

Labview

Un système d'acquisition prêt-à-porter

NOTES D'UN UTILISATEUR DÉBUTANT – AMBROGIO FASOLI – CRPP/EPFL

Le but de cet article est de fournir des informations générales et des considérations pratiques personnelles à propos du programme *Labview 2* (par National Instruments, NI) qui, associé à une ou plusieurs cartes enfichables et un MacIntosh II, constitue un système d'acquisition de données utilisable dans différentes applications.

INTRODUCTION

Un nombre important de laboratoires scientifiques de petite et moyenne taille se trouvent confrontés aujourd'hui au problème de rénover, ou tout simplement installer un système d'acquisition de données sur ordinateurs. La demande de résultats de plus en plus complets et détaillés d'une part, et le développement des techniques et outils de mesures, de plus en plus rapides et précis, d'autre part, rendent nécessaire une intégration automatisée des données sur ordinateur. Par ailleurs, les performances des PC (IBM ou MacIntosh) et en particulier des applications mathématiques et graphiques, ont été tellement améliorées que dans la plupart des cas l'analyse et la présentation des données (brutes et traitées) peuvent y être conduites entièrement.

Dans la pratique, le problème fondamental ne réside pas dans le

hardware pour la conversion analogique-numérique, pour laquelle il y a, dans certaines limites (pour les cartes enfichables NI, par exemple, de 20 bits/1 Hz à 8 bits/100 MHz de résolution/taux d'échantillonnage) une réponse adéquate pour presque tous les cas et toutes les bourses, mais plutôt dans l'intégration des données dans un environnement d'analyse.

La grande difficulté (certains parlaient de *cauchemar*) des expérimentateurs confrontés à cette question était traditionnellement la programmation de l'interface entre les instruments de mesure et les logiciels de traitement de données. Souvent l'opérateur devait s'instruire pendant longtemps dans le domaine avant de pouvoir entamer la programmation du système d'acquisition. Cela comportait des longs temps morts vis-à-vis de l'expérience et, quelquefois, une spécialisation telle que le système subissait une transition de la catégorie des *outils* à celle des *sujets d'étude*.

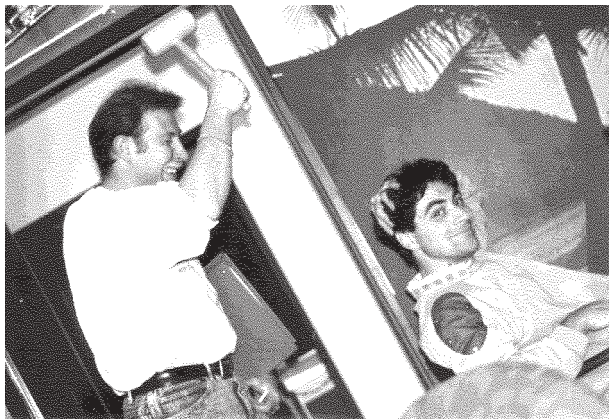
C'est surtout à ce niveau que le soft Labview, apparu récemment dans une version perfectionnée, introduit une véritable révolution. Il s'agit d'un logiciel intégré qui pilote les cartes *plug-in* d'acquisition directe des données ou de dialogue (via GPIB) avec les différents appareils de mesure et qui permet le transfert, le stockage et un traitement mathématique et graphique

Dans ce numéro

Labview – un système d'acquisition prêt à porter AMBROGIO FASOLI	1
SIC-INFO	2
Le nouveau standard Fortran 90 STEVE WILLIAMS	3
Vous avez l'heure ? CLAUDE LECOMMANDEUR	7
Conférence ERCOFTAC au CNRS – L'Europe face au défi des machines Teraflops GABRIEL CLERC, MARIE-CHRISTINE SAWLEY ET JAN-BERNHARD VOS	14
Intégration du serveur de fichiers Nestor à l'environnement de travail du Cray2 JEAN-MICHEL CHENAIS	16
Les saisons de 1991 du projet SUSP – seconde partie – Le service Pascal est pour bientôt MARIE-CHRISTINE SAWLEY	18
Matlab V4.0 JEAN-JACQUES DUMONT	20
Journée d'études sur les réseaux de neurones A3E2PL	21
Connaissez-vous Speedup ? MARIE-CHRISTINE SAWLEY	21
Formation JOSIANE SCALFO	22
Calendrier	24

sic – info

DE VIEILLES NOUVELLES TÊTES



Ion a la tête dure mais il y a de meilleurs moyens...

Depuis le 1er janvier, les utilisateurs de matériel Hewlett-Packard et les nostalgiques d'Apollo savent que Claudé Lecommandeur n'est plus le responsable du support de cette ligne de produit. Il a eu envie d'aller reprendre certaines tâches (urgentes !) à la section logiciel du SIC.

Ion Cionca a quitté ses fonctions à l'IMECO (DME) pour venir nous et vous aider. Et auparavant, il était parti de Roumanie afin de supporter (dans plusieurs sens) l'EPFL !

Fred-Ami Rougemont, SIC-Informatique individuelle

LES NOUVEAUX FLEURONS DE LA SECTION LOGICIELS

La section logiciels a la douleur de vous faire part de l'arrivée en ses rangs, à dater du 1er avril (sans blague !), de deux nouveaux fleurons:

Didier Wagenknecht, 26 ans, probablement toutes ses dents, est chargé des problèmes de sécurisation informatique: définition et support de configurations sûres pour les systèmes, élaboration de concepts de sécurité en vue d'éviter la propagation de virus et mise en place de moyens de défense contre les intrusions éventuelles de personnes non autorisées sur les réseaux de l'Ecole.

De formation purement genevoise et long comme un jour sans pain, Didier est un garçon sans histoire(s): pas d'activité subversive, et aucun prix Nobel jusqu'à présent. De nombreuses qualités inavouables et un seul défaut connu: un goût trop prononcé pour les motos de cross

(j'en connais d'autres à qui cela a porté malheur !) Pourtant, c'est une figure bien connue déjà à l'EPFL, puisqu'il apporta une aide précieuse au Service Informatique du DI alors même qu'il n'avait pas terminé ses études dans ce Département. C'est maintenant chose faite, et il est prêt à affronter les lourdes responsabilités susmentionnées.

Pierre Collinet, 29 ans et d'origine belge (mais il se soigne), est également un pur produit du Département d'Informatique de l'EPFL. Après une participation post-diplôme au projet «Commande numérique de machines» au Laboratoire d'Informatique Technique, l'envie lui prit d'explorer d'autres voies: développement d'un système de gestion de fréquences radio pour un centre de recherche technique à Bruxelles, applications comptables et financières à Fribourg, puis contrôle de moteurs d'A340 pour l'Aérospatiale. Sans parler de son intérêt pour les langues, pour les problèmes de société et de communication (volontariat au Centre Vaudois d'Aide à la Jeunesse, au Children's Hospital de Pittsburgh).

Nous ne connaissons pas encore ses défauts, mais il faut lui reconnaître au moins une qualité: le courage. Celui d'avoir accepté au SIC la responsabilité d'applications et systèmes distribués, avec comme première priorité la mise en place d'un nouveau système de messageries, plus homogène et convivial que celui que nous connaissons actuellement. Le courage ne suffisant pas toujours, nous lui souhaiterons également bonne chance...

Jean-Jacques Dumont, SIC-Logiciels

Flash Informatique

Les articles de ce journal ne reflètent que l'opinion de leurs auteurs. Toute reproduction, même partielle, n'est autorisée qu'avec l'accord de la rédaction et des auteurs.

Rédacteur en chef : Marie-Christine Sawley

Comité de rédaction : Jean-Daniel Bonjour,
Jean-Michel Chenais, Milan Crvcenin,
Jacqueline Dousson, Pierre-André
Haldy, Véronique Jost, Pierre-Jean
Paris, Nathalie Rabier, François Roulet
& Jacques Virchaux

Composition : Appoline Raposo de Barbosa

Impression :



Tirage : 4000 exemplaires

Adresse : SIC-SA EPFL 1015 – Lausanne

☎ 021/693 22 42 & 22 47

Prochaine parution : 26 mai 1992

Délai de rédaction : 7 mai – 12h00

Le nouveau standard Fortran 90

PAR STEVE WILLIAMS, SIC-ASSISTANCE

INTRODUCTION

Le langage Fortran est un langage d'utilisation facile qui convient aux applications contenant beaucoup de calcul numérique. C'est la raison pour laquelle Fortran 77 est très populaire dans le monde scientifique notamment dans le cadre de nombreuses applications et librairies, utilisées tous les jours. Cette situation a convenu aux utilisateurs pendant longtemps.

Pendant la période d'application du code Fortran 77, un certain nombre de nouveaux langages de programmation ont été développés et certains d'entre eux présentaient des fonctionnalités plus performantes que celles de Fortran 77. Afin de répondre aux besoins immédiats de leurs utilisateurs, des constructeurs ont procédé à l'extension des possibilités de leur propre compilateur. Cette politique signifiait que la portabilité entre les différentes machines était compromise.

Fortran 90 a été conçu dans le but de re-standardiser le langage et parallèlement d'ajouter les fonctionnalités nécessaires à la programmation des nouvelles machines vectorielles et parallèles.

Cet article expose les particularités principales du standard Fortran 90 dans le but de familiariser les utilisateurs de l'ancien modèle aux spécificités de Fortran 90. Pour les personnes intéressées, un compilateur Fortran 90 de NAG, en test sur la machine sicsun, sera mis à leur disposition autour de la période de Pâques et pour une durée d'un mois.

HISTOIRE

Pour rappel historique, il faut préciser que le standard Fortran 77 est utilisé depuis 1978. Ce standard n'a pas été révisé pendant quinze ans, malgré les tentatives d'ISO et ANSI. En avril 1991, ISO a accepté la proposition du comité WG5 consistant à définir un nouveau standard (Fortran 90). Aux Etats-Unis, il semble que ANSI va aussi l'accepter dans un avenir proche.

Le standard répondait à deux buts contradictoires: être compatible avec Fortran 77, et devenir un langage *moderne* en permettant les extensions nécessaires. Le premier but n'a jamais été contesté; l'existence d'un investissement vaste en code Fortran 77 excluait l'idée de tout refaire.

Il a été constaté depuis longtemps que Fortran 77 ne possédait pas certaines fonctionnalités présentes dans d'autres langages. Parmi les nouvelles caractéristiques les plus importantes de Fortran 90 on trouve:

- a) les opérations concises sur les tableaux,
- b) des types de données structurées,

- c) la modularisation,
- d) l'allocation dynamique de la mémoire,
- e) davantage de procédures intrinsèques.

Ces nouvelles fonctionnalités ont pour but de promouvoir la portabilité, la fiabilité, la facilité d'entretien, et une efficacité de l'exécution sur tout un ensemble de plates-formes.

NOUVELLES FONCTIONNALITÉS

Tableaux

Les changements décrits dans cette section montrent les grands progrès faits depuis le standard précédent. Ce nouveau Fortran permet de formuler des expressions et des assignations de tableaux telles que:

$$A = B + 2.0 * C * \text{SIN}(D),$$

où A, B, C et D sont des tableaux. Ici, les opérations se font élément par élément; c'est-à-dire que l'opération intrinsèque SIN est appliquée sur chaque élément de D, l'opération intrinsèque de multiplication de tableaux est exécutée entre ce résultat et le tableau C, chaque élément de ce nouveau tableau est multiplié par le scalaire 2.0, et ajouté à l'élément correspondant dans le tableau B. Pour finir, le résultat est assigné au tableau A. Bien sûr, il faudrait que chaque résultat intermédiaire ait les bonnes dimensions. Ce type d'expression est non seulement plus concis et lisible, mais aussi, dans le cas de tableaux multidimensionnels, il se prête mieux à une exploitation vectorielle et parallèle par un compilateur.

Des sous-tableaux, ou *sections*, peuvent être utilisés de la même manière que les tableaux. Par exemple, le tableau A(10,15) peut être utilisé aussi bien qu'un tableau A(:,4), tableau unidimensionnel de taille 10 dans la colonne 4 du tableau A. Autres exemples: A(1:3,5:8) est le sous-tableau compris entre les colonnes 5 et 8 et dans les rangs 1 à 3; A((/1,7,3,2/),:) est le sous-tableau consistant en les rangs 1, 7, 3, et 2, selon cet ordre; et A(5,15:1:-1) est le rang 5 dans l'ordre inverse des colonnes.

Pour effectuer une opération sur un tableau, mais seulement pour certains éléments, on utilise la fonction WHERE. Exemple:

$$\text{WHERE} (A > 0) A = \text{SQRT}(A)$$

prend la racine de chaque élément positif du tableau A, et ne change pas les autres éléments.

Le fait que les tableaux Fortran 77 sont statiques est très limitatif. Fortran 90 résout ce problème en introduisant des tableaux 'automatic', 'allocatable' et 'assumed-

shape'. La déclaration d'une sous-routine peut se faire de la manière suivante:

```
SUBROUTINE SWAP(A,B)
  REAL, DIMENSION(:) :: A, B
  REAL, DIMENSION( SIZE(A) ) :: WORK
```

Ici, les tableaux A et B sont 'assumed-shaped' (leurs tailles sont issues des tableaux actuels lorsque la sous-routine SWAP est appelée), et le tableau WORK est un tableau 'automatic' (créé en entrant dans SWAP et détruit en en sortant). Ces possibilités rendent la tâche du programmeur plus facile dans les domaines des appels aux bibliothèques et la gestion de la mémoire.

