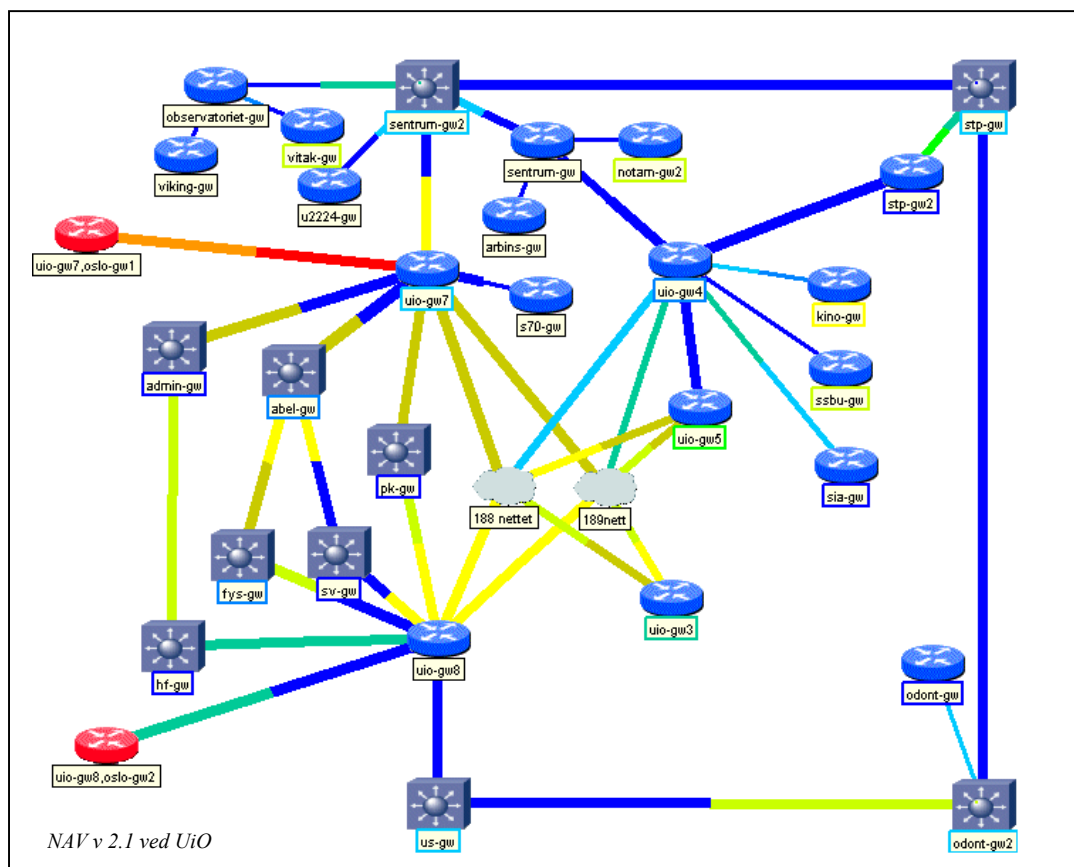


Sluttrapport NAVRun



Installasjonspakke for NAV

UNINETT Prosjekt 353101.12 kontrakt K-04-041

Prosjektleder
Vidar Faltinsen

Prosjektgruppe
Morten Vold
John Magne Bredal – Kristian Eide
Sigurd Gartmann – Knut-Helge Vindheim

ITEA, NTNU
13. november 2002

Sammendrag

Prosjekt NAVRun har til hensikt å gjøre NAV¹ lett tilgjengelig for interesserte høyskoler og universitet. Prosjektet er utført av NTNUs IT-seksjon ITEA på oppdrag fra UNINETT. Arbeidet har pågått fra 6. mai til 13. november 2002. Prosjektgruppen har bestått av 6 personer, totalt timevolum har vært 1118 timer.

NAVRun har fortsatt der prosjekt NAVMe slapp. I NAVMe løste vi NAV fra NTNU og gjennomførte en vellykket pilot ved UiTø. Fokus var *ikke* på installasjon, prosjektgruppen satt NAV manuelt på plass ved UiTø.

NAVRun har fokusert på distribusjon, hovedaktiviteten har vært å lage en installasjons-pakke. Arbeidet har krevd ytterligere rydding i struktur og system (katalogstruktur, databasekall, cronbruker m.m). Vi har i detalj sett på de ytre krav som stilles til NAV.

Installasjonsscriptet er laget generelt med tanke på utvidelser. Vi har selv implementert støtte for Mandrake og RedHat Linux. UiTø har allerede utvidet med FreeBSD-støtte.

Installasjonsscriptet gjør mer enn å installere NAV. Det sjekker først om nødvendige pre-rekvisitter er installert. Hvis ikke vil den selv installere ved å bruke et pakkesystem (apt-rpm for RedHat, urpmi for Mandrake)². Installasjonen håndterer også nødvendig konfigurasjon av prerekvisitter, så som PostgreSQL, Apache, Tomcat og syslogd.

Utover selve installasjonen har NAVRun laget hjelpescript for å starte- evt delstarte NAV, samt monitorere at alle delsystemer kjører. Det er også laget script for å lettere fylle data i databasen.

Vi har informert alle høyskoler og universitet om vårt arbeid og bedt interesserte melde seg. 19. oktober annonserte vi NAV v 2.1.0 for tilgjengelig. Alle fire universitet og 12 høyskoler viste aktiv interesse og deltok på en heldags workshop som vi arrangerte 5. november (totalt 37 deltagere).

Vi jobber kontinuerlig med å utbedre dokumentasjon. Et verktøy for bugrapportering er på plass og vi har etablert et driftsforum for NAV gjennom epostlisten nav-users@itea.ntnu.no. Listen er godt mottatt og har allerede stor aktivitet. Her vil vi gjøre oppfølging også etter at prosjekt NAVRun nå formelt avsluttes.

Vi benytter CVS mer gjennomført med politikk for hvordan vi skal grene av og sette merkelapper/versjonsnummer. Sist men ikke minst har vi utviklet et system for å håndtere oppgradering/bugfixing. Første patch ble sluppet rett før prosjektavslutning. I skrivende stund er det NAV versjon 2.1.1 som gjelder.

Status ved prosjektslutt er at 10 NAV-installasjoner er oppe og kjører, mer eller mindre fullverdig³. Flere installasjoner er like rundt hjørnet. Vi konkluderer med at prosjekt NAVRun har oppnådd sin målsetning.

¹ NAV står for Network Administration Visualized og er NTNUs egenutviklede nettadministrasjonsløsning. Utviklingen startet i 1999 og har pågått i snart 4 år. UNINETT har fra 2001 støttet NAV gjennom tre prosjekter; NAVMe, NAVRun og NAVMore.

² FreeBSD-installasjonen krever at prerekvisitter installeres manuelt på forhånd.

³ Den største installasjonen, UiO, pryder forsiden av denne rapporten.

