

KLACHTEN EN MENINGEN VAN SARA-GEBRUIKERS
BINNEN HET MATHEMATISCH CENTRUM

verzameld door

Paul Klint

KLACHTEN EN MENINGEN VAN SARA-GEBRUIKERS BINNEN HET MATHEMATISCH CENTRUM
verzameld door
Paul Klint.

0. WAAROM DIT RAPPORT?

Sinds geruime tijd bestaat er binnen het MC ontevredenheid over de computerfaciliteiten die door SARA (Stichting Academisch Rekencentrum Amsterdam) -het gemeenschappelijke rekencentrum van Universiteit van Amsterdam, Vrije Universiteit en Mathematisch Centrum- geboden worden.

Aangezien de apparatuurkommissie, die zich buigt over het probleem van uitbreiding van de SARA-apparatuur, ons verzocht heeft op zeer korte termijn eventuele klachten te melden, is een inventarisatie van de ontevredenheid aktueel. Om aan dit verzoek te voldoen zijn *gedurende drie werkdagen* klachten van de SARA-gebruikers binnen het MC verzameld. Deze klachten (en meningen) vindt u gerubriceerd terug in de volgende hoofdstukken. Het beeld dat door deze klachten opgeroepen wordt kan uiteraard noch volledig noch onpartijdig zijn. Het kan echter wel dienen als illustratie van de heersende ontevredenheid.

In hoofdstuk 3 wordt de kritiek samengevat in een aantal hoofdpunten. Hoofdstuk 4 bevat suggesties voor een eventuele apparatuuruitbreiding.

Tenslotte wijzen wij u nog op "A critique on the Cyber", geschreven door A.S. Tanenbaum [1], waarin zeer uitgebreid ingegaan wordt op met name de architectuur van de CYBER.

1. HARDWARE

1.1 *Architectuur CYBER*

1.1.1 Arithmetiek

De beperkte overflow detectie en slechte arithmetiek (afronding en dubbele lengte operaties) geven regelmatig aanleiding tot problemen.

1.1.2 Characterset en 60-bits woorden

De kleine (63) characterset maakt tekstverwerking al bij voorbaat vrijwel onmogelijk. Er zijn bijvoorbeeld alleen maar hoofdletters en geen kleine letters aanwezig. Hierbij komt dat geen byte-adressering beschikbaar is om gemakkelijk met characters te manipuleren. Uiteraard kan "om dit probleem heen" geprogrammeerd worden, maar dat moet dan wel gebeuren in ieder programma dat met characters manipuleert.

Tengevolge van de 60-bits woorden (waar 10 6-bits characters in passen) moet iedere regel aangevuld worden met (onnodige) spaties om het laatste woord helemaal te vullen. Een ander nadeel van de 60-bits woorden betreft het instructieformaat. Aangezien machine-instructies 15, 30 of 60 bits in beslag nemen, is het niet altijd mogelijk een woord geheel te vullen. Hierdoor blijft een gedeelte van een dergelijk woord ongebruikt en expanderen objektprogramma's.

1.1.3 Registers

De registerorganisatie maakt het moeilijk om goede objektcode te genereren. Bovendien moet de gegenereerde code zo compact mogelijk zijn (zie 1.1.2). Hierdoor is veel detailwerk vereist om goede codegenerators te schrijven. Zie met name WIRTH [2] en SHAPIRO & SAINT [3]. Bovendien is voor dit doel geen standaard software beschikbaar.

1.1.4 Geheugenorganisatie

De afwezigheid van virtueel geheugen (segmentatie, paging) op de CYBER heeft ernstige gevolgen. Aan de hand van een viertal voorbeelden zullen we laten zien dat dankzij virtueel geheugen een groot gedeelte van de capaciteitsproblemen op het CYBER systeem zich niet voor zou doen, bij een gelijkblijvende hoeveelheid hulpmiddelen (grootte van het geheugen, aantal disks).