Un tableau peut être déclaré 'allocatable': sa dimension est définie, sa taille et sa durée de vie sont sous contrôle total du programmeur avec les opérations ALLOCATE et DEALLOCATE. Un exemple d'une telle déclaration est:

```
INTEGER, ALLOCATABLE :: A(:, :)
```

Ici, le tableau A est de dimension deux, mais sa taille est indéfinie. Plus tard dans le programme on peut allouer et désallouer de la mémoire de manière dynamique, comme par exemple:

```
ALLOCATE( A(N,M) ) et
DEALLOCATE( A )
```

Types de Données Structurées

Fortran 90 permet de regrouper différentes données dans une structure. Par exemple, si on doit manipuler les données sur un groupe de personnes, il serait logique de créer une structure par personne telle que:

```
TYPE PERSON
  CHARACTER (LEN = 10) NAME
  REAL                AGE
  INTEGER              ID
END TYPE PERSON
```

Par la suite on peut définir des variables du type PERSON, assigner des valeurs à chaque champ de la structure, et manipuler ces données en bloc. Voici quelques possibilités de déclarations permettant d'utiliser la structure PERSON:

```
TYPE ( PERSON ) SMITH, JONES
SMITH%NAME = 'Bob'
SMITH%AGE = 23.5
SMITH%ID = 4325
JONES = PERSON( 'Doug', 24.0, 2364 )
```

La première ligne déclare deux variables du type PERSON. Les trois lignes suivantes assignent des valeurs aux champs du variable SMITH. On peut sélectionner un composant d'une structure avec le symbole %. La dernière ligne montre l'utilisation d'un constructeur; tous les champs de la variable JONES sont assignés d'un seul

coup.

Pointeurs

Un objet peut être déclaré avec l'attribut POINTER, indiquant que l'objet n'a pas de mémoire allouée avant que soit faite une allocation explicite avec l'opération ALLOCATE ou que soit associé l'objet avec un autre objet existant. Cette association est faite par la déclaration suivante:

```
POINTER => TARGET
```

Celle-ci associe le pointeur POINTER à l'objet existant TARGET.

Les pointeurs permettent la construction de 'linked lists' et d'arbres. Par exemple, pour construire un arbre binaire, un type NODE doit être défini de la manière suivante:

```
TYPE NODE
  INTEGER :: VALUE
  TYPE(NODE), POINTER :: LEFT, RIGHT
END TYPE NODE
```

Chaque objet de type NODE contient un entier VALUE et deux pointeurs, LEFT et RIGHT, vers deux autres objets de type NODE. Ensuite, peut être créé et manipulé un arbre d'objets NODE:

```
TYPE(NODE), POINTER :: TREE
NULLIFY(TREE)
IF (.NOT.ASSOCIATED(TREE)) THEN
  ALLOCATE(TREE)
  TREE%VALUE = X
  NULLIFY(TREE%LEFT)
  NULLIFY(TREE%RIGHT)
END IF
```

La première ligne déclare la racine de l'arbre. L'opération intrinsèque NULLIFY dissocie TREE de tout objet (ce n'est pas nécessaire ici, mais c'est plus propre). L'opération ASSOCIATED(X) donne le résultat .TRUE., si X est associé à un objet existant. ALLOCATE(TREE) alloue un objet de type NODE, et les trois lignes suivantes remplissent les champs de ce nouvel objet.

Les pointeurs peuvent aussi être associés à des tableaux. Exemple de l'usage des pointeurs:

```
REAL, POINTER, DIMENSION(:, : ) :: APTR
ALLOCATE( APTR(10, 12) )
```

Dans le cas présent, le pointeur APTR fait référence à un tableau bi-dimensionnel. Il faut noter qu'on ne spécifie la taille du tableau que lors d'une déclaration ALLOCATE, dans ce cas APTR fait référence à un tableau de 10 par 12.

Structures de contrôle

Fortran 90 offre un autre moyen de sélectionner une des options, similaire à la structure IF, mais plus efficace. L'exemple permet de comprendre l'usage de la structure CASE:




```

SELECT CASE (CHAR) ! CHAR of type CHARACTER
CASE ('C', 'D')
  DATA = 1
CASE ('I': 'N')
  DATA = 2
CASE DEFAULT
  DATA = 3
END SELECT

```

Si la variable CHAR vaut 'C' ou 'D', DATA est mis à 1, si elle est comprise entre 'I' et 'N', DATA est mis à 2, et dans tous les autres cas, DATA est mis à 3.

Il existe aussi un nouveau format possible pour les boucles DO qui n'utilisent pas de labels:

```

OUTER: DO ! Boucle nommée OUTER
DO I=1,N
  IF (...) EXIT OUTER ! Sortir de la boucle OUTER
IF (...) CYCLE ! Aller à la fin de la boucle
END DO
END DO OUTER

```

Structure de programme

Déclaration PROGRAM, FUNCTION, SUBROUTINE ou MODULE		
Déclarations USE		
Déclarations FORMAT	Déclaration IMPLICIT NONE	
	Déclarations PARAMETER	Déclarations IMPLICIT
	Déclarations PARAMETER et DATA	Déclarations de types, blocs d'interface
	Déclarations exécutables	
Déclaration CONTAINS		
Sous-programmes		
Déclaration END		

figure 1 : ordre des déclarations

Un programme Fortran 90 consiste en un programme principal, des sous-programmes externes et des modules. L'ordre des déclarations est résumé dans la figure 1. Les modules constituent la principale nouveauté par rapport au Fortran 77. Un module est un ensemble de données, définitions de types, et définitions de procédures. L'utilité d'une telle structure est de permettre d'encapsuler la définition d'un type avec toutes les opérations devant manipuler les variables de ce même type. Par exemple, un module permettant de manipuler un type *intervalle* peut être écrit de la manière suivante:

```

MODULE INTERVAL_ARITHMETIC
  TYPE INTERVAL
    REAL LOWER, UPPER
  END TYPE INTERVAL
CONTAINS
  FUNCTION ADD_INTERVALS (A, B)

```

```

  FUNCTION SUBTRACT_INTERVALS (A, B)
  etc.
END MODULE INTERVAL_ARITHMETIC

```

Pour accéder aux type et procédures de ce module ailleurs dans le programme il faut ajouter la déclaration USE INTERVAL_ARITHMETIC. L'option PRIVATE peut être ajoutée aux types et procédures d'un module afin de les cacher de l'extérieur. Les bibliothèques de Fortran 90 vont sans doute être composées de modules et non pas de procédures comme en Fortran 77.

Procédures

Une procédure peut être appelée de façon récursive si sa définition inclut RECURSIVE. Quant aux paramètres, ils peuvent être déclarés OPTIONAL et IN, OUT ou INOUT. Dans le premier cas, il serait possible d'omettre certains arguments dans un appel si la procédure n'en a pas besoin. Une fonction intrinsèque PRESENT peut déterminer si un paramètre est passé à la procédure ou non.

Dans les déclarations des paramètres, il est conseillé de spécifier la direction de transfert de données. Exemple:

```

SUBROUTINE SHUFFLE (NCARDS, CARDS)
  INTEGER, INTENT(IN) :: NCARDS
  INTEGER, INTENT(INOUT), DIMENSION(NCARDS):: CARDS

```

Dans cet exemple, on peut voir que l'entier NCARDS est passé à SHUFFLE en mode read-only (IN), tandis que le tableau CARDS est passé en mode read-write (INOUT). Cette pratique fournit non seulement une bonne documentation, mais aussi elle peut aider le compilateur à contrôler le bon passage de paramètres.

Fortran 90 permet de définir plusieurs procédures sous le même nom, si ces différentes procédures se distinguent par le nombre ou le type d'arguments. Ce concept s'appelle 'overloading'. Dans le cas d'une structure de données, on peut faire un 'overload' de l'opération élémentaire «+» de la manière suivante:

```

MODULE POINT
  TYPE POINT_TYPE
    REAL X, Y
  END TYPE POINT_TYPE
  INTERFACE OPERATOR (+)
    MODULE PROCEDURE ADD_POINT
  END INTERFACE
CONTAINS
  FUNCTION ADD_POINT (A, B)
    TYPE (POINT_TYPE) ADD_POINT, A, B
    ADD_POINT%X = A%X + B%X
    ADD_POINT%Y = A%Y + B%Y
  END FUNCTION ADD_POINT
END MODULE

```

Dans un programme utilisant ce module, il serait possible de définir deux variables A et B de type POINT_TYPE et faire l'opération A=A+B. Le compilateur saurait quelle procédure utiliser en fonction du type d'arguments. ➡

Procédures Intrinsèques

Fortran 90 fournit plusieurs procédures intrinsèques nouvelles. Dans le cadre de cet article, il n'est pas possible de toutes les présenter. Parmi les procédures les plus intéressantes, sont à noter les routines permettant de manipuler des entiers bit par bit (par exemple, IAND, IOR, IEOR, NOT, et ISHFT pour le décalage). DOT_PRODUCT et MATMUL sont deux nouvelles procédures d'opérations sur tableaux. Il existe aussi des fonctions réduisant des tableaux: MAXVAL, MINVAL, SUM, PRODUCT... Ces fonctions peuvent utiliser l'argument optionnel MASK. Exemple:

```
X = SUM( A, MASK = A>0 )
```

Ici on additionne tous les éléments positifs du tableau A. Enfin, les procédures CALL DATE_AND_TIME et CALL SYSTEM_CLOCK sont utilisées pour accéder à l'horloge temps-réel.

Autres nouveautés

Les principales autres nouvelles fonctionnalités sont les suivantes:

- absence de contraintes concernant la position des déclarations (autrefois elles devaient commencer dans la colonne 8 et finir avant la colonne 72),
- les noms des variables, procédures, etc., peuvent avoir 31 caractères (6 comme dans l'ancien standard),
- le '!' introduit dans une ligne permet de considérer tous les caractères à droite comme commentaires,
- les constantes binaires, octales, et hexadécimales sont permises,
- des types et/ou des constantes peuvent être définis avec une précision et une gamme d'exposants différents des types intrinsèques,
- il existe des expressions logiques telles que X.AND.Y,
- les lettres .LE., .GT., .LE., .GE., .EQ. et .NE. peuvent être remplacées par les symboles respectifs <, >, <=, >=, == et /=,
- l'expression IMPLICIT NONE signifie que le type de toutes les variables doit être déclaré explicitement (il est conseillé d'utiliser cette déclaration),
- il existe une option nouvelle permettant d'éviter la lecture complète des enregistrements et de connaître à l'avance leur longueur.

COMPATIBILITÉ AVEC FORTRAN 77

Pour la première définition de Fortran 90, tous les programmes écrits en Fortran 77 seront acceptés par un compilateur Fortran 90 si le programme est conforme au standard Fortran 77 d'ANSI. Néanmoins, il est déconseillé d'utiliser certaines fonctionnalités de Fortran 77. Ces fonctionnalités sont classées en deux groupes: obsolète (obsolescent features) et déconseillé (deprecated features).

Les fonctionnalités du groupe obsolète seront peut-être mieux acceptées dans la prochaine version de Fortran

90. Ce groupe inclut:

- IF arithmétique, p.ex., IF (P-Q) 10, 20, 30
- un index non-entier d'une boucle DO, p.ex., DO 10 A= 1, 15.7, 2.1
- terminaison d'une boucle DO sur une ligne autre que CONTINUE ou END DO
- terminaison de plusieurs boucles DO sur la même ligne
- branchement vers une ligne END IF depuis l'extérieur du bloc IF
- PAUSE, et
- ASSIGN et les GO TO assignés, p.ex., ASSIGN 5 TO JUMP, GO TO JUMP (4,5,6)

Les fonctionnalités du groupe déconseillé sont redondantes dans Fortran 90, et pourraient passer au groupe obsolète dans la prochaine révision. Ce groupe inclut:

- la déclaration EQUIVALENCE qui peut être remplacée par des pointeurs,
- les blocs COMMON à remplacer par des modules,
- la déclaration INCLUDE à remplacer par la déclaration USE,
- la boucle DO WHILE à remplacer par une boucle normale avec utilisation de la déclaration EXIT à l'intérieur de la boucle,
- le format source 'fixed',
- la déclaration DOUBLE PRECISION à remplacer par la déclaration REAL selon la précision désirée,
- le GO TO calculé, à remplacer par la structure CASE.

COMPILATEURS

NAG (Numerical Algorithms Group) a été le premier fabricant à offrir un compilateur Fortran 90. Il est compatible avec le standard ISO. Ce compilateur peut être considéré comme traducteur puisqu'il génère du code C et invoque le compilateur C natif pour produire du code exécutable. Il existe sur un nombre limité de machines dans l'immédiat, à savoir les stations Sun, HP/Apollo et IBM. Nous avons la possibilité d'avoir ce compilateur pour une période de test d'un mois aux alentours de Pâques. Cependant cette collection va s'étendre dans la mesure où le produit dépend uniquement des différents compilateurs C. Il est difficile de donner plus de précision quant à leur disponibilité. Dans un premier temps il pourra être introduit sur la machine *sicsun* et si le nombre de personnes intéressées est conséquent, il sera possible d'en commander de nouveaux pour d'autres machines. Au sein de l'école, le plus grand intérêt a été montré pour Cray, Silicon Graphics et HP. La plupart des constructeurs vont créer leur propre compilateur Fortran 90, et dans ce cas il serait probablement plus intéressant d'utiliser un compilateur natif. Malheureusement ces compilateurs n'apparaîtront pas avant fin 1992.