Innhold

FORORD	6
KAPITTEL 1: INNLEDNING	7
1.1 HVA INITIERTE DETTE PROSJEKTET?	7
1.2 HVA ER PROSJEKTETS HOVEDOPPGAVE?	7
1.3 HVILKE EFFEKTER VIL PROSJEKTET HA FOR UNINETT?.....	7
1.4 DELOPPGAVER / TIDSBRUK	7
1.5 DOKUMENTASJON	9
1.6 RELATERT ARBEID	9
KAPITTEL 2: INSTALLASJONSARBEIDET	10
2.1 PRENAV INSTALLASJONSKRAV	10
2.2 INSTALLASJONSFASENE.....	10
2.2.1 Sjekk/installasjon av prerekvisitter.....	11
2.2.2 Prekonfigurasjon av prerekvisitter	11
2.2.3 NAV install.....	11
2.2.4 Postkonfigurasjon av prerekvisitter.....	12
2.3 ETTER INSTALLASJON	12
2.4 TILPASNING AV NAV V2.0 (TIL V2.1) MED TANKE PÅ DISTRIBUSJON ..	13
2.4.1 Katalogstruktur.....	14
2.4.2 Enbruger	14
2.4.3 Database.....	14
2.4.4 Konfigurasjonsfiler og kildefiler.....	15
2.5 CVS.....	15
2.6 DISTRIBUSJON AV NAV	16
2.7 NAV VERSJONER	16
2.8 SYSTEM FOR OPPGRADERINGSSCRIPT	17
2.9 NAV LOCAL GREN	18
KAPITTEL 3: DIALOG MED HØGSKOLER OG UNIVERSITET ...20	20
3.1 UiO BETATESTER.....	20
3.2 MANGE HØGSKOLER VISER INTERESSE.....	20
3.3 NAV-USERS@ITEA.NTNU.NO	21
3.4 NAV WORKSHOP	21
3.5 STATUS PÅ INSTALLASJONER	22
3.6 BIDRAG FRA UiTØ.....	22
3.7 BUGTRACKER	22
KAPITTEL 4: NY FUNKSJONALITET I V 2.1	23
KAPITTEL 5: VIDERE ARBEID	25
KAPITTEL 6: OPPSUMMERING	26
VEDLEGG	27
VEDLEGG A: NAV v2 PREREKVISITTKRAV	27
VEDLEGG B: INNSPILL FRA NAV-WORKSHOP.....	28
VEDLEGG C: NAV VERSJON 2.1.1 CHANGELOG	29

Forord

Prosjekt NAVRun er utført av en prosjektgruppe ved ITEA/NTNU på oppdrag fra UNINETT ved Olav Kvittem. Prosjektet har et timeforbruk på 1118 timer og er utført i perioden juni-november 2002. Prosjektdeltagere har vært:

- Vidar Faltinsen (prosjektleder)
- Morten Vold (70% av arbeidet, ansvarlig for installasjonspakken)
- John Magne Bredal
- Kristian Eide
- Sigurd Gartmann
- Knut-Helge Vindheim

Prosjektleder benytter her anledningen til å takke hele prosjektgruppen. En spesiell takk til Morten for en formidabel innsats. Med prosjekt NAVRun har NAV blitt en mye ryddigere ”pakke”!

`./navin.sh` (say no more...).

Vidar Faltinsen
Prosjektleder

Kapittel 1: Innledning

1.1 Hva initierte dette prosjektet?

Prosjekt NAVRun overtar der prosjekt NAVMe slapp. I NAVMe gjorde vi høsten 2001 en pilotinstallasjon av NAV v 2.0 ved UiTø. Vi jobbet mye i prosjektet med å løsrive NAV fra å være en NTNU-spesifikk løsning. Det ble imidlertid *ikke* fokusert på installasjon. Prosjekt-gruppen installerte alt manuelt på pilotmaskinen ved UiTø (og videreutviklet manuelt på NTNUs v2.0 produksjonsmaskin).

For å kunne distribuere NAV i større skala, dvs til alle interesserte høyskoler og universiteter, er det tvingende nødvendig å ha en installasjonspakke og senere oppgraderingspakke. Det er med dette for øye NAVRun fikk sitt mandat.

1.2 Hva er prosjektets hovedoppgave?

NAVRun har til hensikt å gjøre NAV lett tilgjengelig for alle UNINETT's medlemmer. Det skal lages en installasjons- og oppgraderingspakke som gjør det enkelt for en høyskole å laste ned og kjøre NAV på egen hånd. Prosjektet vil også fokusere på opplæring, oppfølging og løpende utbedring av dokumentasjon og programkode (feilretting, mindre utbedringer).

1.3 Hvilke effekter vil prosjektet ha for UNINETT?

Som følge av NAVRun har alle UNINETT medlemmer et stående tilbud om benytte NAV som sitt nettadministrasjonssystem.

1.4 Deloppgaver / tidsbruk

Prosjektplanen definerte 7 deloppgaver, med angitt tidsbruk og fremdrift. Det er gjort arbeid på alle disse deloppgavene, men tyngdefordeling og tidsrammer er forskjøvet. Vi har blåst opp den største deloppgaven i underpunkter og viser i tabellen faktisk og budsjettert tidsforbruk:

Nr	Deloppgave	Regnskap	Budsjett
1	Prosjektadministrasjon	32	50
2	Installasjonsarbeid	602	210
2.1	Avdekke krav, manuelt oppsett, se på avh.	38	
2.2	Design installer	38	
2.3	FTP oppsett, script som lager tarball	38	
2.4	CVS	113	
2.5	Installasjon: Håndtering av prerequisister	75	
2.6	Installasjon: pre- og postconfig	75	
2.7	Installasjon: NAVinstall	150	
2.8	NAV monitor / NAV delstart	30	
2.9	Oppgraderingsscript	45	
3	Systemdokumentasjon	23	70
4	Opplæring høyskoler / universitet	76	35
5	Oppfølging NAV-brukere ved høyskoler	38	70
6	Dokumentasjon	93	35
7	Feilretting, mindre utbedring, løpende drift	254	170
	Sum NAVRun	1118	640

Omtrent 70% av arbeidet i NAVRun har blitt utført av Morten Vold. Han har stått for alt rundt installasjonsarbeidet. Morten er ny i NAV-sammenheng og ble engasjert med installasjonsjobben for øye. Det har vært nyttig å få inn friskt blod, som kan se NAV fra utsiden, og komme med forslag til forbedringer sett fra et ”programpakkeperspektiv”. NAV er utviklet over mange år, av mange ulike personer. Det har vært et koordinert prosjekt med et koordinert mål, men allikevel ikke like konsistent i alle ledd. Mye arbeid er gjort fordi man primært trengte viktige verktøy, ikke fordi man lett skulle distribuere dette til andre.

Morten brukte en del tid i oppstartsfasen på å få oversikt, dernest prøve å installere NAV manuelt (noe som på ingen måte var rett frem). Han har innhentet kravliste fra de ulike utviklerne, og i høy grad utbedret denne etter som nye forhold har blitt avdekket.

Morten har også brukt tid på å etablere en testboks som er egnet på å rulle inn og ut ulike OS raskt og dernest teste NAV installasjoner igjen og igjen.

Vi kommer mer inn på de ulike deloppgavene NAVRun i senere kapittel. Dette fordeler seg slik:

Kap 1.5:	Dokumentasjon	NAVRun deloppgave 3 og 6
Kap 2:	Installasjonsarbeidet	NAVRun deloppgave 2 og 7
Kap 4:	Dialog /oppfølging kunder	NAVRun deloppgave 4 og 5
Kap 5:	Ny funksjonalitet NAV v 2.1	NAVRun deloppg 7 og prosjekt NAVMore

1.5 Dokumentasjon

Det er jobbet kontinuerlig med dokumentasjon i hele prosjektet, alt er lenket opp fra dokumentasjonssiden til NAV (<http://metanav.ntnu.no>).