Kapaciteitsvergroting betekent dat meer en grotere interactieve jobs (b.v. 60-100) tegelijkertijd in executie kunnen zijn en zeer grote batch-jobs gedraaid kunnen worden met een redelijke turn-around tijd. Deze capaciteitsvergroting wordt bereikt doordat maar een gedeelte van het virtueel geheugen (dat meestal vele malen groter is dan het "echte" (kern)geheugen) in het fysische geheugen aanwezig is. Dankzij deze organisatie hoeft maar een gedeelte van iedere job op één moment in het fysische geheugen aanwezig te zijn. Niet gebruikte stukken van een programma (b.v. het foutmeldingsmoduul van een compiler) komen zelfs helemaal niet in het fysische geheugen. Tenslotte kunnen files ook in het virtueel geheugen staan (MULTICS, OS/VS2), zodat de gedeelten van files die het meest gebruikt worden in het fysisch geheugen kunnen verblijven, waardoor file-accessen versneld worden en de totale executietijd van een job vermindert. Per slot van rekening wordt naar sommige files vaker gerefereerd dan naar programma's (sorteren).

Verminderde software produktiekosten kunnen verwacht worden doordat grote programma's minder complex en flexibeler worden. Het is op de CYBER namelijk noodzakelijk om voor grote programma's en databestanden "software paging" toe te passen, omdat zij òf niet in het geheugen passen òf zoveel geheugen nodig hebben dat een onaanvaardbaar lange turn-around tijd ontstaat. Op de CYBER moeten programma's in een expliciete overlay structuur geschreven worden, terwijl door een virtueel geheugen-systeem deze taak juist overgenomen wordt, op een voor de programmeur transparante wijze. Verminderde produktiekosten van programmatuur zijn het direkte gevolg.

Het zojuist genoemde voordeel is zo mogelijk nog duidelijker bij databestanden. Op de CYBER moet voor ieder file-access door de *software* getest worden of de gewenste data wel in het (kern)geheugen aanwezig zijn. In een virtueel geheugen-systeem wordt deze test door de *hardware* uitgevoerd. Bovendien moet men bij de CYBER-organisatie een vaste structurering in de databestanden aanbrenge. Men moet zich namelijk van tevoren bezighouden met de vraag welke gedeelten van het bestand in het (kern)geheugen en welk gedeelte op achtergrondgeheugen bewaard moet worden. Een verminderde flexibiliteit en een toegenomen complexiteit van programma en data zijn het gevolg. Als bovendien achteraf blijkt dat een gekozen structurering onjuist

of minder gunstig is, kost het veel werk om programma en data aan een betere structurering aan te passen. Metingen in BRAUN & GUSTAVSON [4] wijzen uit dat in de meeste gevallen een virtueel geheugen-systeem betere beslissingen kan nemen dan de programmeur inzake de verdeling van programma en data over fysisch geheugen en achtergrondgeheugen.

De genoemde punten zullen ook een positieve invloed hebben op de kwaliteit en betrouwbaarheid van de systeemsoftware als geheel.

"*Shared memory*", een ander aspekt van virtueel geheugen, maakt het mogelijk dat meerdere gebruikers kunnen beschikken over één kopie van een programma. Dit in tegenstelling tot de CYBER-organisatie. Als op de CYBER 10 jobs de FORTRAN compiler gebruiken, moeten er 10 kopieën van die compiler in het geheugen staan. Hierbij kan zich de paradoxale situatie voordoen dat de kopie van de éne gebruiker juist naar achtergrondgeheugen geswapt wordt, terwijl de kopie van een andere gebruiker van achtergrondgeheugen weer (op dezelfde plaats?) in het (kern)geheugen gehaald wordt. Het gevolg is overbodig swappen van jobs en de moeilijkheden met multi-user jobs (zoals de editor).

