootsig.bin not found
28 May 17 20:31:35 nebula dnsmasq-tftp[382]: sent /tftpboot/bootcode.bin to 192.168.1.40
29 May 17 20:31:35 nebula dnsmasq-tftp[382]: file /tftpboot/bootsig.bin not found
30 May 17 20:31:35 nebula dnsmasq-tftp[382]: sent /tftpboot/bootcode.bin to 192.168.1.16
31 May 17 20:31:35 nebula dnsmasq-tftp[382]: file /tftpboot/bootsig.bin not found
32 May 17 20:31:35 nebula dnsmasq-tftp[382]: sent /tftpboot/bootcode.bin to 192.168.1.27
33 ..
34 ..
35 ..
36 May 17 20:32:56 nebula dnsmasq-dhcp[382]: 925035958 next server: 192.168.1.10
37 May 17 20:32:56 nebula dnsmasq-dhcp[382]: 925035958 sent size: 1 option: 53 message-type 2
38 May 17 20:32:56 nebula dnsmasq-dhcp[382]: 925035958 sent size: 4 option: 54 server-identifier 192.168.1.10
39 May 17 20:32:56 nebula dnsmasq-dhcp[382]: 925035958 sent size: 4 option: 51 lease-time 12h
40 May 17 20:32:56 nebula dnsmasq-dhcp[382]: 925035958 sent size: 4 option: 58 T1 6h
41 May 17 20:32:56 nebula dnsmasq-dhcp[382]: 925035958 sent size: 4 option: 59 T2 10h30m
42 May 17 20:32:56 nebula dnsmasq-dhcp[382]: 925035958 sent size: 4 option: 1 netmask 255.255.255.0
43 May 17 20:32:56 nebula dnsmasq-dhcp[382]: 925035958 sent size: 4 option: 28 broadcast 192.168.1.255
44 May 17 20:32:56 nebula dnsmasq-dhcp[382]: 925035958 sent size: 4 option: 3 router 192.168.1.10
45 May 17 20:32:56 nebula dnsmasq-dhcp[382]: 925035958 sent size: 3 option: 12 hostname c12
46 May 17 20:32:56 nebula dnsmasq-dhcp[382]: 925035958 available DHCP range: 192.168.1.10 -- 192.168.1.50
47 May 17 20:32:56 nebula rpc.mountd[495]: authenticated mount request from 192.168.1.22:999 for /opt/ohpc/admin/
    images/centos7.4 (/opt/ohpc/admin/images/centos7.4)
```

Durch die letzte Zeile (authenticated mount request from 192.168.1.22) ist ersichtlich, dass der Computenode c12 gestartet wurde und nach dem Betriebssystem anfragt und dieses mounten will.

C.5 Registration Computenodes

Die Installation ist soweit abgeschlossen, die optionalen Pakete werden später installiert und beschrieben. Nun werden die Computenodes registriert, ab diesem Zeitpunkt kennt der Managementnode jeden Computenode. Die Computenodes werden wie folgt in den Datastore aufgenommen.

```
1 echo "GATEWAYDEV=${eth_internal}" > /tmp/network.$$
2 wwsh -y file import /tmp/network.$$ --name network
3 wwsh -y file set network --path /etc/sysconfig/network --mode=0644 --uid=0
4 for ((i=0; i<$num_computes; i++)) ; do
5 wwsh -y node new ${c_name[i]} --ipaddr=${c_ip[i]} --hwaddr=${c_mac[i]} -D ${eth_internal}
6 done
```

C.6 Monitoring installieren

Das Nagios Monitoring wird wie folgt installiert. Dabei werden alle Grundüberwachungen installiert und können Out of the Box verwendet werden.