Dokumentasjonen er ikke helt konsistent i alle ledd, det er et dynamisk arbeid å kontinuerlig utbedre denne. Det er nå også en sammenblanding av engelsk og norsk. Vi har besluttet at vi skal ha et langsiktig mål om å komme over på engelsk som språk. Dette har ikke topp prioritet, men vi ønsker at nye ting vi innfører i database og programkode skal være på engelsk.

Den viktigste dokumentasjon som er laget i NAVRun er:

- Install dokumentasjon, både bruker⁴ og adminvariant⁵ (på engelsk)
- Videreutvikling av NAV v2 admindok⁶ og systemdok⁷
- Konsistent "header" i all programkode som distribueres. Ytterligere kommentarer i kode.
- NAVRun prosjektside⁸
- Sluttrapport

1.6 Relatert arbeid

Prosjekt NAVMore løper parallelt med NAVRun og har fokus på videreutvikling av NAV. NAVMore jobber i hovedsak med NAV v3 funksjonalitet, der endelig målsetning er å jobbe frem en v3 pilot ved NTNU. En del videreutvikling fra NAVMore er likevel tatt inn i v 2.1.0 av NAV og således inkludert i produktet som NAVRun tilbyr. Vi kommer nærmere inn på dette i kapittel 5.

⁴ <http://metanav.ntnu.no/NAVRun/doc/user.html>

⁵ <http://metanav.ntnu.no/NAVRun/doc/developer.html>

⁶ <http://metanav.ntnu.no/v2/admindok/>

⁷ <http://metanav.ntnu.no/v2/systemdok/>

⁸ <http://metanav.ntnu.no/NAVRun/>

Kapittel 2: Installasjonsarbeidet

Arbeidet med installasjonspakken er selve kjernen i NAVRun. Vi legger i dette kapitlet frem resultatet av arbeidet. Vi kommer også inn på en del veivalg som er gjort.

2.1 PreNAV installasjonskrav

Før NAV kan installeres må man naturligvis sette opp en maskin og installere operativsystem. Det settes en del krav til hardware, diskpartisjonering m.m. NAVRun har gitt oss et mer presist bilde av alle krav. Oppdatert informasjon ligger under admindokumentasjon på metanav⁹.

Når det gjelder valg av operativsystem kreves det i det minste en unix-variant. Vi har videre valgt å gi særskilt støtte for Mandrake og RedHat. Med særskilt støtte menes at det er lagt inn støtte for å installere evt manglende prerequisitter for disse operativsystemene. Det er fullt mulig å kjøre NAV under andre unix-varianter, for eksempel FreeBSD¹⁰, men da er man mer på egenhånd. Man må manuelt installere/kompilere prerequisitter. Og vi garanterer ikke at alt vil virke uten videre.

Ved installasjon av operativsystem bør man ellers skjele til NAV sine prerequisittkrav (se vedlegg A) og søke å få med de krav som ligger der. De fleste linux-distribusjoner har mulighet for å velge en ”server-install”, noe som fører til at de viktigste prerequisittene faller på plass.

2.2 Installasjonsfasene

Installeren er designet generelt, med tanke på plattformuavhengighet. Objektorientert perl er brukt, der en generell klasse er definert og spesifikke installere lages per OS som skal støttes (se Navin Developer Documentation¹¹ for mer). Pr dato er det laget en spesifikk installer for:

- ❑ Mandrake 8.2 og Mandrake 9.0
- ❑ RedHat 7.3
- ❑ RedHat 8.0 er under arbeid (ferdig til 15/11-02)

Selve installasjonen er inndelt i fire faser. De er:

1. Sjekk/installasjon av prerequisitter
2. Prekonfigurasjon av prerequisitter (kun PostgreSQL)
3. NAV install
4. Postkonfigurasjon av prerequisitter

⁹ <http://metanav.ntnu.no/v2/admindok/>

¹⁰ Akkurat FreeBSD-støtten er på plass takket være innsats fra UiTø. Dette er omtalt nærmere i kapittel 3.

¹¹ <http://metanav.ntnu.no/NAVRun/doc/developer.html>

Dersom installeren feiler på et punkt vil den avbryte, og man må kjøre forfra igjen. Installeren logger alt den gjør til loggfil, det viktigste vises på skjerm under installasjon.

Det er et bevist valg at installeren setter krav til at prerequisitter er på plass før NAV-installasjon starter. Vi kunne tillatt NAV-installasjon uten å tenke på prerequisitter, men da ville fallgruvene vært langt flere.

2.2.1 Sjekk/installasjon av prerequisitter

Installeren vil her sekvensielt sjekke om alle prerequisittene allerede er installert. En fullstendig liste over prerequisitter er vist i vedlegg A, noen eksempler er: PostgreSQL, Apache, RRDtool. Om en prerequisitt er på plass, vil installeren fortsette. Vi bruker mest mulig generelle metoder for å avdekke om en prerequisitt er installert, slik at dette skal virke for de fleste operativsystem (OS).

Dersom du installerer NAV på et OS vi har implementert spesifikk installer for (Mandrake eller RedHat pr dato), så vil *ikke* installeren feile dersom en prerequisitt mangler. Den aktuelle spesifikke installeren vil da selv installere aktuelle prerequisitt. Dette blir gjort med en frontend til pakkesystemet RPM, dvs urpmi for Mandrake og apt-rpm for RedHat. Foruten å laste ned og installere prerequisitten, så håndterer frontenden evt avhengigheter og løser opp i disse. Nedlasting av de fleste pakkene kan skje fra CD eller over nett mot UNINETT sin ftp mirror (gjelder både Mandrake og RedHat).

Enkelte prerequisitter er også lagt ut på ftp.nav.ntnu.no¹² og hentes der i fra, noen med RPM, noen som tarball og utpakking og kompilering (java og Tomcat hentes som binær, en del perl-moduler kompileres). Vi har også laget en egen RedHat RPM pakke for RRDtool og for Gammu.

2.2.2 Prekonfigurasjon av prerequisitter

Etter at alle prerequisitter er på plass, vil installeren ta kontroll over PostgreSQL. Dette er nødvendig for å ha nødvendige privilegier under databaseoppsett i NAV install. Rettighetene som gis her er midlertidige og skrur av igjen under postkonfigurasjon. Det som i essens gjøres er at det settes opp trust for å logge seg inn uten passord, det åpnes for TCP/IP (men tillates ikke kommunikasjon ut av boksen, kun loopback) og det initialiseres cluster.

2.2.3 NAV install

Selve "NAV install" installerer alle NAV-spesifikke ting. Det omfatter i korte trekk:

¹² ftp.nav.ntnu.no er etablert som en deloppgave i NAVRun, se kapittel 2.6 for mer.