Tenslotte wordt op de CYBER een *gebrek aan protektie* veroorzaakt door het ontbreken van een onderscheid tussen programma- en datasegmenten. Dit betekent dat er geen hardware schrijfprotektie van programmasegmenten is en foutieve programma's executeerbare code kunnen overschrijven; het hoeft geen betoog dat dit kan leiden tot zeer moeilijk te lokaliseren fouten.

Na bovenstaande bespreking van capaciteitsvergroting, vermindering van software produktiekosten, shared memory en protektie, resteren nog twee opmerkingen ten aanzien van virtueel geheugen.

Ten eerste zijn er programma's waar een virtueel geheugen-systeem minder gunstig voor is. Men kan hierbij denken aan een lijstverwerkingstaal zoals LISP. Al sinds lang zijn er echter algorithmen bekend om dit ongunstige effect te compenseren (BOBROW & MURPHY [5], FENICHEL & YOCHELSON [6]).

Ten tweede vestigen wij er de aandacht op dat zelfs CONTROL DATA ertoe overgegaan is zijn nieuwere machines (CDC-star, MP 17) van virtueel geheugen te voorzien!

1.2 *Peripherals en diversen*

- De terminals zijn vaak defekt en maken moeilijk te controleren zendfouten. Verder is er voor een gebruiker van een teletype-achtige terminal meestal geen verschil te zien tussen een programma in een loop, een programma dat nog niet in executie is genomen of het "down" gaan van het systeem. De verbinding wordt namelijk niet verbroken.
- De kaartponsters op central site zijn niet betrouwbaar, dan wel dubbel geponste kaarten worden niet verwijderd.
- De bandponsters maken zéér veel ponsfouten. Bovendien raken ponsbanden vaak weg.
- De PDP11-kaartlezer-regeldrukker combinatie bij de MC-balie is meerdere malen per week defekt.
Het is vaak niet duidelijk of het hier hardware of software storingen betreft. Het gevolg is dat er dan vanuit het MC geen kaartprogramma's meer gedraaid kunnen worden.

2. SOFTWARE

2.1 SCOPE systeem

2.1.1 Algemene organisatie

Het zou te ver voeren om hier uitgebreid op moderne systeemprogramme-ringstechnieken in te gaan. We volstaan met het signaleren van een aantal punten die met name illustreren dat het SCOPE systeem verouderd is.

Ten eerste is er *geen automatische deadlock*-preventie, hierdoor moeten jobs regelmatig door operateursingreep "gedropt" worden om deadlock situ-aties op te lossen. Herstarten van gedropte jobs geeft moeilijkheden of blijft achterwege.

Ten tweede is er *geen protektie van systeem-requests*. De kommunikatie tussen systeem en gebruikersprogramma vindt plaats door middel van een kommunikatiewoord (RA+1) dat in de adresruimte van het gebruikersprogramma ligt. Hierdoor kunnen verzoeken aan het systeem overschreven worden.

Ten derde werkt het systeem met *tabellen van vaste grootte*. Dit betekent dat voortdurend problemen zullen blijven ontstaan bij groei van het systeem. Dergelijke problemen doen zich op dit moment al met de editor en permanent file directory voor. Een ander recent voorbeeld betreft de toewijzing van gebruikers-identifikaties (user-id). Aanvankelijk had iedere gebruiker een vaste user-id van twee symbolen. Bij groei van het aantal gebruikers was deze user-id niet meer toereikend en moest overgegaan worden tot een systeem van dynamische toekenning van user-id's.

Dit vereist onnodig, extra systeem programmeerwerk.

Ten vierde is het systeem gebaseerd op *polling*. Dit betekent dat ter- minals die niet in gebruik zijn toch voortdurend geïnspekteerd moeten worden, om te kijken of ze *nu* actief zijn geworden. Het is duidelijk dat een on- nodige systeembelasting het gevolg is. Een systeem dat op interrupts ge- baseerd is, heeft deze nadelen niet.