Les personnes qui désirent essayer le compilateur NAG doivent s'adresser à l'auteur. Il faudrait vous assurer que dans un premier temps vos anciens programmes soient conformes au standard Fortran 77. Sur Sun, il faut compiler avec l'option «-ansi», sur le Cray avec

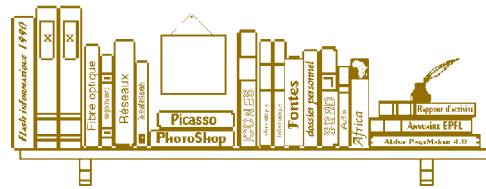
l'option «-en» et sur le VAX avec l'option «/STANDARD».

CONCLUSIONS

Cet article a tenté d'informer le public des avantages et difficultés que représente ce nouveau langage de programmation. Il est entendu que cet article ne fait état que de généralités. Pour plus de détails, consulter le livre de Metcalf et Reid dans la bibliographie ci-après.

Le standard Fortran 90 ne devrait pas effrayer les scientifiques qui ont beaucoup investi dans le code Fortran 77. Dans la plupart des cas, seulement des modifications mineures seront nécessaires. Concernant le nouveau code, il faudrait le rendre conforme au standard ANSI Fortran 77. En outre, l'utilisation des fonctionnalités obsolètes et déconseillées n'est pas conseillée.

Il est prévu que les révisions futures de Fortran 90 vont faire apparaître plus de fonctionnalités pour la programmation parallèle et pour le traitement d'exceptions. Cependant, il n'est pas certain que les constructeurs de systèmes massivement parallèles utilisent Fortran 90 pour leurs machines plutôt qu'un autre langage à définir.



- «Fortran 90 explained», par Mike Metcalf et John Reid, Oxford University Press, ISBN 0-19-853772-7 - *Une description informelle de tout le langage (peu d'exemples)*
- «Programmer's Guide to Fortran 90», par W.S. Brainerd, C.H. Goldberg et J.C. Adams, McGraw-Hill, ISBN 0-07-000248-7 - *Une description des principales nouvelles fonctionnalités*,
- Le standard ISO peut être commandé chez: ISO, Publications Department, Case Postale 56, 1211 Genève 20
- «Fortran 90, the new Fortran standard», par John Reid, .EXE Magazine, Vol. 6, Issue 3, Août 1991 - *Une description condensée du même style que cet article*

Pour tout renseignement adressez-vous à Steve Williams, 693 22 53, ou par courrier électronique williams@sic.epfl.ch ■

Vous avez l'heure ?

PAR CLAUDE LECOMMANDEUR, SIC-LOGICIELS

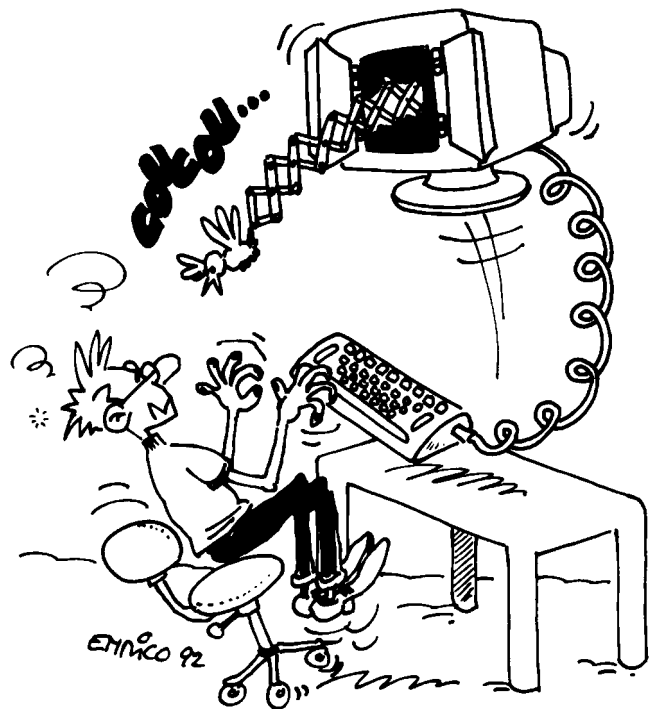
LA SYNCHRONISATION DES ORDINATEURS

Les ordinateurs sont follement précis dans leurs calculs, mais leurs horloges ne sont pas à la mesure de cette précision. Il n'est pas rare que certains micros ou stations prennent plusieurs minutes de retard ou d'avance par mois.

Ces dérives sont souvent sans importance et passent inaperçues tant que les interactions entre machines ne sont pas trop fortes. Mais dès que les couplages entre ordinateurs par le biais des réseaux deviennent plus importants, la synchronisation des horloges devient nécessaire.

Par exemple, lors de l'utilisation de la commande «make» qui tient compte des dates de modification des fichiers. Si une partie des fichiers en cause est sur une partition vue à travers NFS et d'autres fichiers sont locaux, une bonne synchronisation de l'horloge locale et de celle de la machine distante est fondamentale.

Ce couplage des ordinateurs est amené à augmenter énormément dans un futur proche avec l'avènement des outils d'informatique distribuée qui se préparent. Les systèmes de fichiers distribués (AFS/DFS), l'authentification entre ordinateurs (Kerberos) font un usage intensif du temps et rendent obligatoire une bonne synchronisation. ■



LES DIFFÉRENTS PROTOCOLES

Il existe une grande quantité de protocoles des ordinateurs. Nous citerons les 3 principaux : TIMED, NTP et DTS.

TIMED

Le plus ancien, et le plus simple. La plupart des machines UNIX le supportent. Une machine est supposée disposer d'un temps fiable, et les autres vont l'interroger quand elle veut (généralement au moment du boot) pour se synchroniser. Les commandes PC de synchronisation (setclock, ...) utilisent ce système.

NTP

Ce protocole, apparu plus récemment, est infiniment plus sophistiqué. Chaque machine cliente se synchronise sur plusieurs serveurs de temps. Elle fait des statistiques sur les temps de transit des requêtes, les dérivées éventuelles des horloges de ces serveurs, la dérive de sa propre horloge interne, etc...

A l'aide de tous ces calculs elle déduit l'heure la plus probable. La fiabilité de ce protocole est extrême, mais sa complexité en est à la mesure.

Ce protocole est actuellement le plus utilisé sur les stations de travail unix à travers le monde.

DTS

Ce protocole est l'œuvre de DEC (Digital Equipment) et a été choisi par OSF pour son environnement d'informatique distribuée DCE. Son principe est légèrement différent des précédents dans la mesure où il considère le temps non comme un nombre réel, mais comme un intervalle. Ce qui introduit une notion intrinsèque de précision.

Sinon, il a, comme NTP, la notion de serveurs qui maintiennent l'heure, et de clients qui s'y réfèrent.

L'INSTALLATION AU SIC

L'installation actuelle, utilise à la fois DTS et NTP. Une horloge radio (Precitel) est connectée sur un port série de la machine Cognac qui synchronise un serveur DTS. Malheureusement la version DTS dont nous disposons actuellement ne tourne que sur matériel DEC et, de plus, ne communique que sur le protocole OSI. Il y a donc aussi un serveur NTP qui se synchronise sur l'horloge locale de Cognac ainsi que sur 2 autres serveurs de temps à l'extérieur de l'EPFL. C'est ce serveur NTP qui distribue le temps dans l'école. Pour des raisons de redondance et de fiabilité, il y a 2 autres machines au SIC qui se synchronisent elles-mêmes sur Cognac et les 2 machines externes qui distribuent le temps.

LA DISTRIBUTION DU TEMPS A L'EPFL

Ce à quoi on voudrait arriver :

Minimiser les synchronisations à l'extérieur de l'école.

Plan proposé :

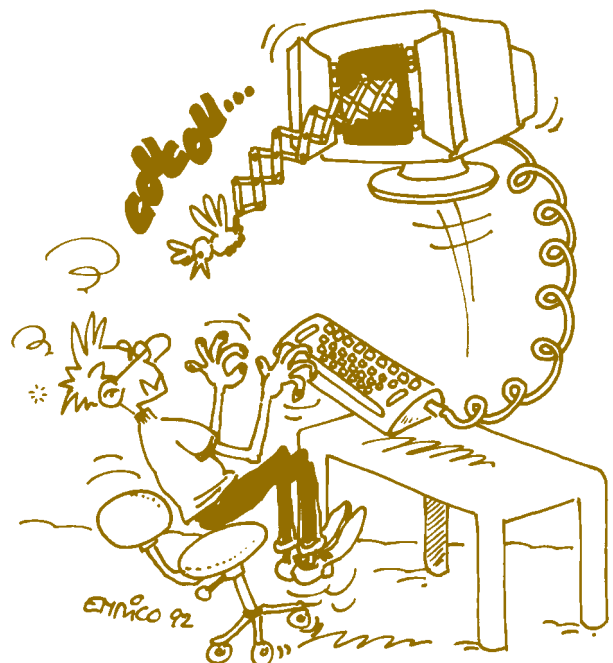
Dans le cadre du groupe COGNAC, un sous-groupe de travail a réfléchi à la question et proposé l'architecture suivante :

- Mise en place au SIC de trois serveurs qui se synchronisent sur la machine Cognac et sur 2 serveurs externes.
- Pour tous les départements intéressés : mise en place de trois serveurs locaux, se synchronisant en tant que clients sur les trois serveurs du SIC et entre eux.
- Toutes les machines de ces départements se synchronisent sur ces trois serveurs locaux.
- Si la charge pour ces machines est trop forte, ou si les problèmes réseaux sont trop fréquents, ou si une sous-entité d'un département désire une plus grande indépendance, un département peut rajouter une couche suivant le même principe de trois serveurs pour cette sous-entité.
- Le(s) responsable(s) du service informatique du département concerné choisit(ssent) ses trois serveurs.

Qui peut se synchroniser ?

Le logiciel tourne sur toutes les stations Unix (HP, DEC, SG et SUN). Il existe aussi une version MacIntosh qui permet la synchronisation au moment du boot. Il n'y a pas, à ma connaissance, un tel système pour PC, mais je peux me tromper.

Prenez le temps d'avoir le temps. ■



des données. (☛ *suite en page 9*)

L'aspect innovateur de Labview (et de ses homologues qui ont été développés en parallèle), quelque peu *révolutionnaire*, permet à l'utilisateur de n'être plus confronté à la programmation des interfaces ordinateur - carte ou instrument de mesure. En principe, tout ce qui lui est demandé est d'*organiser* le programme en fonction des nécessités particulières de ses expériences.

La question qui est posée, surtout par ceux qui sont dans le processus de choisir un système d'acquisition pour leur expérience, est donc: est-il possible, en peu de temps et avec peu de connaissances spécifiques, d'*organiser* les structures de Labview d'une façon telle que les données d'une certaine expérience de laboratoire puissent être acquises, traitées et enregistrées efficacement?

Au-delà des réponses officielles des promoteurs, l'expérience directe d'un groupe de recherche peut aider à vérifier ces possibilités. Avant d'aborder un exemple particulier d'utilisation de Labview sur Macintosh, voyons quelques caractéristiques générales du système software et quelques configurations *hardware* possibles. Evidemment, une évaluation économique d'un certain système ne peut être conduite qu'en considérant aussi la période de

(figure 2).

Le point crucial d'un tel modèle est le software qui est le garant de l'intégration des différentes options dans un seul système. Toutes les fonctions d'un système d'instruments doivent pouvoir être assurées, de l'acquisition des données proprement dite au contrôle des instruments, de l'analyse mathématique à la présentation des données.

Avec ce type de soft, il serait possible de construire son propre *instrument* satisfaisant les besoins et les contraintes d'une application particulière. Le soft deviendrait ainsi *de facto* l'instrument scientifique.

En s'appuyant sur cette idée, le logiciel *Labview* (version pour Mac, *Labwindow* est son *alter ego* sur PC-IBM) est apparu dans sa première version en 1986.

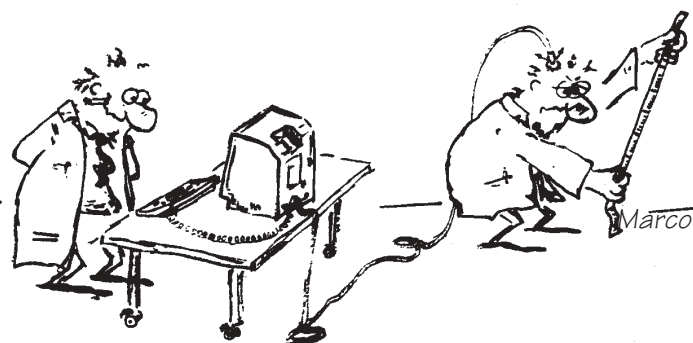
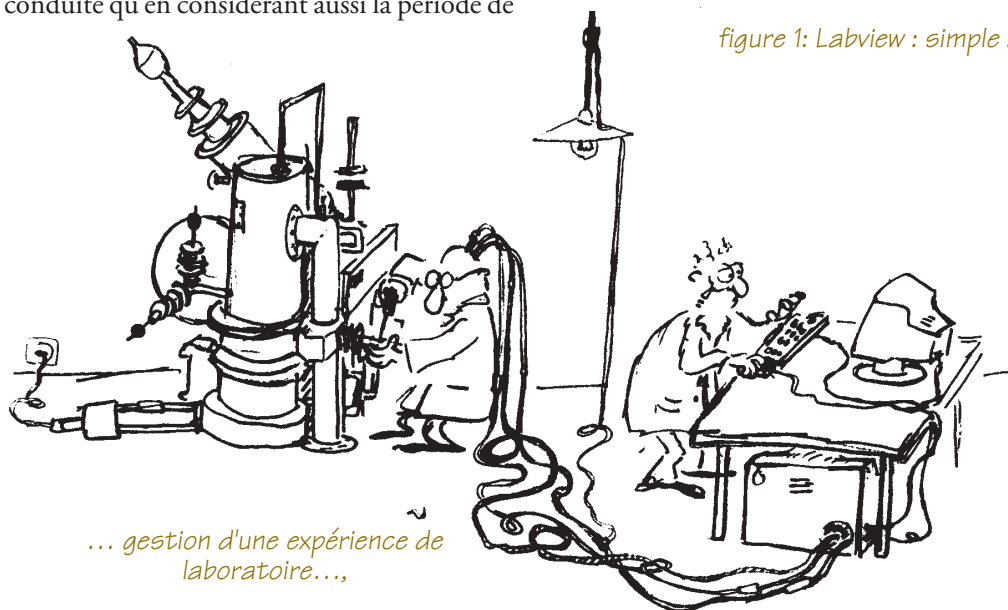


figure 1: Labview : simple acquisition de données....