1. Oppretter følgende NAV-grupper i /etc/groups: nav, navroot, navrcs
2. Oppretter NAV-brukeren navcron
3. Gir navcron rettighet til com-port (for mobiltelefon, utsending av SMS)
4. Pakker ut navme-tre under /usr/local/nav/navme
5. Lager local kataloger og localfiler under /usr/local/nav/local (mklocal.sh)
6. Lager nødvendige symbolske lenker
7. Spør om informasjon fra bruker. Input her går inn som variable i nav.conf eller i lokale konfigurasjonsfiler (se under):
 - Domenenavn til installasjonsstedet
 - organisasjonens navn
 - NAV webhostname
 - IP prefiks (adresserommet til installasjonsstedet)
 - NAV administrators epostadresse
 - NAV administrators mobiltelefonnummer
8. Lager lokale konfigurasjonsfiler basert på definerte maler (templates) og brukerinformasjon.
9. Setter filrettigheter.
10. Initialiserer RCS for kildefiler.
11. Installerer Cricket.

2.2.4 Postkonfigurasjon av prerekvisitter

Til slutt gjøres en del postkonfigurasjon. Følgende punkter inngår her:

1. Sjekker at PostgreSQL sin partisjon er på minst 5 GB
2. Settes opp NAV databasene (manage, trapdetect, navlog)
3. Setter opp brukere og passord i PostgreSQL. Slår av trust. Skrur på crypt.
4. Konfigurerer Apache sin http og ssl server. Setter JAVA_HOME i /etc/profile.
5. Konfigurerer Jakarta Tomcat servlet container med Apache bro.
6. Konfigurerer syslogd for å kunne ta i mot cisco syslogmeldinger.

2.3 Etter installasjon

Etter fullført installasjon er alle nødvendige komponenter på plass. Men det er fortsatt en del ting man må sørge for:

- En del systemtjenester må kjøre (og automatisk starte ved reboot)
- To NAVdemoner må kjøre (og automatisk starte ved reboot)
- navcron sin cronjobb må kjøre (fullt eller delvis)

For å gi et statusbilde har NAVRun laget kommandoen `nav status`. Den vil sjekke om alle disse bitene er på plass. Et eksempel på output er:

```

[root@bb]# /usr/local/nav/navme/bin/nav status
== NAV Daemons ==
smsd.pl (pid 17674) is running...
snmptrapd (pid 1631) is running...
== NAV Cron-based tasks ==
database: Up
live: Up
hysterese: Up
cricket: Up
backup: Up
trapdetect: Up

```

nav-scriptet tilbyr videre en enkel måte å starte eller stoppe ulike delsystemer i NAV (`nav start/stop [delsystem]`). De ulike delsystemene fremgår av `nav status` kommandoen, de er nærmere forklart her:

Delsystem	Forklaring
Database	Innsamlingen av data fra rutere, svitsjer m.m. Bygger og vedlikeholder NAVdb, som blir en modell av det virkelige nettverket
Cricket	All RRD/Cricket datainnsamling
Live	Statusmonitoren. Tar seg av pinging (sjekker om alt er oppe). Sender alarmer til Trapdetect.
Hysterese	Terskelmonitoren som "lukter" på RRD-data og sender alarm (til Trapdetect) dersom verdier overstiger 80% av maksverdien.
Trapdetect	Hendelses- og varslingssystemet. Mottar alarmer. Sender epost og SMS basert på brukerprofiler.
Backup	Backup av databasen. Kjører hver natt.

Å starte/stoppe delsystem muliggjør et helt nytt aspekt. Man kan for eksempel velge å bare kjøre database-innsamling for utstyret, og da ikke samle statistikk eller gjøre status-monitorering, varsling osv. Eller man kan velge en annen ønsket delsettingsetning.

Utover å starte NAV, må konfigurasjonsfiler tilpasses, kildefiler og passordfiler fylles. Dette er et kontinuerlig arbeid, slik at installasjonen kan leve i takt med utviklingen i nettet den overvåker.

NAVRun har laget et hjelpescript for raskt å oppdatere databasen og lastinnsamling basert på endringer i kildefilene (endringene vil ellers ikke tre i effekt før påfølgende natt). Scriptet heter `kilde2db.pl`

2.4 Tilpasning av NAV v2.0 (til v2.1) med tanke på distribusjon

Det har vært flere forhold å rydde i for å hensiktsmessig kunne distribuere NAV. Noe av dette ble påbegynt i prosjekt NAVMe og videreført her.

2.4.1 Katalogstruktur

Prinsippet med en todeling av katalogstrukturen er essensiell. NAV-rot-katalog er alltid /usr/local/nav. Under denne har vi to grener:

- navme: Selve "NAV-programmet". Består i essens av en rekke kataloger, script, konfigurasjonsfiler m.m. Ingen filer under navme skal endres lokalt.
- local: Her ligger lokale kildefiler, konfigurasjonsfiler og loggfiler. Lokale tillegg på NAV sin hovedside legges til ved hjelp av lokale html-sider. Tanken er videre at man lokalt kan utvide NAV med egne behov, og at script, dokumentasjon m.m. legges på local-gren (se kapittel 2.9 for mer).

Todelingen av katalogstrukturen startet i prosjekt NAVMe, men er mer konsistent nå. Det er også ryddet i strukturen innen de to trærne. Det er faset ut script som ikke lenger brukes. I noen tilfeller er script slått sammen, omdøpt eller flyttet rundt på.

2.4.2 Enbruker

NAV t.o.m NAVMe spredde seg over tre ulike brukere (cricket, cron og root). Det har også vært en sammenblanding mellom lokale jobber og NAV-jobber for disse brukerne. Med NAVRun er vi over på et enbrukersystem, hele NAV kjøres av en bruker, navcron. Dvs vi forutsetter at en del OS-messige pre-requisitter (Apacheserver, syslogd m.m.) startes og kjøres på vanlig vis. Men alle NAV-cronjobber kjøres av navcron, samt de to NAV-demonene (trapdetect og smsd). For å få til dette har vi i noen tilfeller måtte gjøre suid root (ved pinging).

Enbruker-konseptet forenkler installasjonen, og det er mye ryddigere i seg selv. Ved en slik samling av alle cronjobber, har vi nå også strukturert disse bedre i NAV delsystemer og kan starte og stoppe disse enkeltvis som forklart i kapittel 2.3.

2.4.3 Database

Vi har også ryddet i en del felt i databasen. Vi er smertelig klar over at endringer i databasen etter en release er tungt. Det krever en del arbeid å sømløst jobbe inn en endring her, så det ønsker vi helst å unngå. Større databaseendringer skal *ikke* komme før NAV v3. Mindre justeringer kan tillates, men også det skal sitte langt inne. Med mindre justeringer menes en utvidelse med nye felt i eksisterende tabeller eller utvidelse av feltlengder (det kan nevnes at PostgreSQL så langt gjør det svært tungvint for oss å håndtere for eksempel feltendringer i eksisterende tabeller).

NAVRun har også ryddet opp i mengden av forskjellige databasebrukere og tilgangen til disse. Videre er det ryddet i hvordan de ulike NAV-scriptene kobler seg mot databasen. Her var det essensielt å fjerne informasjon om brukernavn/passord fra de enkelte script. For å løse disse problemstillingene er en ny global konfigurasjonsfil innført, db.conf. db.conf definerer hvilke databaser som finnes (manage, trapdetect og navlog), dernest hvilke brukernavn og passord de ulike scriptene skal bruke i sin oppkobling mot gitt database.