```
1 [root@nebula /]# yum -y install ohpc-nagios
2 [root@nebula /]# yum -y --installroot=$CHROOT install nagios-plugins-all-ohpc nrpe-ohpc
3 [root@nebula /]# chroot $CHROOT systemctl enable nrpe
4 [root@nebula /]# perl -pi -e "s/^allowed_hosts=/# allowed_hosts=/" $CHROOT/etc/nagios/nrpe.cfg
5 [root@nebula /]# echo "nrpe 5666/tcp # NRPE" >> $CHROOT/etc/services
6 [root@nebula /]# echo "nrpe : ${sms_ip} : ALLOW" >> $CHROOT/etc/hosts.allow
7 [root@nebula /]# echo "nrpe : ALL : DENY" >> $CHROOT/etc/hosts.allow
8 [root@nebula /]# chroot $CHROOT /usr/sbin/useradd -c "NRPE user for the NRPE service" -d /var/run/nrpe -r
   -g nrpe -s /sbin/nologin nrpe
9 [root@nebula /]# chroot $CHROOT /usr/sbin/groupadd -r nrpe
10 [root@nebula /]# mv /etc/nagios/conf.d/services.cfg.example /etc/nagios/conf.d/services.cfg
11 [root@nebula /]# mv /etc/nagios/conf.d/hosts.cfg.example /etc/nagios/conf.d/hosts.cfg
12 [root@nebula /]# for ((i=0; i<$num_computes; i++)) ; do
13 perl -pi -e "s/HOSTNAME$((i+1))/${c_name[i]}/ || s/HOST$((i+1))_IP/${c_ip[i]}/" \
14 [root@nebula /]# /etc/nagios/conf.d/hosts.cfg
15 done
16 [root@nebula /]# perl -pi -e "s/ \\\bin\\mail/ \\\usr\\bin\\mailx/g" /etc/nagios/objects/commands.cfg
17 [root@nebula /]# perl -pi -e "s/nagios\\@localhost/root\\@${sms_name}/" /etc/nagios/objects/contacts.cfg
18 [root@nebula /]# echo command[check_ssh]=/usr/lib64/nagios/plugins/check_ssh localhost >> $CHROOT/etc/
   nagios/nrpe.cfg
19 [root@nebula /]# htpasswd -bc /etc/nagios/passwd nagiosadmin ${Eigenes_Passwort}
20 [root@nebula /]# chkconfig nagios on
21 [root@nebula /]# systemctl start nagios
22 [root@nebula /]# chmod u+s 'which ping'
```

Nagios kann nun eingesetzt werden und ist über <http://nebula/nagios> erreichbar.

Das Performance Monitoring wird mit Ganglia realisiert welches sich wie folgt installieren lässt.

```
1 [root@nebula /]# yum -y install ohpc-ganglia
2 [root@nebula /]# yum -y --installroot=$CHROOT install ganglia-gmond-ohpc
3 [root@nebula /]# cp /opt/ohpc/pub/examples/ganglia/gmond.conf /etc/ganglia/gmond.conf
4 [root@nebula /]# perl -pi -e "s/<sms>/${sms_name}/" /etc/ganglia/gmond.conf
5 [root@nebula /]# cp /etc/ganglia/gmond.conf $CHROOT/etc/ganglia/gmond.conf
6 echo "gridname MySite" >> /etc/ganglia/gmetad.conf
7 [root@nebula /]# systemctl enable gmond
8 [root@nebula /]# systemctl enable gmetad
9 [root@nebula /]# systemctl start gmond
10 [root@nebula /]# systemctl start gmetad
11 [root@nebula /]# chroot $CHROOT systemctl enable gmond
12 [root@nebula /]# systemctl try-restart httpd
```

Ganglia kann nun eingesetzt werden und ist über <http://nebula/ganglia> erreichbar.

C.7 Miner installieren

Es wird die cpuminer-multi Version von tkinjo verwendet. Diese ist mit dem ARMv8 Prozessor kompatibel und der Miner kann kompiliert werden.

Die folgenden Schritte sind für die Installation notwendig.

Verzeichnis erstellen in dem der Miner installiert werden soll.

```
1 [root@nebula /]# mkdir -p /opt/miners
```

Das tkinjo cpuminer-multi Repository klonen. Dabei wird das /opt/miners/cpuminer-multi Verzeichnis erstellt.