Tenslotte werkt de vaste taakverdeling van CPU en peripheral processors (PPU) een sub-optimaal gebruik van totale resources in de hand. De gebruiker heeft geen toegang tot de PPU's die op een zeker moment niet door het systeem benut worden. Overigens wordt deze opzet ook weer door de hardware gedikteerd, omdat de PPU's geen protektie kennen.

De bovenstaande problemen worden geïllustreerd door de *onbetrouwbaarheid* van het systeem als geheel: *het gaat vele malen per week (soms zelfs per dag) "down"*.

2.1.2 Filestysteem

De nu volgende bezwaren kunnen allemaal ondergebracht worden in de categorieën *onveilig* en *onhandig*.

- Het is onhandig dat niet meteen na het "inloggen" een gebruiker al zijn files tot zijn beschikking heeft, maar deze expliciet moet "attachen". Bovendien is het aantal files dat tegelijkertijd in gebruik mag zijn beperkt (vaste tabellen!).
- Er is geen faciliteit om, op een voor de gebruiker transparante wijze, lange tijd niet gebruikte files naar tape te schrijven.
- Aan files op tape kan geen naam gegeven worden; de gebruiker moet de tape expliciet positioneren op de gewenste file, maar helaas:
- SKIPF en SKIPB werken niet goed op tape. Hierdoor moeten soms hele tapes naar de disk gekopieerd worden als men een file aan het einde van de tape nodig heeft, om over de verspilde diskruimte maar te zwijgen.
- Lees- en schrijfsleutels werken niet goed op tape. Hierdoor lopen tapes voortdurend het gevaar per ongeluk overschreven te worden.
- De vierdaagse purge periode gekombineerd met de slechte backup-faciliteiten hebben tot gevolg dat regelmatig files per ongeluk verloren gaan. Het is hierbij zeer hinderlijk dat de dagen in het weekeind meetellen.
- "Rewinden" en "connecten" van files is soms onduidelijk. Het is met name verwarrend dat sommige systeemprogramma's hun in- en uitvoer files eerst rewinden en/of connecten voor gebruik en andere dat weer niet doen.
- Bij default worden files *zonder protektie* gekatalogiseerd.
- Het is niet mogelijk om eigen disk packs te gebruiken (privacy!).
- Het is soms niet mogelijk om een permanent file te kreëren (permanent file directory vol: vaste tabellen!).

2.1.3 Diverse klachten over het systeem als geheel

- De jobcontrollanguage is primitief; het is bijvoorbeeld niet mogelijk voor de gebruiker, zelf kommando's te definiëren of een file met kommando's te laten executeren. Verder is het niet mogelijk om vanuit een gebruikersprogramma een ander (systeem)programma aan te roepen, dat wèl via een controlcard bereikbaar is. Vrijwel alle operatingsystemen bieden faciliteiten op dit punt.
- De parsing van SCOPE controlcards is niet gecentraliseerd of gestandaardiseerd; hierdoor moet ieder programma zijn eigen controlcard interpreteren. Een grote verscheidenheid in parameterscheiders en -namen is het resultaat.
- De recovery na jobterminatie geeft problemen. Verschillende systeemprogramma's en utilities (het FORTRAN run-time systeem bijvoorbeeld) falen op dit punt. Tengevolge hiervan wordt o.a. niet alle output geleverd die al geproduceerd was op het moment van terminatie.
- De identifikatie van jobs via een (steeds wisselend) nummer is onhandig. Een praktische vorm van job-identifikatie, namelijk met behulp van "dispose" vanaf een terminal, kan niet gebruikt worden, omdat geen mailingcode opgegeven kan worden.
- De foutmeldingen zijn vaak onduidelijk tot onbegrijpelijk. Het opsporen van de betekenis van een foutmelding is vaak een langdurig en hoogst frustrerend zoekproces.
- De manuals zijn slecht en/of onduidelijk en/of onjuist. Het SCOPE 3.4.1 manual (het huidige operating systeem!) is niet meer verkrijgbaar.
- Er is voortdurend verwarring over de vraag of getallen decimaal of oktaal zijn. Zo moet de rekentijd-limiet bijvoorbeeld oktaal opgegeven worden, terwijl de verbruikte rekentijd decimaal afgedrukt wordt.
- Er zijn minieme verschillen tussen de door CDC gebruikte charactercodes (ASCII, EBCDIC) en de internationale standaards. Het zijn juist de *zeer geringe* afwijkingen die tot fouten aanleiding geven.
- De systeemklok is onnauwkeurig en vertoont niet-reproduceerbare afwijkingen.