... gestion d'une expérience de laboratoire....

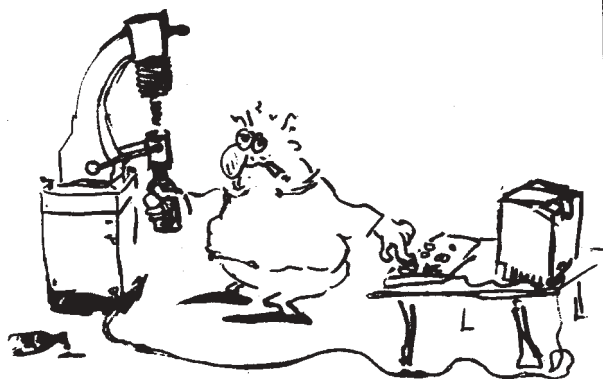
Véritable langage de programmation graphique à diagramme à blocs, Labview est caractérisé par la métaphore des *Instruments Virtuels*. Un *Instrument Virtuel* (VI) est un module de programme représenté sous forme graphique, composé par trois parties principales: le panneau de commande (ou face avant, *front panel*), le diagramme à blocs, et l'icône avec les connecteurs.

Le panneau de commande constitue en effet

temps (heures . personnes) nécessaire pour la mise en opération, y compris le temps dont l'utilisateur a besoin pour apprendre à se servir réellement du système. Cela est important, car, selon la solution adoptée, il peut s'agir d'une échelle de temps de jours, semaines, voire mois !

DESCRIPTION GÉNÉRALE DU LOGICIEL LABVIEW

Le modèle général du système d'acquisition de petite et moyenne taille des années '90 est constitué par des instruments GPIB, des cartes d'acquisition *plug-in*, et des instruments VXI et RS-232; ces modules de hardware standardisés doivent être intégrés sur des ordinateurs qui sont effectivement des standards IBM-PC ou Macintosh



... ou contrôle de processus.

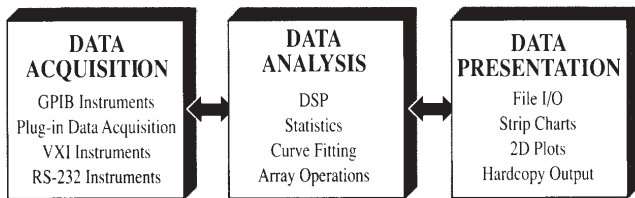


figure 2

l'interface avec l'utilisateur. Il comporte des objets de contrôle (boutons, curseurs, switches,...) et des objets d'indication (graphes, tableaux,...). La structure du panneau, lié évidemment à celle du diagramme, est créé par l'utilisateur. Elle peut donc être adaptée à l'application particulière (et à l'opérateur particulier) et en même temps être forgée de manière standard, indépendante des réels instruments de mesure utilisés. C'est à travers le panneau de commande que le contrôle d'instruments indépendants modulaires ou des cartes d'acquisition est opéré (pour un exemple voir figure 3).

Le diagramme à blocs est le «programme» qui détermine les fonctions accomplies par l'instrument virtuel. Pour le construire il suffit de connecter (à l'aide d'un outil graphique qui *pose* des fils entre les broches) les différents éléments graphiques (VI ou icônes) proposés par les menus *pop-up*, exactement comme si l'on traçait simplement un schéma de principe de la chaîne de mesure. Différentes bibliothèques d'objets et de VI sont à disposition.

Pour la première étape d'interface hardware/ordinateur une bibliothèque de *drivers* est fournie (*LabDriver*). Ces VI de base contrôlent les diverses cartes d'instrumentation (conversion analogique/numérique, fonction de timing, dialogue GPIB, entrées/sorties digitales). Dans *LabDriver* on trouve l'icône pour initialiser la carte, l'icône pour le «reset», celle pour le réglage du «trigger», celle pour la conversion A/D, etc. Et cela pour toutes les cartes existantes (chez NI) ...

Les entrées et les sorties des différents modules apparaissent dans le panneau de commande respectivement sous forme de contrôles et indicateurs.

Dans le logiciel Labview proprement dit, des bibliothèques d'objets et VI pour le traitement des données peuvent être facilement trouvées et sélectionnées. Particulièrement riche est le répertoire DSP (*Digital Signal Processing*) contenant entre autre des routines FFT et toute sorte de filtres numériques. Des calculs statistiques, des opérations matricielles et différentes sortes d'interpolations de courbes expérimentales peuvent aussi être effectuées. Toutefois, à l'heure actuelle, cela nous paraît loin d'être exhaustif. Un développement dans ce domaine est en cours pour les nouvelles versions du logiciel. Pour des applications poussées dans cette direction, il est encore nécessaire de transférer les données sur d'autres applications (par exemple Matlab).

Les mêmes considérations s'appliquent à la représentation graphique des résultats. Une lecture directe sur le

Front Panel

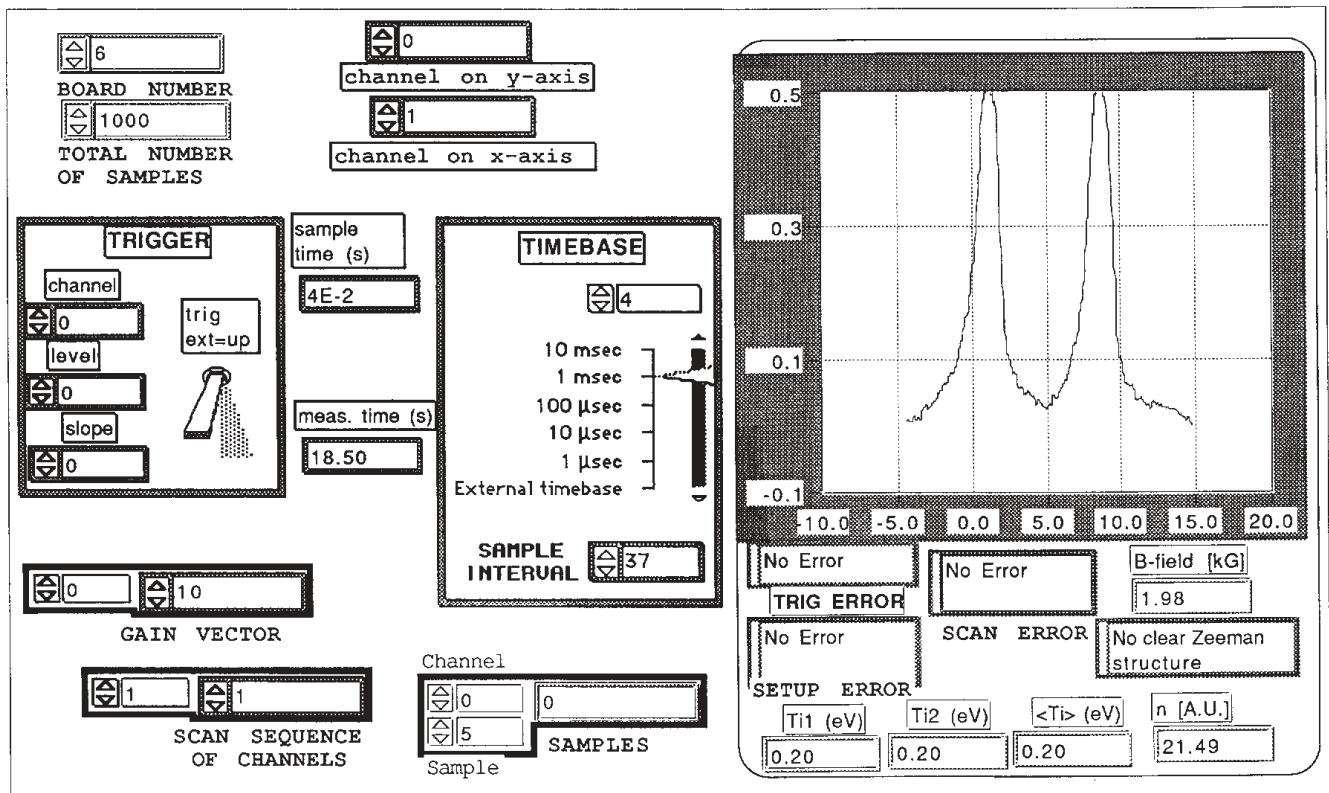


figure 3: Exemple de panneau de commande d'un programme en Labview pour l'acquisition de signaux provenant de l'expérience LMP. Le signal de LIF est ici représenté en fonction de la fréquence du laser. Chacun des deux pics fournit une mesure directe de la fonction de distribution en vitesses des ions du plasma. Température, densité et champ magnétique sont directement calculés par Labview.

panneau de Labview est clairement possible, mais une qualité apte à une publication des graphes ne peut être obtenue qu'en transférant les fichiers sur des programmes spécialisés (Kaleidagraph, Igor, ...). Pour cela, des VI ont été développés et sont à disposition pour un transfert direct des données dans ces programmes. Par exemple, une routine permet d'enregistrer des ensembles de données acquises en Labview comme fichiers Igor; ces fichiers peuvent ensuite être ouverts en Matlab.

Actuellement plusieurs groupes construisent une interface Matlab-Labview, afin d'utiliser directement, dans le cadre du programme d'acquisition, fonctions et éventuellement programmes de calcul préexistants en Matlab. Citons parmi ceux-ci l'Institut d'Automatique du DME au sein de l'EPFL.

Pour revenir à la programmation en Labview (ou langage graphique *G*), il est évident que lorsque l'on crée une application très particulière, on peut avoir besoin de sortir des limites des VI proposés dans le menu du programme. Dans ce cas, Labview permet l'intégration de routines *externes* écrites en d'autres langages (C, Pascal, Fortran, ...), ce qui peut être spécialement pratique par exemple pour récupérer des codes déjà existants ou optimiser des algorithmes en assembleur.

Toutefois, à notre avis, l'atout principal de Labview comme système de programmation graphique reste sa simplicité d'utilisation. En se servant des outils et des VI *standard* de Labview, un expérimentateur sans aucune compétence spécifique peut dans l'espace de quelques semaines installer et commencer à utiliser couramment le système d'acquisition. Paradoxalement, ce faisant, on réalise que ce qui demande le plus de temps et d'effort de réflexion n'est pas la programmation, mais la définition exacte et quantitative du processus de mesure que l'on tâche d'appliquer...

Dans le diagramme les *instructions* sont exécutées dans l'ordre du flux de données: les objets accomplissent leur tâche au fur et à mesure que les données les traversent. Par contre, lorsque l'ordre d'exécution doit être spécifié autrement, des structures de programmation

(séquences ou instructions courantes, comme les boucles For, While, ...) peuvent être employées. Elles apparaissent en forme de cadres entourant les icônes qu'elles contrôlent.

La modularité du programme est une autre caractéristique fondamentale de Labview. La plupart des objets du diagramme sont des VI en eux-mêmes, et peuvent être ouverts, modifiés et exécutés indépendamment. Chaque application est constituée d'une hiérarchie de VI, et peut être vue à son tour comme un seul VI. Elle aura donc une icône la représentant et des broches de connexion pour les entrées et les sorties, et elle pourra être utilisée ensuite à l'intérieur de programmes de plus en plus sophistiqués.

Concernant la vitesse d'exécution, important défaut de jeunesse de Labview, elle est, dans les nouvelles versions, comparable à celle d'un programme en C compilé, grâce à l'introduction d'un compilateur graphique.

Quelques configurations hardware possibles supportées par Labview

Il s'agit principalement de cartes enfichables dans les slots NuBus du MacIntosh. Même si des modèles spéciaux ont été produits pour LC ou SE/30, il est préférable de disposer au moins d'un MacIIci qui, avec 3 slots, permet une éventuelle extension progressive du système.

Il y a essentiellement 3 types de cartes :

- Cartes de DSP, contenant une mémoire jusqu'à 384 Kwords, et permettant un pré-traitement des données et une accélération de l'acquisition, pour des applications particulières (ex. analyse audiofréquence ou traitement d'image).
- Cartes d'interface GPIB et de contrôle du DMA (*Direct Memory Access*). Ces cartes garantissent le dialogue entre l'ordinateur et les instruments GPIB; si couplées avec une carte d'acquisition, elles permettent de transférer des données directement dans la mémoire de l'ordinateur (à la place d'arrêter le processeur pendant que l'enregistrement a lieu, en mode *interrupts*).