```
1 [root@nebula /]# cd /opt/miners
2 [root@nebula miners]# git clone https://github.com/tkinjo1985/cpuminer-multi.git
3 Klonen nach 'cpuminer-multi'...
4 remote: Counting objects: 3805, done.
5 remote: Total 3805 (delta 0), reused 0 (delta 0), pack-reused 3805
6 Empfangen Objekte: 100% (3805/3805), 18.98 MiB | 3.77 MiB/s, done.
7 Löse Unterschiede auf: 100% (2589/2589), done.
```

Die geklonte Version kann nun kompiliert werden. Dabei kann die Warnung: Implizite Deklaration der Funktion, ignoriert werden. Zur Übersicht wurde die Ausgabe unten gekürzt.

```
1 [root@nebula miners]# cd cpuminer-multi/
2 [root@nebula cpuminer-multi]# ./build.sh
3 make: *** Keine Regel, um >>clean<< zu erstellen. Schluss.
4 clean
5 configure.ac:15: installing './compile'
6 configure.ac:4: installing './config.guess'
7 configure.ac:4: installing './config.sub'
8 configure.ac:9: installing './install-sh'
9 configure.ac:9: installing './missing'
10 Makefile.am: installing './depcomp'
11 checking build system type... aarch64-unknown-linux-gnu
12 checking host system type... aarch64-unknown-linux-gnu
13 checking target system type... aarch64-unknown-linux-gnu
14 checking for a BSD-compatible install... /usr/bin/install -c
15 checking whether build environment is sane... yes
16 checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
17 In file included from algo/./scryptjane/scrypt-jane-romix.h:2:0,
18      from algo/scrypt-jane.c:24:
19 algo/./scryptjane/scrypt-jane-chacha.h: In Funktion >>scrypt_test_mix<<:
20 algo/./scryptjane/scrypt-jane-chacha.h:125:20: Warnung: Implizite Deklaration der Funktion >>detect_cpu<< [-Wimplicit-function-declaration]
21     size_t cpuflags = detect_cpu();
22                        ^~~~~~
23 make[2]: Leaving directory '/opt/miners/cpuminer-multi'
24 make[1]: Leaving directory '/opt/miners/cpuminer-multi'
```


C.8 Quellenverzeichnis

D Diplomeingabe

Projekt: Mining Cluster

Christoph Amrein TSBE 16B

Praktische Diplomarbeit 2018

Ausgangslage

Gemäss der Webseite coinmarketcap.com, einer Webseite zur Verfolgung der Kurse von Kryptowährungen, gibt es zurzeit 1349¹ Kryptowährungen und täglich werden es mehr. Aus diesem Grund habe ich anfangs Jahr selbst mit dem Erzeugen von Coins diverser Kryptowährungen begonnen. Jedoch bin ich mit meiner aktuellen Ausrüstung, einem leistungsstarken Computer mit einer guten Grafikkarte, nicht zufrieden. Die Stromkosten sind zu hoch und die Erträge zu gering. Aus diesem Grund will ich ein Projekt durchführen, welches meine Erträge auf ein neues Niveau heben soll. Dazu soll auf einem Cluster, der aus Raspberry PI's besteht, ein Miner² installiert werden, welcher alle verfügbaren Ressourcen in diesem Cluster nutzt. Ich habe vor, dieses Projekt im Rahmen meiner Diplomarbeit durchzuführen. Das Projekt soll durch den Handel mit Kryptowährungen finanziert werden.