2.4.4 Konfigurasjonsfiler og kildefiler

NAVRun har ryddet i ”jungelen” av konfigurasjonsfiler. Konfigurasjonsmulighetene er også utvidet en del med NAVRun. Følgende forbedringer er gjort:

- ❑ To globale konfigurasjonsfiler: db.conf (nevnt over) og nav.conf. nav.conf inneholder en del global informasjon, som for eksempel navnet på installasjonsstedet (NTNU, UiO etc), admin epostkasse og mobiltelefon, IP prefiks m.m.
- ❑ Alle konfigurasjonsfiler som kan/må tilpasses lokalt er nå samlet på local-gren, under /usr/local/nav/local/etc. Det gjelder også brukerfiler for webtilgang, og kildefiler.
- ❑ Noen konfigurasjonsfiler finnes bare på navme del og skal ikke røres lokalt. Dette brukes av utviklerne.
- ❑ Noen konfigurasjons-/kildefiler finnes i en navme og lokalvariant. To eksempler er: type.txt og ragen.conf. Her gir NAV deg et utgangspunkt, men du har anledning til å utvide, og da gjør du dette på local-gren. Prinsippet er at ved overlapp så overstyrer local-del.

Et nytt mal-konsept er innført. Malfiler (templates) lages for alle nødvendige lokale konfigurasjon- og kildefiler. Malfilene blir lagt i CVS under navme-grenen (i CVS finnes bare navme-gren). Installasjonsscriptet vil generelt sett opprette lokale konfigurasjonsfiler basert på definerte maler. Dette er en svært fleksibel løsning, der vi lett kan definere tomme, eller delvis utfylte lokale konfigurasjons-/kildefiler.

2.5 CVS

Bruk av CVS blir stadig viktigere i NAV. Det fikk sin spede begynnelse i NAVMe-prosjektet, men ble da mest som et felles arkiv for lagring og versjonshåndtering av alle script, en slags RCS i stor skala.

Med NAVRun er potensialet i CVS tatt vesentlig lenger, ved at vi har grenet av en versjon 2 (v2) utviklingsgren (v3 utvikles i head), og ved at vi setter merkelapper som tilsvarer en gitt release, eller en gitt patch av NAV.

De fleste delene av NAV består av kjørbare kildefiler (perl, php, html), men vi har også java, som jo er kompilert kode. I NAVMe la vi binærfilene i CVS, noe som ikke er helt riktig/ryddig. Dette er nå ryddet opp i, CVS inneholder nå i stedet kildefilene til javaprogrammene.

Vi har i prosjektet diskutert CVS politikk og blitt enig om hvordan vi skal grene av og hvordan vi skal sette merkelapper.

2.6 Distribusjon av NAV

Vi har også diskutert hvordan NAV skal distribueres. Som kjent skal NAV tilbys alle høyskoler/universitet som ønsker det. Følgende konsept følges:

- ❑ Vi ønsker ikke at NAV skal bli "fritt vilt". Vi har etablert en ftp-server (ftp.nav.ntnu.no) med brukernavn/passord-kontroll. Alle interesserte høyskoler/universitet får bruker (15 brukere utdelt hittil). Dette gir oss en kontroll, og det letter mulighet for oppfølging.
- ❑ Brukeren mottar et "installasjons-laste-script", navin.sh, som i essens kobler seg opp mot ftp-serveren, henter installasjonspakken, pakker ut denne og sparker i gang selve installasjonsscriptet. (navin.sh sjekker først om det finnes en nyere versjon av seg selv og om så er, bytter den ut seg selv. Vi kan således endre navin.sh på en transparent måte).
- ❑ Merk at navin.sh også blir brukt ved senere oppgraderinger/patcher, se kapittel 2.8.

Utover å sette opp selve ftp-serveren, har vi laget hjelpescript for å lage tarballer av en gitt NAV distribusjon. Her sørges det bl.a. for å kompilere opp java kildekode og inkludere den kompilerte koden i tarballen og fjerne kildekoden.

2.7 NAV versjoner

Installasjonsscriptet er laget generelt, men jobber som følge av NAVRun konkret med installasjon av NAV v 2.1.0. Dette tilsvarer en merkelapp i CVS som er utgangspunktet for NAV-tarballen som legges på ftp-serveren.

Parallelt med NAVRun har NAVMore prosjektet fokusert på utvikling av neste generasjon NAV, nemlig versjon 3. Her er det store strukturelle endringer i database, hendelsessystem m.m. Som nevnt er versjon 2 grenet ut fra hovedgrena i CVS. Prosjektgruppen har videre diskutert hva vi skal tillate av ny funksjonalitet i versjon 2. Vi har landet på:

- ❑ Alle større funksjonelle endringer i NAV tilhører versjon 3, de blir ikke tilbydd som versjon 2. Eksempler er: tjenesteovervåker, nytt varslings-system, historiefunksjon for utstyr, serverstatistikk. NAV v3 vil bli betatestet på NTNU vinteren 2003.
- ❑ Feilretting og mindre justeringer vil bli gjort under NAV v2.1. Dette blir gjort fortløpende etter hvert som ting oppdages. Når en alvorlig nok feil

er avdekket, eller volumet med rettelser er stort nok (skjønnsmessig vurdering), blir det satt en ny merkelapp på versjon 2 grenen, typisk 2.1.1, 2.1.2 etc. Ved hver slik v2 merkelapp lages en tilsvarende patch.

- Det kan bli aktuelt å innføre mindre ny funksjonalitet på v2 (ikke omfattende endringer, ingen databaseendringer). Eksempler kan være sorterte RRD-statistikker, varsling på modulnivå, bedre portrapporter m.m. Dette kommer da som v2.2. For å tillate utvikling av versjon 2.2-kode i CVS, mens bugfixing parallelt skjer på 2.1, grenes 2.1 av (nok en gang). Når 2.2.0 senere slippes, så smeltes 2.1-bugfiks-grenen inn på v2-hovedgrenen (vi bugfikser altså ikke 2.1 etter at 2.2 er sluppet).
- Når v3.0 kommer vil den introdusere store endringer. Det er ikke sikkert alle "kunder" ønsker å oppgradere. Det kan da bli aktuelt å bugfikse på v2 parallelt med videre-utvikling/bugfixing av v3. På dette tidspunktet bør uansett v2 være rimelig stabil.