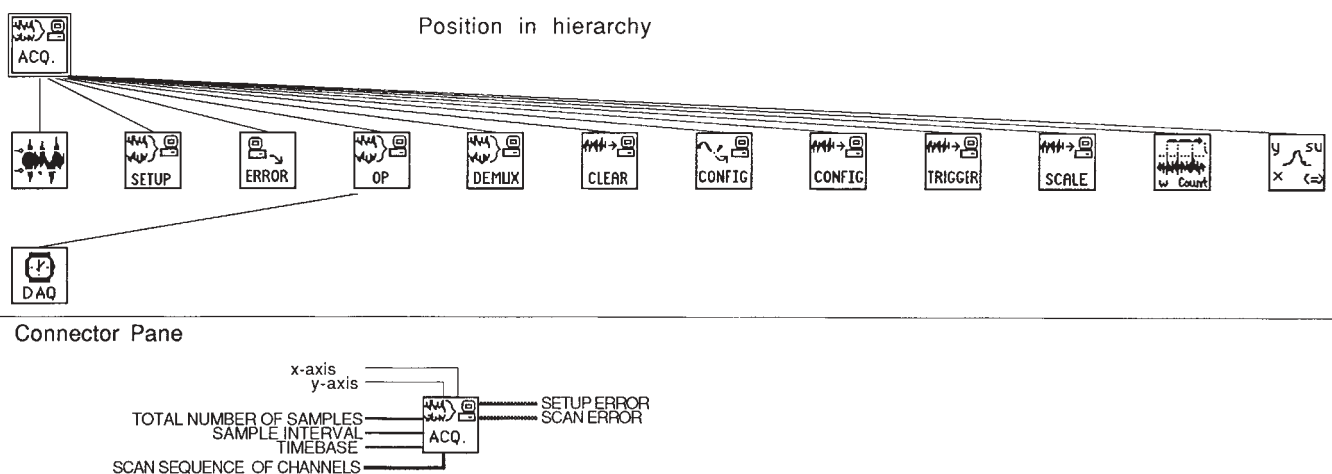


figure 4: Hiérarchie des VI appelés par le programme d'acquisition «ACQ». L'icône «ACQ» peut être utilisée comme simple VI (avec les broches de connexion indiquées en bas) dans d'autres applications.

- Cartes d'acquisition NB permettant l'acquisition et la conversion de signaux analogiques; plusieurs d'entre elles sont douées en même temps d'entrées/sorties digitales et sorties analogiques.

Plusieurs modèles existent, allant de 16 bits / 18 μ sec de période du convertisseur (vitesse d'acquisition maximale: 55 ksamples/sec) à 12 bits, 1 Msamples/sec (NB - A2000).

Une solution relativement économique et à large spectre d'utilisation est celle d'une carte NB-MI0-16, fonctionnant à 12 bits, avec 16 canaux *single-ended* ou 8 en mode différentiel. Le convertisseur peut être choisi à 9 μ sec, ce qui assure un taux d'échantillonnage effectif de l'ordre de 100 ksamples/sec. Pour des applications plus rapides, une carte DMA peut être couplée avec la carte NB-MI0-16 à travers la connexion RTSI (*Real Time System Integration*).

Les deux cartes, intégrées dans un Mac II Ci et pilotées par Labview, peuvent constituer la base d'un système d'acquisition et éventuellement de contrôle (via GPIB ou directement via les sorties analogiques et numériques) d'une expérience de physique. Remarquons que le prix total des deux cartes est bien inférieur à 10'000.— Frs.

C'est effectivement sur cette ligne que les systèmes d'acquisition pour les expériences de physique des plasmas de «basse densité» au CRPP, dans le domaine des aspects non-linéaires et des applications industrielles des plasmas, ont été installés.

Exemple d'application à une expérience de physique des plasmas

Focalisons notre attention sur l'expérience LMP (*Linear Magnetized Plasma*), dont le but principal est la compréhension de phénomènes de dynamique de plasma dans différentes conditions d'interaction entre des ondes électromagnétiques et les particules chargées constituant le plasma.

Cette expérience, mise en opération au début des années '80, dispose déjà d'un système de contrôle complexe (gérant source à plasma, générateur pour le champ magnétique, systèmes de sécurité, etc.). Au système Labview nous ne demandons donc que le contrôle et l'opération de l'acquisition des données sur ordinateur.

Les signaux que nous devons acquérir proviennent essentiellement de sondes électrostatiques ou de photomultiplicateurs qui transforment des signaux lumineux (issus de la Fluorescence Induite par Laser, LIF, dans le plasma) en signaux électriques. Pour obtenir un rapport signal sur bruit convenable, les signaux doivent être pré-traités avant d'être envoyés sur la cartes d'interface. L'expérience nous a appris qu'il est préférable d'effectuer ce pré-traitement en analogique, par exemple par analyseur boxcar ou amplificateur lock-in.

Une mesure typique qui nous fournira un exemple de configuration hardware + software est la mesure de la fonction de la distribution des ions du plasma par LIF. La fréquence d'un laser à colorant est balayée à travers la ligne d'absorption des ions, élargie par effet Doppler dû au mouvement thermique des ions dans la direction du

faisceau laser. Le nombre de photons émis par fluorescence induite pour chaque point du balayage en fréquence du laser (donc pour chaque classe de vitesse des ions) est proportionnel au nombre de particules appartenant à cette classe de vitesse. Puisque la largeur de bande du laser est beaucoup plus petite que l'élargissement thermique de la ligne d'absorption, le balayage complet donnera donc la forme de la fonction de distribution dans la direction du faisceau laser. Le temps du balayage étant relativement long (> 0.1 s), cette application ne nécessite pas l'utilisation du DMA.

Le «pré-conditionnement» par différents types d'amplificateurs permet d'avoir des signaux de voltage de l'ordre de quelques Volts d'amplitude. Les signaux, véhiculés par des câbles BNC, sont introduits sur la carte d'acquisition à travers un panneau d'adaptation. Une attention particulière doit être donnée au montage de ces transducteurs, car d'importants phénomènes d'interférence (*cross-talk*) peuvent se présenter, surtout en mode différentiel (par ex. les câbles correspondant aux deux entrées d'un canal doivent être attachés ensemble le plus étroitement possible).

Le set-up de la carte, comme indiqué précédemment, est fait par certaines icônes du *Labdriver* (le nombre de canaux, la séquence d'acquisition, l'échelle de temps, le gain analogique imposé au convertisseur, ...).

Une fois que la conversion A/D est faite pour les différents canaux, un pré-traitement mathématique a lieu. Ceci consiste dans notre cas en une conversion de Volts en unités physiques, une interpolation gaussienne, un calcul de la température des ions du plasma, et du champ magnétique (d'après le splitting Zeeman de la ligne d'absorption du faisceau laser).

Un exemple de panneau pour l'acquisition d'une fonction de distribution ionique par fluorescence induite par laser est donné en figure 3. Ici, les deux signaux x et y sont respectivement la rampe de balayage de la fréquence laser (donc l'axe des vitesses) et le signal de fluorescence. On observe clairement les deux composantes Zeeman; chacune représente une copie de la fonction de distribution (dans ce cas parallèle au champ magnétique de confinement des particules).

La structure hiérarchique et le schéma des diverses entrées et sorties du programme apparaissent en figure 4. Un exemple d'une séquence de diagramme à blocs est montré en figure 5. Il s'agit ici du cadre qui calcule la valeur du champ magnétique à partir de la séparation des deux pics.

Les mesures sont transférées directement sur des fichiers Igor et ensuite, si nécessaire, en Matlab. Une analyse mathématique plus profonde peut ainsi être opérée par exemple lorsque les effets observés des interactions onde-particules doivent être comparés à des modèles théoriques.

Il est clair que cette configuration est applicable à des mesures «lentes» ou rendues telles par un pré-traitement adéquat.

Dans le cas de mesures rapides, par exemple en régime pulsé, il faudrait utiliser des instruments opérant directement la conversion, par exemple des oscilloscopes digi-

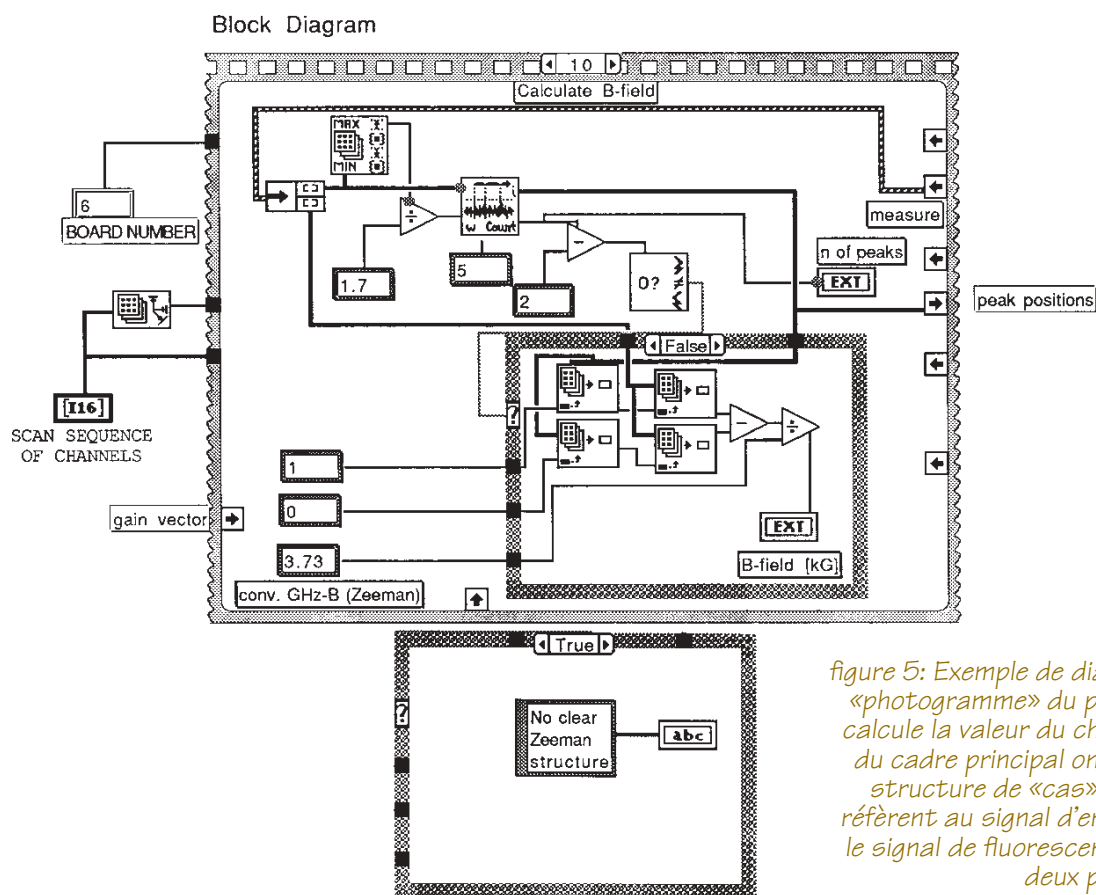


figure 5: Exemple de diagramme en Labview: «photogramme» du programme ACQ qui calcule la valeur du champ B. A l'intérieur du cadre principal on peut localiser une structure de «cas» (false et true se réfèrent au signal d'erreur généré lorsque le signal de fluorescence ne contient pas deux pics)

taux rapides ou des boîtes type Camac, pilotés par Labview à travers la carte de dialogue GPIB. Signalons à ce sujet que NI fournit des VI déjà programmés pour le contrôle/interface d'un nombre très élevé (>150 actuellement) d'instruments à large diffusion; le développement de ces VI (mais ceci est vrai en général pour tous les VI) est tellement rapide qu'il est parfois difficile de trouver un répertoire complet... L'option du DMA est aussi indispensable pour pouvoir construire, en Labview, des générateurs de signaux (jusqu'à 1 kHz pour NB-M10-16 + carte NB-DMA-8-G) et, apparemment, pour disposer d'un trigger analogique en acquisition multicanal.

CONCLUSION

En définitive, le système d'acquisition composé par une ou plusieurs cartes de Input/Output sur Mac II, et par le logiciel de contrôle Labview qui les pilote, nous apparaît très avantageux surtout pour des applications relativement simples, comme celles sommairement décrites. Quelques semaines paraissent être suffisantes pour la mise en opération d'un tel système.

Pour des expériences plus complexes, qui nécessiteraient plusieurs MacII, un nombre important de cartes d'acquisition et par conséquent un grand effort d'intégration via software (l'organisation d'un programme en Labview), la solution Labview n'apparaît plus forcément

la plus simple ni la plus économique.

Les mêmes considérations sont valables pour le cas du contrôle complet et automatisé d'une expérience de physique. En outre des aspects *philosophiques* (est-il sensé de vouloir contrôler entièrement automatiquement un système qui est supposé enquêter sur des phénomènes nouveaux et, parfois, imprévus?), en pratique, cela comporte un effort non négligeable de programmation et une *standardisation* de tous les instruments (par ex. GPIB) qui, dans un laboratoire déjà équipé, peut être très onéreuse.

Dans le cas du contrôle de processus industriels basé sur des procédés bien établis, par contre, Labview trouve une large application potentielle.