Begründung

Es soll ein skalierbarer Cluster aus ca. 40 Raspberry PI's installiert werden, auf dem eine Mining-Software betrieben wird. Das Einsatzgebiet des Clusters soll jederzeit ohne viel Aufwand angepasst werden können. Nach der Umsetzung sollen durch den Cluster möglichst effizient Coins erzeugt werden. Dabei soll der Ertrag aus den erzeugten Coins den Stromkosten gegenübergestellt werden. Ein weiteres Ziel der Projektarbeit ist es, mit diesen auf den üblichen Handelsplattformen tätig zu sein. Aus Kostengründen habe ich als optionales Ziel vorgesehen, Grafikkarten in den Cluster einzubinden und mit diesen ebenfalls möglichst effizient Coins zu erzeugen. Der Fokus der Arbeit liegt hierbei aber hauptsächlich auf dem Cluster und dessen CPU Nutzung. Die Kosten für das Projekt sollen innerhalb von 2 Jahren amortisiert werden. Danach will ich damit über längere Zeit Geld verdienen.

Themenbereiche

Es wird ein fundamentales Wissen in **Elektrotechnik** benötigt, da die Raspberry PI's durch eine gemeinsame Stromquelle versorgt werden sollen. Damit die Kommunikation zwischen Raspberry PI's eingerichtet werden kann, wird Wissen in **Linux** und **Netzwerktechnik** benötigt. Nach der Installation des Clusters sind Kenntnisse in **Entwicklung**, **Programmieren**, **Monitoring** und **Systempflege** von Nöten.

¹ Stand 13.12.2017

² Software zum Minen der Kryptowährung

Projektziele

Operationelles Ziel

Der Cluster soll durchgehend und selbstständig funktionieren. Die Arbeiten auf dem Cluster sollen sich auf das Patching und Updaten des Betriebssystems sowie der Miner- und Cluster-Software beziehen.
Bei Problemen ausserhalb der oben genannten Aufgaben soll automatisch durch die Systemüberwachung Alarm ausgelöst und eine Nachricht versendet werden.

Abwicklungsziele

- Zeitplan einhalten
 - o Es sollen keine grossen Abweichungen zum Zeitplan entstehen.
 - o Die Meilensteine müssen eingehalten werden.
- Arbeitsjournal führen
 - o Es wird ein lückenloses und verständliches Arbeitsjournal geführt.
 - o Das Arbeitsjournal soll zeitnahe geschrieben und ergänzt werden.

Wirkungs- und Nutzenziele

- Es soll das Maximale an Ressourcen aus den Raspberry PI's herausgeholt werden.
- Der Cluster kann schnell für andere Anwendungsgebiete konfiguriert werden.
- Durch die Lösung sollen verschiedene Coins diverser Kryptowährungen erzeugt werden.
- Es soll mit den erzeugten Coins auf Handelsplattformen gehandelt werden.

Lieferobjekte

Initialisierung

- Detaillierter Projektplan
- Projektauftrag
- Dokumente zum Kick-Off-Meeting

Voranalyse

- Diplombericht
- Initiale Voranalyse
- Präsentation

Konzept

- Hostnamenkonzept der einzelnen Raspberry PI's
- Backupkonzept
- Miningkonzept
- Überwachungskonzept

Realisierung

- Dokumentation der Arbeit
- Abnahmetests, Testprotokoll
- Cluster aus Raspberry PI's
- Transaktionsauszug der erzeugten Währung (Wallet zu Wallet)

Projektabschluss

- Management Summary
- Abschlussbericht
- Präsentation der Arbeit
- Arbeitsjournal

Projektplan

Monat	Januar				Februar				März					April				Mai				Juni				
Kalenderwoche	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Phasen																										
Initialisierung																										
Voranalyse																										
Konzept																										
Realisierung									Ms				Ms						Ms							
Abschluss																										
Meetings			Me									Me												Me		
Dokumentation																										

Organisation

Infrastruktur

Das Projekt wird bei mir zu Hause durchgeführt.