2.8 System for oppgraderingsscript

På tampen av NAVRun (12/11-02) slipper vi vår første patch (versjon 2.1.1). Se vedlegg C for et overblikk over de bugs som er fikset. Vel så viktig, vi har nå på plass selve ramme-verket for håndtering av patcher. Dette omfatter:

- Det er laget et generelt hjelpescript for å lage en gitt patch-tarball og herunder selve upgrade-scriptet i tarballen.
 - En patch er sin enkleste form bare en utbytting av de filer som er endret. Mer logikk må til om vi endrer struktur/ konfigurasjonsfiler på local gren. Sistnevnte vil vi søke å unngå på versjon 2. Vi ønsker også å unngå endringer i databasen.
 - Merk at vi alltid bytter *hele* filer. Vi kunne basert oss på cvsdiff og kun endret den biten i koden som faktisk var forandret. Å bytte hele filen er mer robust. Da vil vi lettere kunne tillate raske, lokale fikser før en offisiell patch er tilgjengelig.
 - Patch-tarballen vil altså inneholde alle filene som skal erstattes og et upgradescrpt som håndterer oppgraderingen.
- Det er også laget en overbygning som styrer oppgraderingsprosessen. Dette scriptet sjekker installert NAV-versjon mot det tilgjengelige sett av patcher. Scriptet vil dernest hente ned de patchene som trengs for å oppgradere til siste versjon og sekvensielt kjøre i gjennom disse.
- I dette konseptet har vi tatt høyde for kumulative patcher, oppgraderingsscriptet vil da forstå raskeste vei til siste versjon. (Det er ikke sikkert vi vil innføre kumulative patcher, men vi har tatt høyde for det i vårt design).
- Som nevnt i kapittel 2.6 er navin.sh i stand til å bytte ut seg selv. Nyeste versjon av navin.sh vil nå tilby installasjon av patcher etter vellykket installasjon. På den måten vil alle nyinstallasjoner få med seg de siste bugfikser.
- Samme navin.sh med en parameter (-upgrade) vil *hoppe over* installasjonsdelen og kun patche. NAV-administrator trenger således bare å forholde seg til navin.sh.

- Når NAV v3 slippes (tidsramme: sommer 2003), vil en direkte installasjon av NAV v3 tilbys. Dette blir en modifikasjon av installasjonsscriptet for v2. Her har vi alle de viktigste bitene på plass. Vi vil også ved v3 installasjon ta høyde for tilfellet der v2 allerede er installert, slik at oppgradering fra v2 og nyinstallasjon av v3 håndteres i samme løsning.
- Oppgradering fra v2 til v3 vil medføre en del arbeid pga de er store endringer i databasen. Det ligger en særlig utfordring i å få med seg historiske databasedataene (arp, cam, trapdetect).

Merk at både NTNU og UiTø har reinstallert sin egen NAV til 2.1.0. Det har vært viktig å en gang for alle terminere alle ”avarter” av NAV. NTNU og UiTø vil fra nå av gå helt i takt med resten. Vi vil følge de samme prosedyrer for installasjon og oppgradering fremover. På den måten blir også NTNU selv en ivrig bidragsyter til bugrapporteringen.

For å jobbe mer profesjonelt i forhold til utvikling, har prosjekt NAVRun innført en dedikert v2 utviklingsmaskin (sith) som er i stand til å kjøre tester i full (NTNU-)skala. Videre er politikken fra nå av aldri å utvikle på NTNUs egen produksjonsmaskin (bb). Trenger vi fikser på produksjonsmaskinen skal de primært komme gjennom patcher. Vi kan ta snarveier, gitt at patchene bare bytter filer, men vi skal altså *ikke* kjøre ny og *utestet* kode på produksjonsmaskinen (på høy tid).

NAVRun har også innført en tredje maskin (navin) som er dedikert for uttesting av installasjon- og oppgraderingsscript på ulike OS-plattformer. Siste men ikke minst har vi en v3 utviklingsmaskin (bigbud), som også vil kjøre v3 pilot i NAVMore regi (for øvrig samme maskin som har ftp, CVS og metanav).

2.9 NAV local gren

En potensiell ulempe med vår versjonspolitikk er at det kan ta lang tid før ny funksjonalitet kommer med i NAV. Det kan bli tungt å få inn nye ideer. Kreativiteten kan i verste fall bli hemmet, likeså muligheten for bidrag fra andre universitet/høgskoler. Denne problemstillingen tar vi på alvor. Vi må *ikke* tape NAV sin sjel og grunnfilosofi på veien. Prosjektgruppen har diskutert disse aspektene og funnet en løsning.

Man kan utnytte filosofien som ligger bak NAV sin localgren og legge egenproduserte script her (php, perl, hva som helst). Man styrer selv om scriptene skal ligge åpent, begrenset eller med intern adgang. Scriptene vil typisk jobbe mot v2 av databasen og tilby ny funksjonalitet. NTNU har en del slike script allerede, noen eksempler er :

- Script som viser svitsjereporter i ulike varianter
- Sorterte cricket statistikker

- Opplegg som integrerer tftp-aksess mot NAV sin database

Slike egenproduserte script kan i neste instans utveksles mellom NAV installasjoner og settes manuelt i drift på ulike localgrener. Dette vil de mest ivrige ha glede av. Det gir også betatesting av funksjonalitet som kan tas inn i en senere versjon. Konkret vurderer vi å tilby noen av de ovenstående tingene som NAV v 2.2.

Det er viktig at alle localscript utvikles i henhold til et felles sett med konvensjoner. Det vil bl.a. si at man aldri legger brukernavn og passord i script, men bruker db.conf. Det er laget egne bibliotek for perl (nav.pm) og php (getpw) som gjør det enkelt å jobbe på denne måten.

Bidrag fra andre kan inngå i CVS v3 arkiv etter avtale. Dette må avklares i hvert tilfelle i forhold til rettigheter m.m.

Kapittel 3: Dialog med høyskoler og universitet

Den viktigste deloppgaven i Prosjekt NAVRun er utvilsomt installasjonspakken. Men i NAVRun skulle vi gjøre mer, vi skulle få NAV ut, markedsføre det og gi opplæring og veiledning ved behov. Det har vi gjort, og dette arbeidet har gitt resultater. Status idet vi nå avrunder prosjekt NAVRun er at alle 4 universitet og totalt 12 høyskoler har installert eller vil installere NAV i nær fremtid (10 har installert pr dato)! Vi gir i dette kapitlet en kort oppsummering på hvordan vi har jobbet og hvordan vi fremover vil jobbe med informasjon og oppfølging.

3.1 UiO betatester

Det ligger mye arbeid forut for NAV v2.1.0. NTNU startet sin egenutvikling for nesten 4 år siden (vår 1999). Vi har ved flere anledninger demonstrert våre resultater, bl.a. på UNINETTs nasjonale nettmøter. NAV har sakte men sikkert blitt godt kjent for de andre universitetene. UiTø markerte først interesse og ble valgt som pilotinstallasjon i NAVMe-prosjektet sommer og høst 2001. Våren 2002 kom det stadig mer konkrete hint/ønsker fra UiO og UiB. I dette prosjektet ble det da naturlig å velge UiO som betatester av installasjonspakken. Vår fremdriftsplan sprakk imidlertid, vår opprinnelige plan var å innlemme UiO i slutten av juni. Denne planen var svært ambisøs, det har vært en betydelig jobb å få installasjonspakken på plass (jfr kapittel 2). Mer enn antatt måtte vi også bugfikse og rydde i eksisterende struktur. Først 2. oktober ga vi UiO en betaversjon av NAV (v 2.1.0b)¹³. Denne ble testet de neste 2,5 ukene, både ved UiO og ved NTNU, noe som var svært nyttig.

Samarbeidet med UiO har vært upåklagelig, de kom raskt på banen med avanserte spørsmål/utfordringer. Denne dialogen pågår enda, nå kanalisert gjennom epostlisten nav-users@itea.ntnu.no, slik at andre også kan dra nytte av erfaringene/problemstillingene.