Pour les différents domaines d'application, National Instruments assure aussi un support aux utilisateurs de Labview en organisant régulièrement des rencontres régionales où des échanges d'idées, d'applications, etc, peuvent avoir lieu. Il existe aussi une adresse bitnet aux Etats-Unis¹ grâce à laquelle des contacts peuvent être pris (pour des questions spécifiques, des *troc* de VI,...) à différents niveaux. Une *famille* d'utilisateurs est donc en train de se former et de s'accroître!² ■

¹ «INFO-LABVIEW@PICA.ARMY.MIL»; pour les inscriptions, les annulations, etc. écrire à: «INFO-LABVIEW-REQUEST@PICA.ARMY.MIL»

² L'auteur désire remercier Dr.C.Hollenstein pour l'avoir introduit à Labview, P.J.Paris pour une lecture critique du texte, et M.Pedrozzi pour les dessins de la figure 1.

Conférence ERCOFTAC au CNRS

L'Europe face au défi des machines Teraflops

PAR GABRIEL CLERC – SP+R, MARIE-CHRISTINE SAWLEY – SIC ET JAN-
BERNHARD VOS – IMHEF

Le 6 mars dernier, nous avons eu l'occasion de suivre un séminaire d'une journée, organisé conjointement par Ercoftac et Dassault Aviation. Ce séminaire s'est déroulé dans les locaux du CNRS à Paris. Cette journée, qui regroupait environ une centaine de participants, a permis de faire le point sur les tendances dans les disciplines touchées par le calcul de très haute performance à la fois dans le domaine de la recherche et de l'industrie. Environ la moitié des participants provenaient d'institutions de recherche, 40% d'entreprises privées (Dassault Aviation, MBB, Snecma, Renault, Fiat, etc.) ou à participation de l'état (EDF, Aérospatiale, etc.), et quelques entreprises informatiques (NEC, Intel, CRAY, Telmat, TMC, Parsytech, Siemens Nixdorf) complétaient l'assistance.

La journée a débuté par une brève allocution de courtoisie de M. Ferrier, du CNRS, la conférence se déroulant dans les locaux de cette prestigieuse institution. Le Prof. Hirsch, chairman de Ercoftac, a ensuite ouvert officiellement la journée.

Remarque: dans la suite de cet article, nous dénoterons par l'inesthétique néologisme *TF Machines* ($TF = 10^{12}$ opérations par seconde) des ordinateurs du type suivant: C90 (CRAY), CM5 (TMC), machines *superscalaires*, MPP de CRAY, etc. dont les performances s'approchent de cette valeur.

ÉTAT DANS LA RECHERCHE FONDAMENTALE

- La partie académique de la matinée a démarré avec la présentation du Prof. Pironneau, de Paris VI, qui avait choisi le titre provocateur suivant: *Is CFD over because of Massively Parallel Processors?* Les domaines de la CFD qui selon lui bénéficieront le plus des TF machines: écoulements turbulents incompressibles, écoulements compressibles hypersoniques visqueux, industrie pétrolière, écoulement avec effet chimique et de dissociation, les écosystèmes, la turbulence très complexe (flamme). A l'heure actuelle, les méthodes couramment employées comme: Euler, NS, Maxwell, etc., souffrent de limitations par le fait de la capacité encore insatisfaisante des ordinateurs existants. Mais attention, les machines massivement

parallèles actuellement disponibles sont peu conviviales à l'utilisation. Pour le Prof. Pironneau, le mot-clef pour l'avenir de la CFD s'appelle: décomposition en domaines, technique à affiner et méthodes à développer pour assurer les conditions de continuité.

- Uriel Frisch, de l'Observatoire de Nice, s'est attaché à démontrer l'importance des TF machines dans la modélisation de la turbulence fondamentale. Celle-ci reste extrêmement difficile: citons par exemple qu'avec les machines actuelles, la simulation de l'écoulement autour d'une voiture est exact pour une vitesse de ...1cm/sec. Le nombre d'opérations à effectuer dans un calcul d'écoulement visqueux étant proportionnel au cube du nombre de Reynolds, nous voyons que gagner 3 ordres de grandeur dans les performances de calcul, (en passant des GF aux TF machines), ne permet que de gagner un facteur 3 dans la vitesse: d'ici 5 ans, la vitesse pour laquelle l'écoulement pourra être calculé autour de la

même voiture atteindra 3 cm/sec, soit 10 Km/h. D'ici là, les limitations de vitesse nous obligeront peut-être à considérer cette vitesse comme maximale !

Reste que la complexité intrinsèque de la turbulence est très difficile à comprendre. A part les performances brutes que les TF machines nous promettent, celles qui offrent un accès interactif facile présentent un avantage important pour le chercheur qui veut percer cette complexité: les interfaces graphiques puissantes et le haut débit de transfert de données permettent un arrêt sur des phases critiques, affichages et éventuellement grossissement ou zoom sur un point particulier.

- Le prof. Rodi (Université de Karlsruhe) a fait le point sur les apports potentiels des TF machines aux trois méthodes largement employées en CFD: *simulation directe* (résolution des équations de Navier-Stokes temporelles avec un sous-maillage assez fin pour pouvoir capter la turbulence), *Reynolds averaged* et *large eddy simulation (LES)*. Un calcul d'écoulement visqueux avec un nombre de Reynolds de 6600 qui est à l'heure actuelle cher (40 heures à 150 MFlops) ne prendra plus que 20 secondes sur une machine à 1 TF en moyenne. Ce qui est exorbitant aujourd'hui ($Re=20'000$, temps cpu=32 jours à 150 MFlops) devient raisonnable (400 sec. à 1 TF), l'irréalisable ($Re=10^5$, temps cpu= 66 ans à 150 MFlops) se met à la portée des ingénieurs (83 heures à 1TF).

- Julian C.R.Hunt, directeur du Royal Institute of Meteorology de Grande-Bretagne, a souligné l'amélioration considérable de la précision des prévisions météo en Europe: les prévisions à 24 heures en 1975 avaient le même taux de

réussite que celles à 3 jours en 1991. Comme dans tous les domaines de CFD, les spécialistes de la météo font usage de modèles différents. La spécificité de cette discipline réside bien sûr dans l'adéquation permanente qui doit exister entre le flux de données des stations et la production de résultats: nous sommes dans le domaine de prédiction d'événements, qui doit évidemment être disponible avant l'occurrence dudit événement. L'avènement de TF machines permettra d'affiner les prévisions, et la simulation du modèle global devrait en bénéficier tout particulièrement. Signe des temps: les offices météo, qui ont pour la plupart des programmes anciens et en constante évolution, s'orientent désormais vers le tandem machines SIMD (ou MIMD) et Fortran 90.

ETAT DANS L'INDUSTRIE

- Le Dr. Vio, de Fiat, n'a pu venir, mais a transmis l'essence de son message à Pierre Perrier (Dassault Aviation), qui en tant que chairman, s'est chargé de transmettre la quintessence du message. Dans l'industrie automobile, trois types de problèmes sont simulés: la combustion, et le problème annexe de la génération des produits de combustion (polluants), l'étude de la fatigue de certaines pièces mécaniques, et le moule de la carrosserie. Si la majorité des études de problèmes de la seconde catégorie s'effectue en grande partie sur stations de travail, accessoirement aidées d'un supercalculateur, ceux de la première et de la troisième catégorie ne peuvent tout simplement pas être traités sur ce type de machines. D'autre part, les contraintes de production et de rentabilité d'un modèle, les coûts d'études préliminaires et ceux du prototype (environ 9 mois), font qu'une réduction du temps de simulation, si possible d'une qualité supérieure, se traduit par une baisse des coûts. Coûts qui sont le véritable nerf de la guerre que se livrent les constructeurs.

- Le prof. E.H. Hirschel (MBB), a exposé les progrès que des calculateurs tels que les TF machi-

nes peuvent apporter lors du développement des produits d'une industrie comme MBB. Il a recensé trois éléments clefs lors de la conception d'un produit: le coût, le temps et le risque. Quant au produit lui-même, ses caractéristiques sont: la performance, le coût d'exploitation, le risque à l'exploitation, le temps de vie. Les impacts étudiés par le Prof. Hirschel se basent sur l'hypothèse, qui reste à vérifier, que sur une TF machine, le coût d'une opération est divisé par 1000, voire 10'000, par rapport aux machines existant à l'heure: ce qui revient à dire que le prix des configurations devraient rester constant pour des performances 1000 fois plus élevées. L'arrivée des TF machines permet d'envisager l'étude de problèmes très complexes, dont l'importance est capitale et trouvent leur application dans des domaines tels que: diminution du bruit du rotor d'un hélicoptère (couplage de l'écoulement avec le champ acoustique), aéroélasticité (interaction entre la structure de l'avion avec l'aérodynamique), optimisation de trajectoire, notamment dans la phase de décollage, optimisation de la configuration. Un exemple particulièrement frappant de l'intérêt qu'un constructeur comme MBB porte à la simulation: en améliorant le design d'un avion, la consommation de carburant pourrait être divisée par 2 !

- Mr. P. Caseau, de l'EDF, a parlé de l'évolution des moyens de calcul sur les 30 dernières années. La tendance montre que tout les dix ans, un gain de l'ordre de 10 s'observe à coût constant. Tel que l'on peut le prévoir à l'heure actuelle, l'intérêt pour EDF d'utiliser des TF machines réside dans le fait que, pour la première fois, des phénomènes intervenant à des échelles d'espace (prise en charge de 5000 noeuds par rapport à un maximum de 1'000 actuellement) et de temps (entre une milliseconde et une heure) très différentes pourront être simulés à l'aide du même modèle. Ce fait constitue une simplification considérable dans le développement et le maintien de codes différents pour tenir compte de ces modèles différents.

- Peter Lindamood (Gartner Group) a présenté les grandes lignes du projet *Grand Challenge in the US*. Préparée dès 1989 par l'Office de la Science et de la Technologie, la charte définissant les buts à atteindre dans le domaine des ordinateurs et des télécommunications à très haute performance, ainsi que les moyens proposés pour y parvenir, a abouti en 1991: une des premières conséquences en a été le doublement du financement disponible entre 1992 et 1996. Les deux buts visés sont: stimuler l'innovation dans le domaine de l'informatique et de la téléinformatique à très haute performance, et conserver ainsi la suprématie des Etats-Unis dans ce domaine.

CONCLUSION

Les points forts qui ont été relevés lors de la table ronde qui a clos la journée, s'articulent autour de la constatation suivante: la complexité des codes étant croissante, leur durée de vie augmente du fait des coûts de développement. Afin de pouvoir vraiment profiter des possibilités d'ordinateurs tels que les TF machines, la collaboration entre centres effectuant des travaux dans les mêmes domaines pour le développement de codes est essentielle: l'échange de savoir-faire, la mise en commun de spécialistes sur un projet seront plus que jamais à l'ordre du jour pour espérer résoudre ces défis ambitieux, et motivants, que ces chercheurs se sont lancés. Le problème du contrôle de qualité du logiciel produit a également été évoqué, et l'idée de *software factories*, véritables usines de production de logiciel, a été reprise par plusieurs participants.

Il est encore, malheureusement peut-être, trop tôt pour voir si l'Europe est en mesure de relever ce défi, défi dont la survie de pans entiers de l'industrie pourrait bien dépendre d'ici une dizaine d'années. Aucun centre de recherche ou industrie n'a sur ce continent la taille suffisante pour pouvoir le relever seul: le maître mot restera bien la **collaboration**. De par sa multidisciplinarité

Intégration du serveur de fichiers Nestor à l'environnement du Cray2

PAR JEAN-MICHEL CHENAIS, SIC-EXPLOITATION

PRÉAMBULE

La période des tests d'installation et d'acceptation de l'ordinateur Cray-YMP-EL étant désormais terminée, le SIC peut envisager maintenant l'ouverture officielle des services liés à la fonctionnalité du serveur de fichiers à l'ensemble de ses usagers, et en particulier ceux du Cray2. Cet article leur est donc spécialement dédié, toutefois, certains des idées et principes exposés ici peuvent s'appliquer aux autres machines clientes de l'Ecole. En particulier, les utilitaires décrits dans cet article peuvent être à bien plaisir installés sur d'autres machines de département. Pour toute information préliminaire et spécifique au serveur de fichiers, dénommé Nestor, se référer aux articles de mon éminent collègue Martin Ouwehand (SIC/SE) parus dans les numéros précédents du FI.

ACCÈS AU SERVEUR DE FICHIERS NESTOR DEPUIS LE CRAY2

Pour bénéficier des services du serveur de stockage et d'archivage Nestor, les usagers doivent faire une demande d'inscription. L'accès au serveur de fichiers est disponible en priorité à toute personne de l'Ecole, ou collaborateur extérieur s'il est associé à des projets de recherche d'unités de l'EPFL (les cas ne rentrant pas dans ces catégories doivent être traités séparément). Un formulaire peut être obtenu auprès de Mr. Adamo Vionnet, ☎ 021.693 2214, E-mail: vionnet@sic.epfl.ch, et lui être retourné. En réponse, l'utilisateur reçoit la quittance de création de

son compte sur Nestor. Dès lors, à partir du moment où les partitions exportées par Nestor sont montées sur la machine cliente, les fonctionnalités liées au serveur de fichiers sont disponibles.

En principe, toutes les partitions exportées par Nestor sont montées d'office sur le Cray2. En d'autres termes, tous les usagers des départements inscrits sur Nestor retrouvent sur le Cray2 les partitions qui leur ont été attribuées. L'utilisateur a donc le choix de travailler soit sur sa propre station cliente ou sur son serveur de département où ces mêmes partitions sont montées, ou encore sur le Cray2, jouant ainsi le rôle de serveur de calcul pour l'utilisateur, et de client vis-à-vis de Nestor pour les fonctions d'archivage.