Beteiligte Personen

Funktion	Name
Auftraggeber	Christoph Amrein
Projektleiter	Christoph Amrein
Ausführender	Christoph Amrein
Begleitender Dozent	Andreas Megert, TSBE
Begleitender Experte	Rolf Schmutz, Post CH AG

E Testprotokoll

E.1 Testobjekte

Nr.	Objekt	Hostname	IP	MAC
1	Managementnode	nebula	192.168.1.11	B8:27:EB:32:39:A7
2	Computenode 1	c1	192.168.1.11	B8:27:EB:32:39:A7
3	Computenode 2	c2	192.168.1.12	B8:27:EB:2E:A3:D1
4	Computenode 3	c3	192.168.1.13	B8:27:EB:50:45:3F
5	Computenode 4	c4	192.168.1.14	B8:27:EB:0D:E6:25
6	Computenode 5	c5	192.168.1.15	B8:27:EB:3E:96:B5
7	Computenode 6	c6	192.168.1.16	B8:27:EB:EE:77:DA
8	Computenode 7	c7	192.168.1.17	B8:27:EB:21:63:E6
9	Computenode 8	c8	192.168.1.18	B8:27:EB:2E:2E:CC
10	Computenode 9	c9	192.168.1.19	B8:27:EB:17:32:96
11	Computenode 10	c10	192.168.1.20	B8:27:EB:B2:1C:A9
12	Computenode 11	c11	192.168.1.21	B8:27:EB:AF:63:1F
13	Computenode 12	c12	192.168.1.22	B8:27:EB:43:00:2C
14	Computenode 13	c13	192.168.1.23	B8:27:EB:13:7B:18
15	Computenode 14	c14	192.168.1.24	B8:27:EB:43:CD:29
16	Computenode 15	c15	192.168.1.25	B8:27:EB:FF:C7:56
17	Computenode 16	c16	192.168.1.26	B8:27:EB:CE:98:66
18	Computenode 17	c17	192.168.1.27	B8:27:EB:5D:63:34
19	Computenode 18	c18	192.168.1.28	B8:27:EB:91:3E:0F
20	Computenode 19	c19	192.168.1.29	B8:27:EB:F4:65:EC
21	Computenode 20	c20	192.168.1.30	B8:27:EB:3E:AB:DC
22	Computenode 21	c21	192.168.1.31	B8:27:EB:66:60:F6
23	Computenode 22	c22	192.168.1.32	B8:27:EB:37:3F:74
24	Computenode 23	c23	192.168.1.33	B8:27:EB:18:5E:F0
25	Computenode 24	c24	192.168.1.34	B8:27:EB:B0:23:B8
26	Computenode 25	c25	192.168.1.35	B8:27:EB:BE:C4:94
27	Computenode 26	c26	192.168.1.36	B8:27:EB:FB:FF:57
28	Computenode 27	c27	192.168.1.37	B8:27:EB:4E:EC:CE
29	Computenode 28	c28	192.168.1.38	B8:27:EB:43:1C:35
30	Computenode 29	c29	192.168.1.39	B8:27:EB:DC:74:5F
31	Computenode 30	c30	192.168.1.40	B8:27:EB:D1:DE:2F
32	Computenode 31	c31	192.168.1.41	B8:27:EB:5E:90:34
33	Computenode 32	c32	192.168.1.42	B8:27:EB:DE:80:24
34	Computenode 33	c33	192.168.1.43	B8:27:EB:A4:79:6F
35	Computenode 34	c34	192.168.1.44	B8:27:EB:0A:4D:C7
36	Computenode 35	c35	192.168.1.45	B8:27:EB:5C:53:5F
37	Computenode 36	c36	192.168.1.46	B8:27:EB:F7:AF:C2
38	Computenode 37	c37	192.168.1.47	B8:27:EB:CE:BA:ED
39	Computenode 38	c38	192.168.1.48	B8:27:EB:59:38:3C
40	Computenode 39	c39	192.168.1.49	B8:27:EB:99:BB:8E
41	Computenode 40	c40	192.168.1.50	B8:27:EB:8F:7A:0D
42	Reservenode 1	c41	192.168.1.51	B8:27:EB:DE:C9:69
43	Reservenode 2	c42	192.168.1.52	B8:27:EB:7E:6F:48
44	Reservenode 3	c43	192.168.1.53	B8:27:EB:5D:DD:FE
45	Reservenode 4	c44	192.168.1.54	B8:27:EB:A6:6D:4D
46	Reservenode 5	c45	192.168.1.55	B8:27:EB:0C:63:10

Tabelle 49: Computenode Namen

E.2 Arbeitsjournal

- Tagesziel eintragen - Ereignisse - Erfahrungen, Gedanken, Ideen, Entscheidungen - Ziel erreicht?