3.2 Mange høyskoler viser interesse

Når det gjelder høyskolene, så ble de for første gang skikkelig orientert på UNINETT sitt høyskoleseminar i Trondheim i november 2001. Vi fikk allerede da mange konkrete tilbakemeldinger. Dette var noe mange ville ha! Vi har i prosjekt NAVRun ved flere anledninger informert høyskolene om fremdrift og planer. Vi har oppfordret dem til å melde sin interesse gjennom et web-skjema. I juni hadde vi 6 interesserte høyskoler på vår liste. Når vi så fredag 19/10 annonserte NAV 2.1.0 for tilgjengelig, vokste listen raskt til 12 høyskoler (og vi tror det er flere interesserte der ute).

¹³ Ja! Vi er svært takknemlige for den tålmodigheten UiO har vist ©

3.3 nav-users@itea.ntnu.no

Vi har opprettet en egen majordomo epostliste (nav-users@itea.ntnu.no) som nå fungerer som vår primære NAV informasjonskanal. Det er vårt ønske at epost-listen skal etablere seg som et nyttig forum for utveksling av erfaring, samt som en kanal for å stille spørsmål til utviklerne. Pr 11/11 er det 36 stykker påmeldt listen, og dette vokser stadig. Vi ser allerede stor nytte av listen, det har kommet bidrag fra flere høyskoler/universitet, pr dato har vi sett over 100 postinger.

3.4 NAV Workshop

5. november gjennomførte vi en heldags NAV workshop i UNINETTs lokaler i Trondheim. Her var det hele 37 deltagere (oss inklusive). Listen over representerte universitet/høyskoler er vist under, NAV-status er angitt i parentes:

1. NTNU (operativ)
2. Universitet i Tromsø (operativ)
3. Universitetet i Oslo (testdrift)
4. Universitetet i Bergen (testdrift)
5. Høgskolen i Oslo (testdrift)
6. Høgskolen i Molde (testdrift/operativ)
7. Høgskolen i Narvik (testdrift)
8. Høgskolen i Vestfold (installert)
9. Høgskolen i Hedmark
10. Høgskolen i Volda (installert)
11. Høgskolen i Nord-Trøndelag
12. Høgskolen i Telemark (installert)
13. Høgskolen i Agder
14. Høgskolen i Bergen
15. Høgskolen i Tromsø
16. Høgskolen i Lillehammer

Workshopen besto av en informasjonsdel, der vi orienterte om status og om tekniske aspekter av NAV sett fra et administrasjonsperspektiv. Vi arrangerte videre et gruppearbeid, der vi hadde konkrete oppgaver for å gi "hands-on" erfaring med NAV. De som allerede hadde installert NAV jobbet på sin installasjonsmaskin (via medbrakt bærbar og trådløst nett). Vi hadde også to NTNU-maskiner tilgjengelige for de som ikke hadde installert.

Opplegget fungerte (tilsynelatende) bra, vi fikk svært mange spørsmål, samt innspill og ideer. De viktigste innspillene er gjengitt i vedlegg B. Oppgavene vi laget kan sees under NAVRun-prosjektsiden¹⁴.

¹⁴ <http://metanav.ntnu.no/NAVRun/oppgaver.html>

3.5 Status på installasjoner

Når vi nå avrunder prosjekt NAVRun har alle fire universitet og totalt seks høyskoler installert NAV, flere er på trappene. Det er varierende hvor langt de enkelte har kommet. Pr dato så har foruten NTNU og Universitet i Tromsø, Universitetet i Oslo, Høgskolen i Molde, Høgskolen i Oslo og Høgskolen i Narvik og Universitet i Bergen kommet lengst. Selv om prosjekt NAVRun som sådan termineres, vil vi kontinuerlig følge opp gjennom nav-users og tilby hjelp og veiledning til eksisterende og nye installasjoner.

3.6 Bidrag fra UiTø

Vi ønsker også å nevne et konkret og svært gledelig bidrag fra UiTø. De har tatt tak i vår installasjonspakke og laget en spesifikk installer for FreeBSD. De har også utbedret noen biter i den generelle installeren som viste seg å være for linux-spesifikk. Dette arbeidet er nå innlemmet i den eksisterende installasjonspakken (UiTø har ikke vektlagt installasjon av prerequisitter, det forutsettes at dette gjøres manuelt). Merk at UiB også har installert på FreeBSD og således nytt godt av arbeidet UiTø har gjort!

3.7 BugTracker

Vi har innført et web-basert verktøy (phpBugTracker, open-source-løsning) for rapportering og sakshåndtering av bugs. Dette er et rimelig åpent system, hvem som helst kan registrere seg (med sin epostadresse) og rapportere inn bugs. Utviklerne får beskjed når nye bugs er funnet i deres kode, og vil fikse (på vår utviklingsmaskin), sjekke fiksen inn i CVS og rapportere inn status via samme verktøy. Som før nevnt, vil vi etter gitte kriterier samle slike bugfikser i en NAV-patch som gjøres tilgjengelig.

Kapittel 4: Ny funksjonalitet i v 2.1

NAVRun introduserer NAV v2.1. Det er en del ny funksjonalitet i v2.1 i forhold til v2.0. Det som er essensielt for installasjonen ble nevnt i kapittel 2.4. I tillegg har vi valgt å inkludere en del NAVMore arbeid. Her oppsummeres kort hva som er nytt i forhold til versjon 2.0 (dette er mer utførlig forklart i sluttrapporten til prosjekt NAVMore):

Camlogging (NAVMore deloppgave 2)

- Innført for alle svitsjer (SW og KANT)
- Ingen innsamling for uplink/downlink-porter

Datainnsamling (NAVMore deloppgave 3)

- Bedre støtte dersom ruterport descriptionfelt ikke følger NAV konvensjon (descriptionfeltet blir lagt inn som nettidet)
- Autoavledning av nettype dersom denne ikke er satt manuelt (i description eller vlan.txt).
- Bedre mekanisme for å tolke vlan (automatisk av virtuelt interfacenavn, eller som en utvidelse til description konvensjon). vlan.txt er ikke lenger et krav, det er frivillig om man ønsker å bruke kildefilen.
- Generelt bedre støtte for å samle inn data etter beste evne. Dersom noen data mangler/er inkonsistent setter vi inn det vi har og rapporter resten gjennom feilmelding.
- Flere feil forbundet med utfasing/ending av utstyrskonfigurasjon er rettet.
- Håndtering av nye utstyrstyper
 - HP kantsvitsjer (HP Procurve 2524 med virtuell stacking)
 - Catalyst 6500 native mode (GSW). Medførte mye arbeid, da dette introduserte en ny kategori: GSW, som er en hybrid mellom en ruter (GW) og en svitsj (SW). På en Cat6500 kan man vekselvis sette porter til å være svitsjeporter eller ruterporter.
 - Catalyst 4000 sup II, Catalyst 3550, Catalyst 2950. Problemene her var nye OIDs for en del variable vi samler inn.

Rapportgeneratoren (NAVMore deloppgave 8)

- Rapportert som funksjon av sikkerhetsnivå
- Lokale rapporter

Nettkartet (vlanplot)

- Inndelt sikker del / åpen del
- Adminpanel styrket
- Endret popupfunksjonalitet
- Støtte for GSW

Varsling

- SMS historikk
- Vi har byttet ut mygnokii med gammu (mygnokii2).