Suite aux résolutions prises par la Commission Technique Informatique du 30 octobre 1990, les *usernames* ainsi que les valeurs numériques associées des UID/GID doivent pour chaque usager être uniques sur tous les ordinateurs de l'Ecole. Par ailleurs, le nom du groupe (au sens Unix) auquel l'utilisateur est rattaché correspond d'office à la cellule administrative (département ou institut) dont il dépend. L'utilisateur ne dispose donc en principe plus sur le Cray2 de la liberté de choisir le nom de son groupe (confondu avec celui de son projet), et par la même, de se définir des groupes au sein de son unité. Les décisions de la CTI étant intervenues après la mise en route du Cray2, des dérogations à ce principe restent néanmoins possibles, mais doivent être dûment justifiées.

Afin de rendre possible l'accès au serveur de fichiers, il faut réaliser le *mapping* pour garantir le bon



acheminement des paquets NFS entre Nestor et Cray2. Le mapping employé s'établit sur la base de l'unicité du nom des usagers (*username*) et des groupes (*group names*). Selon cette philosophie, propre au système Unicos, l'égalité des UID/GID (valeurs numériques) entre serveur (Nestor) et client (Cray2) n'est pas une condition nécessaire pour le bon fonctionnement de NFS. Toutefois, la mise en fonction du serveur de fichiers implique quelques contraintes pour les usagers. En particulier, étant donné que d'office le nom des groupes doit refléter l'appartenance administrative de l'utilisateur, le nom du groupe doit en principe être changé sur le Cray2 et s'aligner sur celui retenu sur Nestor.

UTILISATION DE NESTOR DEPUIS LE CRAY2

Par définition, aucun accès interactif (*login*) sur le serveur de fichiers n'est possible. Nommément, aucun *process* explicitement activé par l'utilisateur ne peut tourner sur Nestor, dont le rôle est confiné à la disponibilité de ses ressources *on-line* (disques) à travers NFS et FTP et, en *background*, ses ressources *off-line* (cartouches pour l'archivage à travers DMF : *Data Migration Facility*).

Au niveau application, les deux seules voies de communication entre Cray2 et Nestor passent l'une par NFS, l'autre par FTP. Le protocole FTP, en mode binaire, est le plus performant : les taux de transfert possibles dans la configuration actuelle par FDDI sont de l'ordre de 1.5 Megabytes/sec. Le protocole

NFS est plus lent (de l'ordre de 0.5 Megabytes/sec.), mais a sur FTP l'avantage de la transparence. Ces chiffres devraient augmenter d'un facteur de l'ordre de 2 à 3 avec l'installation de Giganet.

Dans ces conditions, pour ses applications, l'utilisateur a intérêt à disposer ses fichiers de travail directement sur le Cray2. En effet, l'utilisation directe à travers NFS des fichiers résidant sur le Nestor peut sensiblement augmenter les temps d'exécution des programmes tournant sur le Cray2 : de l'ordre de 20%.

Il est donc intéressant de considérer l'espace disque du serveur de fichiers comme un medium d'archivage intermédiaire, et de mettre à disposition des usagers des moyens simples, sous forme d'utilitaires, pour effectuer ces opérations de changement de résidence entre machine cliente (Cray2) et le serveur de fichiers (Nestor)

L'ensemble de ces utilitaires permettent en quelque sorte de simuler, mais de façon asynchrone, une partie des fonctions de *caching* telles que prévues par AFS/DFS.

LA TRANSPARENCE D'ACCÈS AUX FICHIERS RÉSIDANT SUR NESTOR.

L'idée de base consiste à faire en sorte que l'arborescence des fichiers, telle que définie par l'utilisateur au niveau de la machine cliente, soit reproduite sur la machine serveur. La structure hiérarchique des fichiers résidant sur Nestor est en fait la copie miroir conforme de celle résidant sur le Cray2. L'utilisateur, par l'intermédiaire d'utilitaires ad-hoc, provoque le déplacement de ses fichiers entre machine client et serveur, et la création le cas échéant de toute l'arborescence intermédiaire nécessaire. Un fichier déplacé reste toujours accessible et utilisable selon les moyens traditionnels, car il est remplacé sur la machine cliente par un pointeur (*soft link*) désignant son chemin d'accès (*full pathname*) sur Nestor.

Cette façon de faire permet à l'utilisateur d'avoir une vision globale et unique de son arborescence, que ses

fichiers résident sur Cray2 ou Nestor. La transparence est réalisée par le fait que fonctionnellement le déroulement des applications n'est pas modifié par le changement de résidence des fichiers, aux performances près.

Les commandes sont les suivantes :

putfs : [options] filename – cette commande déplace (commande *mv* au sens Unix) le fichier référencé depuis Cray2 vers Nestor. Le fichier sur Cray2 est remplacé par un pointeur pointant vers sa nouvelle résidence sur Nestor. En cas de nécessité, toute arborescence intermédiaire est créée sur Nestor.

getfs : [options] filename – rapatrie vers le Cray2 le fichier précédemment déplacé. C'est donc l'opération inverse: toute arborescence intermédiaire est effacée si elle ne contient plus aucun fichier.

Les attributs (date de dernier accès par exemple) liés à chaque fichier ne sont pas modifiés lors d'un *putfs* et *getfs* successifs.

Les différentes options permettent de contrôler l'exécution des commandes. En particulier est prévu le traitement des possibilités de *clash* entre noms de fichiers identiques mais issus de différentes machines, donc au contenu vraisemblablement différent. En effet, les copies miroir issues des différentes machines clientes pouvant ne pas être identiques, un répertoire supplémentaire et représentatif de chaque machine cliente peut être inséré (sur option ou par défaut) dans le *full pathname* du fichier déplacé sur Nestor.

Deux nouvelles commandes, (en fait des *alias*), permettent, à travers NFS, de se déplacer librement dans les arborescences de Nestor, pour accéder aisément aux fichiers. Ce sont:

nestor – activée sur le Cray2, cette commande place l'utilisateur au sommet de son arborescence sur Nestor. Le *prompt* affiché reflète ce changement.

cd [argument] – cette re-définition de la commande Unix *cd* traditionnelle autorise tout aussi facilement l'utilisateur à se déplacer dans l'arborescence à sa disposi-

tion, tout en modifiant le *prompt* (modifiable par l'utilisateur) de façon que celui-ci reflète le nom de la machine sur laquelle il se trouve positionné.

Deux nouvelles variables d'environnement sont désormais disponibles. Ce sont : **NESTOR** et **CRAY2**, qui désignent respectivement les *home directory* de l'utilisateur sur Nestor et Cray2.

Remarque :

Ces utilitaires « maison » sont mis à disposition à bien plaisir sur le Cray2, et sont incorporées d'office dans l'espace de travail interactif de chaque utilisateur, par le biais des fichiers d'environnement *.cshrc* et *.login* « standard SIC », et pour autant que sa validation est effective sur Nestor. L'intégration de ces utilitaires dans le Bourne shell, donc pour le système batch NQS du Cray2, est en cours de réalisation.

Pour une description plus complète de ces utilitaires, se référer à l'information délivrée par la commande *news NESTOR* du Cray2.

TRANSFERT DE L'APPLICATION DMF DU CRAY2 VERS NESTOR

L'application DMF, mise en fonction sur le Cray2 depuis l'automne 1991, est déplacée sur Nestor, où elle assure désormais les fonctions d'archivage et de gestion automatique de l'espace disque. Les commandes d'archivage *dmput* et *dmget* sont donc virtuellement remplacées par les commandes *putfs* et *getfs* décrites ci-dessus. Leur utilisation permet en effet aux utilisateurs de déplacer leurs fichiers sur les disques de Nestor, tout en réduisant leur quota sur le Cray2. Ce déplacement peut donc être assimilé à une opération d'archivage de premier niveau sur Nestor.

Il est envisagé, sans que cela soit à l'heure actuelle formellement décidé, de déplacer automatiquement les fichiers utilisateurs depuis le Cray2 vers Nestor, dès que les quotas sont dépassés. Selon des critères à définir (basés principalement sur leur âge), ces fichiers seraient déplacés afin de ramener l'espace disque utilisateur à une proportion donnée de son quota sur Cray2. ■

Les saisons de 1991 du projet SUSP

seconde partie

Le service *pascal* est pour bientôt

PAR MARIE-CHRISTINE SAWLEY. SIC-ASSISTANCE

Non, il ne sera fait nulle référence à la liturgie pascalle dans cet article, contrairement à ce que le calendrier pourrait laisser entendre, ni au langage qui a fait notre bonheur d'étudiant. En fait, le troisième et dernier volet du projet SUSP en 91, le serveur de calcul au SIC, est l'objet de cet article. Un clin d'oeil vous renvoie en fin de texte pour connaître les raisons de cette appellation.

Cet article fait suite à celui du même auteur paru dans le FI no 8/91, du 22 octobre 1991. Le feu vert du CEPF (Conseil des Ecoles) pour la commande du serveur de calcul central du projet SUSP n'ayant été donné qu'à fin novembre de l'année dernière, nous avons préféré faire coïncider l'annonce de la décision avec celle de l'arrivée de la machine.

SEPTEMBRE 91: LES RAISONS D'UN CHOIX

Un complément d'informations et de benchmarks (y compris simulation de charge maximale sur la machine) était alors demandé à chaque constructeur, et les contacts se sont intensifiés afin de rendre l'analyse la plus impartiale et exacte possible. Le 30 septembre 1991, après prise de position du groupe SUSP, de la CTI et de la CI, la Direction de l'Ecole se prononçait en faveur de la solution Cray.

Les sociétés Convex et Cray ont parfaitement collaboré dans cette dernière phase, rendant la décision finale d'autant plus difficile! La partie a été très serrée, car ni les performances, ni les prix globaux n'ont permis de départager significativement les deux solutions, équivalentes sur les points suivants:

- avoir un potentiel d'évolution important à un coût modéré;
- le constructeur retenu doit s'engager fermement sur DCE (Distributed computing environment) d'OSF.

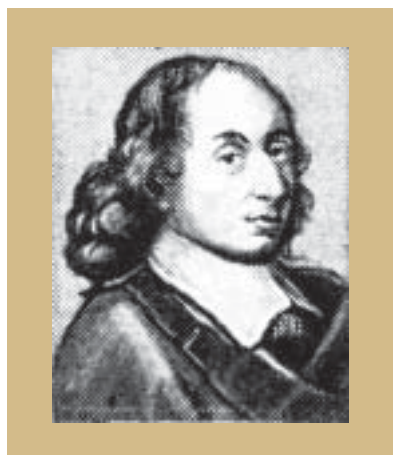
JUIN 91: DEUX SOLUTIONS RETENUES

L'analyse des offres et les résultats des benchmarks effectués pendant le mois de mai laissaient, pour le serveur de calcul SUSP au SIC, deux configurations possibles aux yeux de la CTI et de la CI. La première consistait en un Cray Y-MP4E (version réduite du Y-MP dont le châssis peut contenir un maximum de 4 processeurs), mono-processeur, doté de 512 MB (64 MW) de mémoire centrale (bande passante 5.3 GB/sec. par processeur), de 256 MB (32 MW) de SSD (Solid state disk) et d'un IOC (I/O cluster). La seconde portait sur un Convex C3840, quadri-processeur, doté de 2 GB de mémoire centrale (bande passante: 0.5 GB/sec par processeur) et un I/O processeur.

- ampleur de la configuration (taille et débit de la mémoire, capacités disques);
- puissance de calcul;
- rapport prix/performances;
- bonne intégration par rapport aux mondes VAX et Cray, déjà bien implantés sur le site.

En fait, les éléments qui nous ont permis de trancher sont, pour la plupart, des considérations à moyen et long terme (stratégie d'évolution):

- EXTENSIBILITE: le coût d'exploitation sur 5 ans y compris le doublement de la capacité de calcul (ajout d'un processeur supplémentaire chez Cray, de 4 chez Convex) est sensiblement inférieur chez Cray. A relever également que la solution Convex serait alors au maximum 🐘



CONTEXTE

Il n'est peut-être pas inutile de rappeler ici les contraintes propres à la fonction «serveur de calcul» qui avaient été formulées dans le cahier des charges du projet SUSP:

- montrer des performances scalaires sensiblement meilleures que celles des moyens déjà installés sur le site de l'EPFL;
- offrir une plate-forme de traitement batch compatible avec NQS domaine public;
- avoir un catalogue de logiciels d'applications suffisant;

de sa capacité, alors que le châssis du Cray Y-MP4E permettrait encore l'installation de 2 processeurs supplémentaires, ce qui quadruplerait la configuration initiale;

- DCE: disponibilité plus rapide des logiciels d'intégration chez Cray;
- SUPPORT: en Suisse, le personnel de Cray est à même de fournir un support plus étoffé que Convex. Rappelons que, parmi les critères, un des plus contraignants et qui s'applique à l'ensemble des machines sur le site, s'énonce ainsi: minimisation du personnel supplémentaire nécessaire à la gestion;
- AVENIR: Cray a, pour le moment du moins, un programme plus ambitieux de développement de processeurs massivement parallèles que son concurrent. Le sous-système MPP de Cray sera connecté à des machines *traditionnelles* telles que le Y-MP4E.