2.2 Intercom

2.2.1 Responstijd

Afgezien van het grote aantal klachten over de editor, is het meest genoemde bezwaar van INTERCOM de veel te lange responstijd. Hoewel dit in de huidige situatie moeilijk voor te stellen is, wordt een responstijd van ongeveer *één seconde* in andere systemen als *normaal* beschouwd voor het uitvoeren van edit commando's en andere niet rekenintensieve opdrachten. Beproefde systemen als MULTICS, MTS en VM/370 voldoen aan deze eis bij een belasting met 50 tot 80 terminalgebruikers; de betreffende computers zijn qua grootte met de CYBER vergelijkbaar! *INTERCOM biedt echter een responstijd van 10 tot 30 seconden*. Gezien deze ervaring lijkt het minder verstandig om het aantal terminals uit te breiden (hoewel hieraan wel behoefte bestaat); dit teneinde een verdere overbelasting van het systeem te vermijden.

Ook hier wreekt zich het feit dat de CYBER niet ontworpen is voor time-sharing. Dit betekent dat gewone edit werkzaamheden het systeem onnodig zwaar belasten. Een drastische verbetering van de responstijd zou dan ook verwacht kunnen worden, als het edit werk overgelaten zou worden aan één of meer front-end miniprocessors (zie hoofdstuk 4).

2.2.2 Editor

Alle andere klachten over INTERCOM hebben betrekking op de editor. In vergelijking met goede editors (QED [7] en TECO [8]) blijkt de INTERCOM editor minder hanteerbaar. Bovendien ontbreken verschillende zeer gewenste primitieve operaties zoals verplaatsen van regels, invoer vanaf verschillende files enz. Uit recente VIM newsletters [9] blijkt dat de bezwaren tegen de huidige INTERCOM editor onder veel gebruikers leven. Een greep uit de klachten:

- De editor is onbetrouwbaar: soms krijgt men de edit file van iemand anders te zien ("file-mixing"). Dit maakt de editor onbruikbaar voor gebruikers met geheime files. Bovendien is in veel gevallen de editor de oorzaak van het "down" gaan van het systeem.
- De editor is duur in gebruik: met name het zoeken naar substrings blijkt veel CPU tijd te kosten.

- De regelnummers worden achteraan de regel geplaatst, hierdoor bestaan de meeste regels voor de helft uit zinloze spaties (verspilde diskruimte!). Het regel georiënteerde karakter van de editor maakt het onmogelijk een regel gemakkelijk te splitsen.
- De default regelnummering (begin bij 100 en increment 10) wijkt af van de nummering van alle compilers. Bovendien nummeren de compilers onderling verschillend.
- Het standaard edit-formaat is op FORTRAN gericht. Dat betekent dat in ALGOL programma's de ";" in de eerste 7 posities verdwijnt.
- De gebruiker kan geen edit-macro's definiëren om veel door hem gebruikte series kommando's af te korten. De behoefte hieraan ontstaat o.a. doordat maar één kommando per regel ingetypt kan worden en de responstijd zo slecht is.
- Er is geen tekstvervanging in gegeven kontekst: b.v. alle strings "abc", "123",... vervangen door <abc>, <123>,... .
- Een "move" kommando ontbreekt om stukken tekst in de edit file te verplaatsen.
- Er is geen "merge" kommando om twee of meer files volgens regelnummer te mengen.