Adresserom

- Grafisk visning

navlog

- ❑ Strukturert håndtering av cisco syslogmeldinger (NAVMore deloppg 5)
- ❑ System for NAV feilmeldinger (NAVMore deeloppgave 6)

Se også vedlegg C som viser 2.1.1 changelog (altså bugfiks som kom med i første patch).

Kapittel 5: Videre arbeid

Videre arbeid som nevnes her går *ikke* på ny funksjonalitet i NAV, det blir nevnt i NAVMore sammenheng. Videre arbeid etter NAVRun blir:

- ❑ Å følge opp kunder gjennom epostliste (nav-users) og direkte epost/telefon.
- ❑ Feilretting av v2 ved behov
- ❑ Lage patcher / oppgraderingsscript
- ❑ Se på NAV v3 installasjonspakke (ved å modifisere v2 install)
- ❑ Se på NAV v2 til v3 upgrade. Her blir det en del utfordringer pga store strukturelle endringer i databasen.

Kapittel 6: Oppsummering

NAVRun har fullført sin målsetning og laget en installasjonspakke for NAV. Denne er pr 19/10 2002 gjort tilgjengelig for alle universitet og høyskoler som ønsker det. Det er v 2.1 av NAV som nå er gjeldende og denne er styrket en god del i forhold til piloten som kjørte ved UiTø under prosjekt NAVMe.

NAVRun har ryddet vesentlig i filstruktur, slik at CVS nå brukes mer gjennomført og vi har et konsept for å håndtere oppgradering/patching. Det er også ryddet i databaseaksess og struktur. Alle cronjobbene er samlet under *en* bruker (navcron).

Selve installasjonsscriptet er implementert på en gjennomført måte, der det først sjekkes om alle prerequisitter er installert. Om ikke vil disse bli installert ved å bruke urpmi (Mandrake) eller apt-rpm (RedHat). Vi håndterer også nødvendig konfigurasjon av prerequisitter (PostgreSQL, Apache, Tomcat, syslog).

Det er laget verktøy for å starte eller delstarte NAV, samt å monitorere at alle delsystemer kjører.

Med NAVRun er NAV klar for distribusjon. Vi ser allerede stor interesse, det var 12 høyskoler og 4 universitet tilstede på NAV workshopen vi arrangerte 5. november (totalt 37 deltagere). Når prosjekt NAVRun nå termineres har 6 høyskoler og alle 4 universitet allerede installert NAV, flere er like rundt hjørnet.

100% inhabil erklærer vi prosjekt NAVRun for en suksess.

Vedlegg

Vedlegg A: NAV v2 prerequisittkrav

Installasjonspakken til NAV sjekker om alle prekursitter er installert, dersom de ikke er det vil den selv gjøre dette (for RedHat og Mandrake). Listen over prekursitter er gjengitt under. I noen tilfeller settes det også krav til versjon, enten minstekrav (\geq) eller eksakt krav ($=$):

```
gawk
sed
grep
cc
make
/bin/mail
syslogd
apache           >= 1.3.0
suexec
suidperl
mod_perl
mod_php          >= 4.10
mod_ssl
psql             >= 7.2.0
postgresql      >= 7.2.0
php-pgsql
php-gd
freetype
perl-Term::ReadKey
perl-TimeDate
perl-Mail::Sendmail
perl-CGI
perl-SNMP_Session
perl-DBI
perl-DB_File
perl-Pg
perl-Digest::MD5
ucd-snmp
libsnmp0-devel
libopenssl10-devel
perl-SNMP
RRDtool
gammu
rcs              >= 5.0
java             => 1.3.0
Tomcat           == 3.3.1
```

Vedlegg B: Innspill fra NAV-workshop

NAV-workshopen 5/11 ga oss en del innspill på funksjonalitet som er savnet. Det har også kommet innspill på nav-users (og det vil fortsatt komme). Oppsummert så langt:

- Cricket :
 - Rapporter inn de tillegg vi har gjort til Cricket-utviklere
 - Kjør fortrinnsvis standard versjon
 - Ønske om å ha hele/deler av cricket-data på NAV begrenset
 - Se på 64bits teller-løsning
 - WLAN-støtte

- Websiden
 - ”Logg ut” knapp
 - Farge ”logg inn” knapp - bortgjemt

- Rapportgenerator
 - Funksjon for å liste alle (lokale) rapporter
 - Link fra rapport til cricket
 - RCS for ragen.conf

- Maskinsporing
 - MAC/IP bak en svitsjeport
 - Deteksjon av mac-adresser som ”hopper rundt” mellom svitsjeporter

- Nettkart
 - Mulighet for å se lag2-topologi for link og stam
 - Feilmelding dersom Tomcat ikke kjører
 - Mulighet for å begrense mer info på åpen del

- fyll_db.pl
 - Mer loggmeldinger til skjerm
 - Mer intuitive feilmeldinger
 - Beskjed om manglende snmp v2 (getbulk) støtte (jfr Narvik)
 - Bug ved forsøk på å fjerne adresserom-prefiks

- Brukerprofiler (i forbindelse med varsling)
 - Kun lov å sette opp varsling for registrerte web-brukere

Vedlegg C: NAV versjon 2.1.1 changelog

Vi dokumenterer (på engelsk) alle bugfiksene i en CHANGELOG-fil. Denne følger med patchen. I første patch (versjon 2.1.1) ble følgende fikset:

- ❑ makecricketconfig:
 - Removed warnings about uninitialized values.
 - Support interface-names with spaces.
 - Now detects sub-interfaces and adds sub-interfaces as target-type in the config file.
- ❑ Other Cricket-related changes:
 - Changed default ds-type to DERIVE, and added a rrd-min = 0 to each datatype so that negative values are omitted. This will hopefully get rid of most of the spikes that occur when counters wrap on interfaces. This will also result in data not being collected when the data-value is negative.
- ❑ Root-check removed from SMS Daemon init script, as it prevented safe_smsd from being able to restart a missing SMS daemon.
- ❑ Syntax errors were fixed in several scripts.
- ❑ Maximum length of interface names in the gwport table was extended from 30 to 50 characters.
- ❑ When clicking on IP addresses in ragen, a list of all IP's would be shown instead of just the one that was clicked. This has been fixed.
- ❑ The login link of the webpage is now more visible.
- ❑ Ragen displayed a slightly skewed duplex information line for ios-switches, this is now fixed.
- ❑ Ragen's room report now also reports equipment count. Unused reports were also removed from the configuration.
- ❑ Error messages from cron-based tasks were sent to a hard-coded mail address at ITEA, NTNU. These mails are now sent to the address defined as ADMIN_MAIL in nav.conf.
- ❑ Address space problems were fixed in subnet2web.php.
- ❑ New equipment types were added to systemwide type.txt.
- ❑ Changing vlan on a switch-port caused the switch-port information to disappear from Ragen, this has been fixed.
- ❑ last.pl would use sysname instead of type to determine equipment type, resulting in missing data for some links in vlanPlot. This was horribly wrong, and has been fixed.
- ❑ A bug in last.pl would report that load data was found even if it was not, resulting in zero-load (black) links in vlanPlot.
- ❑ vlanPlot would not allow browsing of vlans, because the vPServer had not been updated after a database change. vPServer has now been updated.