Vorgang	Geplant	Tatsächlich	Differenz
1. Abnahmetest der Fachabteilung	1 h	1 h	






E.3 Protkolle

E.4 Datenblätter

E.5 Produktinformationen

E.6 Benutzerdokumentation

Ausschnitt aus der Benutzerdokumentation:

Symbol	Bedeutung global	Bedeutung einzeln
	Alle Module weisen den gleichen Stand auf.	Das Modul ist auf dem gleichen Stand wie das Modul auf der vorherigen Umgebung.
	Es existieren keine Module (fachlich nicht möglich).	Weder auf der aktuellen noch auf der vorherigen Umgebung sind Module angelegt. Es kann also auch nichts übertragen werden.
	Ein Modul muss durch das Übertragen von der vorherigen Umgebung erstellt werden.	Das Modul der vorherigen Umgebung kann übertragen werden, auf dieser Umgebung ist noch kein Modul vorhanden.
	Auf einer vorherigen Umgebung gibt es ein Modul, welches übertragen werden kann, um das nächste zu aktualisieren.	Das Modul der vorherigen Umgebung kann übertragen werden um dieses zu aktualisieren.
	Ein Modul auf einer Umgebung wurde entgegen des Entwicklungsprozesses gespeichert.	Das aktuelle Modul ist neuer als das Modul auf der vorherigen Umgebung oder die vorherige Umgebung wurde übersprungen.

Abkürzungsverzeichnis

Abbildungsverzeichnis

1	Risiken	15
2	Physikalischer Überblick	16
3	Technischer Überblick	18
4	Physischer Aufbau	31
5	FMW: Fedora Media Writer starten	35
6	FMW: Archiv auswählen	36
7	FMW: Abbild schreiben	36
8	Hostnamen editieren	39
9	Übersicht der Hostnamen Zuweisung	39
10	Statische IP vergeben	39

Tabellenverzeichnis

1	Situationsanalyse Komponenten	1
2	Situationsanalyse Stärken	2
3	Situationsanalyse Stärken	2
4	Projektziele	3
5	Lieferobjekte	4
6	Beschaffungskosten	5
7	Aufwandskosten	5
8	Stromkostenrechnung	6
9	Gesamtkosten	6
10	Wirtschaftlichkeit Hardware	7
11	Grober Projektplan	8
12	Termine	9
13	Projektbudget	9
14	Sachmittel	9
15	Organisation	10
16	Projektablage	10
17	Software Kriterien	11
18	Variantenübersicht	12
19	Anforderungsabdeckung	13
20	Bewertung der Varianten	14
21	Bewertung der Varianten	14
22	Risiken	15
23	Protokolle	19
24	Verbindungen & Kommunikation	19
25	Komponente Managementnode	20
26	Komponente aktive Copmputenodes	21
27	Komponente passive Copmputenodes	21
28	Testobjekte	22
29	Testarten	22
30	Fehlerklassen	23
31	Testfall K-001	23
32	Testfall K-002	24
33	Testfall K-003	24
34	Testfall K-004	24
35	Testfall I-001	25
36	Testfall I-002	25
37	Testfall I-003	26
38	Testfall I-004	26
39	Testfall S-001	26

Tabellenverzeichnis

40	Testfall S-002	27
41	Testfall S-003	27
42	Testfall S-004	27
43	Service Monitoring	28
44	Kryptowährungen	29
45	Managementnode Name	29
46	Reservenode Name	29
47	Computenode Namen	30
48	Variablen Definition	43
49	Computenode Namen	58

Listings