2.3 *Batch*

In feite doen zich bij de batchverwerking dezelfde overbelastingsverschijnselen voor als bij INTERCOM. Het voornaamste bezwaar is dan ook de slechte turn-around tijd van grotere jobs. Twee à drie dagen is geen uitzondering. Zelfs minder grote jobs ($cm \leq 120.000$, $T \leq 100$) krijgt men soms niet eens binnen één dag terug.

Twee andere klachten zijn:

- de uitvoer van batchjobs raakt vaak zoek;
- het identifikatie systeem met steeds wisselend nummer is erg onhandig: de gebruiker moet zelf een administratie bijhouden van nummers van nog niet afgehaalde output (zie 2.1.3).

2.4 Programmeertalen

Als men het aantal en de kwaliteit van implementaties van hogere programmeertalen overziet, is een gevoel van grote teleurstelling op zijn plaats. De meeste implementaties bevatten onnodige beperkingen en fouten. De door vertalers geproduceerde objectcode laat soms zeer veel te wensen over (zie 1.1.3). Bovendien is de objectcode *altijd* onveilig, d.w.z. bepaalde run-time tests blijven achterwege.

De nu volgende klachten zijn wel bijzonder onvolledig, maar betreffen dan ook uitsluitend problemen die verzameld zijn binnen de in de inleiding genoemde drie werkdagen. Er is niet naar gestreefd een volledige vergelijking van alle programmeertalen te maken.

2.4.1 ALGOL 60

- De geproduceerde objectcode is bijzonder langzaam en soms foutief.
- Vreemde foutmeldingen (b.v. "arithmetic overflow" als in feite een unsatisfied external reference aktueel wordt).
- Vertalen van grote programma's geeft soms problemen (in de objectcode wordt in plaats van een relatief adres een absoluut adres gegenereerd; compiler overflow?)
- Meerdere losse procedures in één aanroep vertalen geeft dikwijls de foutmelding: "JOB HUNG IN AUTO RECALL", waar verder niets meer mee te doen is.
- Service is slecht en herstel van gemelde fouten duurt soms zeer lang (tot 2 jaar).
- Het linken van losse procedures via codenummer is zeer onhandig. In ALGOL 4 is dit verbeterd, maar ALGOL 4 voldoet nog steeds niet aan de specificaties.
- Er is geen (met SORT/MERGE vergelijkbaar) sorteersysteem beschikbaar in ALGOL 60.
- Het SCOPE systeem is nauwelijks bereikbaar vanuit ALGOL 60.
- Er is geen betrouwbaar ALGOL 60 plot-systeem.

2.4.2 Andere talen

SIMULA:

- compiler is zeer onbetrouwbaar en raakt vaak in een loop;
- geen service.

LISP:

- systeem is niet stabiel en verandert steeds, het is nooit duidelijk welke versie men gebruikt;
- geen onderhoud.

COBOL:

- de "subcompile capability" staat het gebruik van ALGOL-60 procedures niet toe;
- het is niet mogelijk COBOL data bestanden direkt als invoer voor programma's in andere talen te gebruiken, terwijl er geen konversieprogramma's beschikbaar zijn.

PL/I

- objectcode zeer langzaam en soms fout, b.v. bij afrondig in arithmetische expressies.

2.5 MC-balie

- De procedure voor het melden van hardware storingen bij de balie is onduidelijk of onvoldoende; hierdoor ontstaan vaak onnodige vertragingen bij storingen.
- Door van de balie een "open shop" te maken kan het wachten op printer output beperkt worden.
- Als de printer defekt is, raakt output die naar de VU of MI "gedivert" is, vaak zoek. Bovendien duurt in dit geval bezorging bij MC-balie erg lang en wordt de output met een andere characterset afgedrukt.
- Er is geen kaart- of bandponser, waardoor men lang op dit soort uitvoer moet wachten.

2.6 *Voorlichting en klachten*

- De voorlichting is welwillend maar vaak (ALGOL) onkundig.
- Beantwoording van klachten duurt lang en de antwoorden zijn soms onbevredigend ("U heeft gelijk") of onjuist.

3. SAMENVATTING

De klachten die in het vorige hoofdstuk besproken zijn, kunnen kort samengevat worden in de volgende punten:

- Het SCOPE systeem + CYBER installatie zijn overbelast.
- De CYBER is sterk verouderd en heeft vele ongewenste eigenschappen, en mist vele wèl gewenste eigenschappen.
- Het SCOPE + INTERCOM systeem werkt, nog afgezien van de overbelasting, niet bevredigend.
- De kwaliteit en hoeveelheid van de beschikbare software (compilers, utilities) laat te wensen over.

4. KONKLUSIE EN ADVIES

Het is van groot belang dat de gesignaleerde problemen op zo kort mogelijke termijn opgelost worden en SARA zijn dienstverlenende taak tot tevredenheid van alle gebruikers kan uitvoeren. Om een bijdrage te leveren aan de discussie over de manier waarop dit doel bereikt kan worden, volgen nu enkele suggesties voor een door SARA te realiseren uitbreiding van apparatuur.

Aangezien het huidige CYBER systeem overbelast is, dient op korte termijn een *tweede* computer aangeschaft te worden.

Gezien de slechte ervaringen die in de afgelopen twee jaar opgedaan zijn met hardware en software van CONTROL DATA (CDC), ligt het weinig voor de hand om voor de gewenste uitbreiding opnieuw gebruik te maken van apparatuur van deze fabrikant. Bovendien is de machine die CDC te bieden zou hebben (een model uit de CYBER-170 serie; ROSNER [10]) *bitsgewijs* compatibel met de CYBER-73. Dit betekent dat alle ongewenste eigenschappen van de huidige CYBER-73 behouden blijven!

Een ander probleem bij de aanschaf van een CYBER-170 betreft het operating system. Door CDC wordt het NOS systeem (Network Operating System) geboden, waarvan de definitieve eerste release gesteld is op begin 1976. Het huidige SCOPE systeem is na vele jaren nog steeds niet stabiel en het is zeer ongewenst dat SARA testsite zou worden voor het NOS systeem.

Er zijn verschillende argumenten denkbaar die zouden pleiten vóór de aanschaf van een tweede CONTROL DATA computer:

- a) Er wordt slechts een bescheiden beroep op SARA mankracht gedaan (beperkte hertraining enz.).
- b) De kommunikatie tussen oude en nieuwe machine is eenvoudig.
- c) Nu al aanwezige peripherals (tape en disk units) kunnen op beide machines aangesloten worden; dit zou ruimte besparend kunnen werken.

Naar onze mening kunnen deze argumenten stuk voor stuk weerlegd worden:

ad a)

Voor het onderhoud van het huidige SCOPE systeem zijn tamelijk veel systeemprogrammeurs nodig. Het is onjuist hieraan de konklusie te verbinden dat dit voor ieder groot operating systeem zou gelden. De instabiliteit van SCOPE veroorzaakt veel (onnodig) extra werk. Als men een goed en stabiel

systeem kiest voor de tweede computer, is het niet nodig dat het aantal systeem programmeurs aanzienlijk uitgebreid wordt. Hierbij komt nog dat overschakeling naar een ander operating systeem (hetzij NOS hetzij een ander systeem) toch altijd hertraining vereist.

ad b)

In de door ons voorgestelde oplossing zullen de gebruikersgroepen van oude en nieuwe machine grotendeels gescheiden zijn. Hierop wordt nog nader ingegaan. Tengevolge van deze opzet hoeft de kommunikatie tussen oude en nieuwe machine slechts beperkt te zijn: alleen file transporten moeten mogelijk zijn (desnoods via magneetband). Het is onder de gegeven omstandigheden ongewenst dat het twee-computer-systeem door de gebruiker als één computer systeem gezien wordt.

ad c)

Het valt zeker niet te ontkennen dat aanschaf van een tweede computer, van een andere fabrikant dan CDC, het nodig maakt om andere peripherals aan te schaffen. Toch is dit geen onoverkomelijk bezwaar. De huidige randapparatuur (met name disks) is overbelast en uitbreiding op dit punt is alleen maar gewenst.

De door ons voorgestelde oplossing is gebaseerd op de gedachte dat het onjuist is, een groot batchsysteem en een groot interactief systeem in één gigantisch operating systeem te verenigen. Een groot gedeelte van de huidige klachten wordt o.i. veroorzaakt door deze combinatie van taken.

De gewenste scheiding van deze taken kan op twee manieren tot stand komen. In de eerste plaats is het mogelijk de kommunikatie tussen terminals en centrale computer door lokale minicomputers te laten uitvoeren. In deze minicomputer kan een editor resideren, die de terminals bedient en zo de centrale machine ontlast. Andere voordelen zijn lokaal achtergrondgeheugen voor file opslag en mede daardoor verhoogde betrouwbaarheid. Deze oplossing heeft echter in het oog springende nadelen:

- de configuratie wijkt drastisch af van de huidige opzet van het SARA-systeem met één centrale computer; realisering op korte termijn lijkt daarom onwaarschijnlijk;
- er is geen standaard software om de miniprocessors te besturen.

Gezien deze beide bezwaren lijkt de tweede oplossing die batch en interactief systeem scheidt, op korte termijn eerder haalbaar. Deze scheiding kan namelijk ook bereikt worden door een *tweede centrale computer aan te schaffen die, zowel wat hardware als software betreft, bijzonder geschikt is voor interactie*. Men zou hierbij kunnen denken aan een PDP 10. Op deze manier ontstaat een natuurlijke werkverdeling tussen de oude en de nieuwe machine, waar batch- en time-sharinggebruikers mee gebaat zijn. Het is juist ook deze disjunctie van gebruikersgroepen die het communicatieprobleem minder dwingend maakt.

Uiteraard worden door deze oplossing niet alle moeilijkheden met de batchverwerking op de huidige CYBER opgeheven. Het kan echter wel verwacht worden dat door de verminderde belasting en een vermindering van het aantal INTERCOM gebruikers de prestaties van het SCOPE systeem als geheel zullen verbeteren.

Tenslotte zal de systeemsoftware van de nieuw aan te schaffen computer een welkome aanvulling betekenen van de totaal beschikbare software.

LITERATUUR

- [1] TANENBAUM, A.S., *A critique on the Cyber*, Vrije Universiteit, Amsterdam, 1975.
- [2] WIRTH, N., *On "PASCAL", code generation, and the CDC 6000 computer*, Stanford University, 1972.
- [3] SHAPIRO, R.M. & SAINT, H., *A new approach to optimization of sequence decisions*, in (Halpern & Shaw, ed.) Annual review in automatic programming, Vol. 6 (1971) 257-288.
- [4] BRAWN, B.S. & GUSTAVSON, F.G., *Program behaviour in a paging environment*, in AFIPS FJCC 33(1968) 1019-1032.
- [5] BOBROW, D.G. & MURPHY, D.L., *A note on the efficiency of a LISP computation in a paged machine*, CACM, 11 (1968) 8 558, 560.
- [6] FENICHEL, R.R. & YOCHELSON, J.C., *A LISP garbage-collector for virtual-memory computer systems*, CACM 12 (1969) 11 611-612.
- [7] RITCHIE, D.M. & THOMPSON, K.L., *QED reference manual*, Bell telephone laboratories, inc., Murray Hill, 1972.
- [8] DIGITAL EQUIPMENT, *DEC system 10 users handbook*, DEC Maynard, Mass. 1972.
- [9] CONTROL DATA CORP., VIM newsletter, 114 (january 1975) 36-40.
- [10] ROSNER, D., *Die Control Data CYBER 170 -1. teil: hardware, -2. teil: software*, Elektronische Rechenanlagen 16 (1974) 4 158-163 en 16 (1974) 6 240-244.