CONFIGURATION RETENUE

La machine qui vient d'être livrée le 6 avril dernier présente les caractéristiques suivantes:

- un processeur, temps de cycle 6 ns;
- mémoire centrale de 512 MB (64 MW de 64 bits);
- un IOC, processeur de contrôle relié par un canal HISP à 200 MB/sec au processeur. Une liaison HiPPI à 800 Mb/sec le relie au HUB 1000 d'UltraNet, assurant la liaison avec le Cray2, le serveur de fichiers Nestor et les serveurs de département (pour le moment, seul le serveur SGI du DMA est connectable);
- une batterie de 8 disques DD 60, totalisant un espace disque d'environ 16 GB.

Il est à remarquer que le Solid State Disk de 256 MB (32 MW), qui sera relié par un accès très rapide (1 GB/sec) à la mémoire centrale sera livré dans les prochains mois.

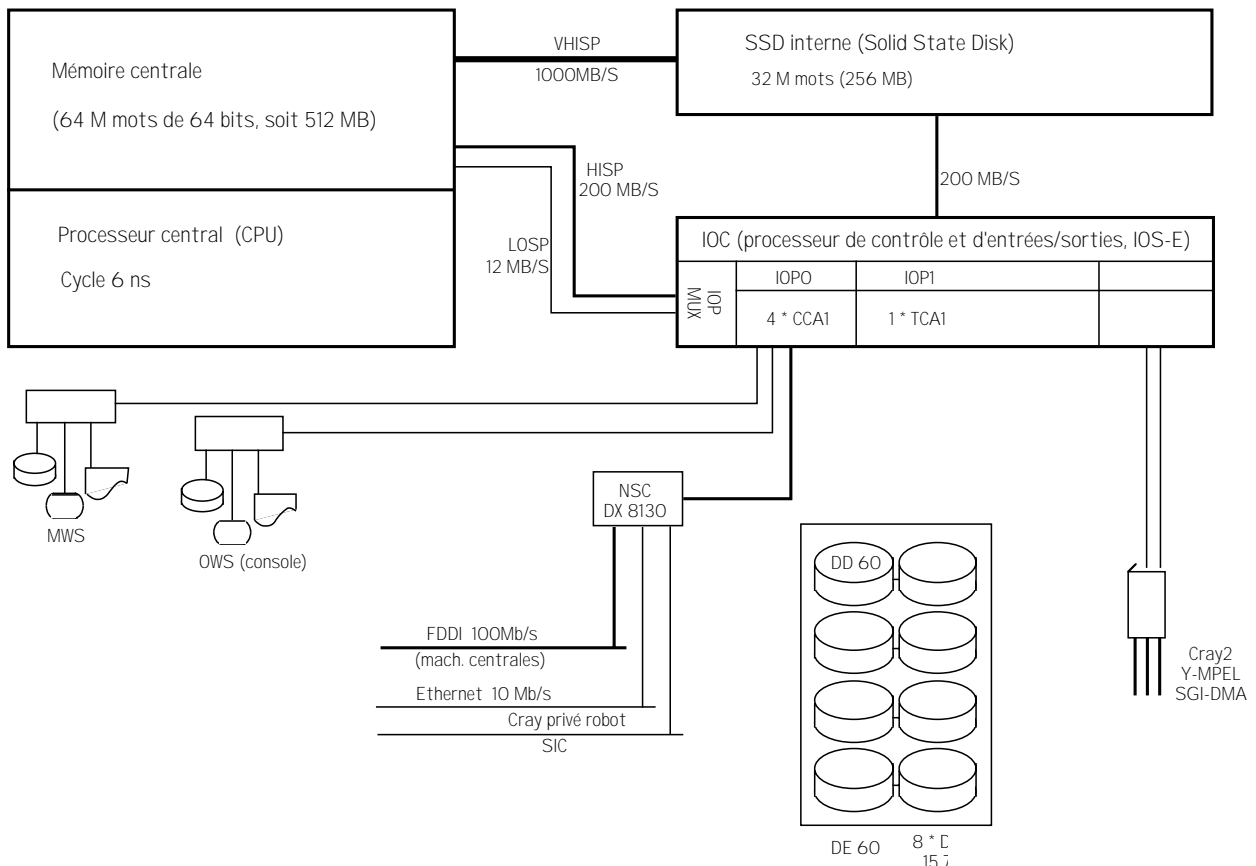
Le plan ci-après vous permet de prendre connaissance des détails de la configuration.

PASCAL: QUEL SERVICE?

Le nouveau service a reçu le nom de *pascal*, en hommage à Blaise Pascal, qui, parmi ses multiples talents, était un mathématicien hors pair. Et non pas, comme un esprit de contradiction aurait tût fait de relever, *pascalaise*. Il n'est pas prévu, pour le moment, d'ouvrir le service à l'extérieur de l'Ecole (sauf pour des collaborations avec des unités de l'Ecole): les besoins clairement recensés pendant la phase d'évaluation doivent d'abord être satisfaits. La mise en service est prévue dans les prochaines semaines, et les détails pratiques vous seront communiqués par voie habituelle: Flash Informatique et News.

Ce service sera totalement intégré sur le réseau et son accès banalisé sous X, comme n'importe quel

CRAY Y-MP4E / 164 (SN 1913)



LES LOGICIELS POUR PASCAL

Suite à une enquête dans les départements, principalement par l'intermédiaire des DDI, le nouveau serveur de calcul Pascal sera équipé des logiciels suivants:

- compilateurs : Fortran, Pascal, C, C++;
- mathématiques : Lapack, NAG, matlab, maple;
- graphiques: Uniras (concernant MPGS et Explorer, les investigations ne sont pas terminées)
- appels à distance: AIT;
- Base de données distribuée: mem-com;
- dès que disponible, le compilateur Fortran 90 de NAG (Nagware).

Si vous croyez avoir été oublié, n'hésitez pas à me contacter:

E-mail: possoz@sic.epfl.ch
tél.: 021/693 22 49.

Anne Possoz, SIC-Logiciels

serveur de département: cette volonté d'ouverture a été maintes fois exprimée dans le cadre du groupe SUSP. Une plus grande flexibilité est à relever dans la licence d'exporta-

tion négociée pour le serveur *pascal*: en résumé, toute personne de nationalité suisse aura bien sûr accès, ainsi que tout résident étranger en Suisse en possession d'un permis B ou C délivré par les autorités helvétiques. Pour les autres personnes de nationalité étrangère, l'accès ne sera pas restreint pour autant qu'ils ne soient pas ressortissants d'un pays figurant sur la liste COCOM: des informations plus précises peuvent être obtenues auprès de Michel Jaunin, chef de la section exploitation. Il s'agit là d'une importante concession et nous ne pouvons que nous réjouir de cet assouplissement. Nous espérons pouvoir annoncer prochainement que les mêmes règles pourront être appliquées au Cray2, dès que les accords formels avec le DOC (Department of Commerce) à Washington seront connus. Le CEPF (Conseil des Ecoles) et l'OFAEE (Office fédéral des affaires économiques extérieures) continuent à négocier une ouverture encore plus importante avec le gouvernement des États-Unis.

Pascal va démarrer son exploitation sous le système UNICOS 6.1.5a, comme ses confrères déjà installés: *cray2* et *nestor*. Anne Possoz vous donne la liste des logiciels qui seront initialement installés sur *pascal* (voir encadré): il est évident que

cette liste ne peut être qu'évolutive, en fonction du développement du service que cette machine sera amenée à rendre et des besoins des utilisateurs.

Les capacités intrinsèques de *pascal* en font une machine intermédiaire entre le super-ordinateur Cray2 et les serveurs de département. Le service offert se développera dans cette direction, assurant le complément de puissance nécessaire entre ses deux mondes. Dans les prochains 18 mois, l'équipement de certains départements va se renouveler considérablement, certains s'équiperont de serveurs SUSP: les années 92/93 sont à ce titre, des années charnières. La continuation d'une bonne collaboration entre les départements et services permettra aux utilisateurs de tirer pleinement profit des moyens à disposition, et à l'École dans son ensemble d'aborder ce passage avec succès.

Post-scriptum: vous avez dit *passage*?

Un lettré nous ferait remarquer qu'en grec ancien, passage se dit *pashka*, origine de *pâque* et de son adjectif pascal. Ah, que c'est bon d'oublier un peu la technique de temps en temps ! ■

MATLAB V4 . 0

The MathWorks Inc. COMSOL AG

EPF Lausanne

Friday 01. May 1992

14.15-16.00

**Service Informatique Central
(SIC)**

Conference room, Building MA

LIVE DEMONSTRATION

PROGRAM

- Introduction
- MathWorks - Company and Products
- MATLAB V4. 0 - What's new ?
- Live-Demonstration MATLAB V4 .0
 - New Functions
 - Scientific visualization
 - Expanded 2-D- and 3-D-Graphics
 - Animation & Sound
 - Expanded debugging capabilities
 - Sparse matrix support
- SIMULINK - Live-Demonstration
 - A graphical System for simulation of dynamical systems (SIMULINK

was former called SIMULAB)

- Discussion, Questions and more Demos upon request

The MathWorks Swiss Tour 1992:

April 29 - Sandoz

Pharma,

April 30 - ETH Zurich,

May 1 - EPF Lausanne

Dipl. Ing. TH Carl Bergström, COMSOLAG, Mr. Steve Lipsey, Vice President of The MathWorks, Inc. USA, Mr. Kevin Evans, The MathWorks, Inc., Mr. Joe Hicklin Co-author of SIMULINK, The MathWorks, Inc.

For further information please contact Carl Bergström, COMSOLAG, ☎031/961 70 11

FI 4 - 28 avril 1992

Journée d'études sur les réseaux de neurones



Dans le cadre de son programme de réactualisation des connaissances, l'A3E2PL, Association amicale des anciens élèves de l'EPFL, propose une journée d'études :

Thème	Les réseaux de neurones
Date	le 22 mai 1992 - 09.00 à 16.30h
Lieu	EPFL (Ecublens) - auditoire CE2

OBJECTIFS

La quête de la compréhension du cerveau possède une connotation presque mystique aussi forte que la conquête de l'espace ou des forces fondamentales. Il est difficile de se rendre compte de l'importance réelle que les réseaux de neurones vont prendre dans l'avenir. Ils constituent un enjeu pour la Science car ils nécessitent la mise en commun de nombreuses disciplines jusqu'alors séparées. Parallèlement, ils apportent au monde industriel des nouveaux outils qui promettent des solutions intéressantes permettant d'aborder des problèmes complexes.

Cette recherche a déjà conduit à de nombreuses applications prometteuses dans des domaines où la modélisation des phénomènes s'est avérée insuffisante. C'est le cas de la robotique, de l'analyse de données et de la reconnaissance des formes en général.

Cette journée a pour objectif de faire le point sur l'état du domaine et transmettre les connaissances des chercheurs vers les ingénieurs de l'industrie désireux de connaître ces nouvelles techniques. On abordera des problèmes concrets dont les solutions neuro-mimes seront expliquées en détail et comparées à des solutions plus traditionnelles.

PRIX DU SÉMINAIRE

sfr.	200.-	membres EPFL ou membres A3E2PL à titre professionnel
sfr.	100.-	membres A3E2PL à titre individuel
sfr.	50.-	étudiants

(comprenant cours, documentation et repas).

RENSEIGNEMENTS

Secrétariat A3E2PL
EPFL - GCB - 1015 Lausanne
☎ 021/693 20 93 - Fax 021/693 50 60

Connaissez-vous

SPEEDUP ?

PAR MARIE-CHRISTINE SAWLEY, SIC-ASSISTANCE,
MEMBRE DU COMITÉ SPEEDUP

Les briscards de l'informatique ont peut-être encore en mémoire ce dessin de Tuna paru dans le FI du 20 octobre 1987: deux Macintosh étaient tendrement assis l'un à côté de l'autre sur un banc public, alors que les câbles de leur souris s'enlaçaient amoureusement; un Mac portait l'écusson vaudois, l'autre l'ours bernois. Notre dessinateur avait été inspiré par l'annonce de la naissance du fruit de ce rapprochement intercantonal: le groupe Speedup. En fait, il s'agissait d'une conception à 4, puisque les initiateurs en étaient: H. Burkhart de l'Université de Bâle, R. Gruber de l'EPFL, R. Henzi de l'Université de Berne et H.-P. Lüthi de l'ETHZ.

L'idée était simple, mais originale: créer en Suisse un forum d'échange d'informations, de connaissances matérielles et logicielles entre chercheurs et ingénieurs, du monde académique et industriel, intéressés au calcul sur machines haute performance. Les moyens: des séminaires (en moyenne 3 par année) se déroulant à Berne, et un journal. Pendant trois ans, Speedup a fonctionné selon ce schéma, permettant dans les séminaires, généraux ou spécialisés, des prises de contact, des échanges d'expérience, des discussions fructueuses.

Depuis l'installation en 1991 à Manno (TI) du Centro Svizzero di Calcolo Scientifico, Speedup et une partie de son infrastructure ont aussi pris le chemin du Sottoceneri. Mais ce n'est certainement pas dans le but de s'isoler, bien au contraire. Avec un comité élargi à une douzaine de personnes, le groupe s'est fixé maintenant pour but l'organisation de séminaires bisannuels d'une durée de 2 jours, et la parution d'un journal, le Speedup Journal, normalement à une fréquence semestrielle. Si le premier numéro du nouveau journal devrait voir le jour à la fin du printemps, deux séminaires, l'un en juin 91, l'autre en mars 92, ont déjà eu lieu au Tessin selon une nouvelle formule consistant à se rencontrer du jeudi midi au vendredi midi, passant de la salle de conférence au repas en commun le soir, ce qui évite les courses en fin de journée pour attraper son train de retour, et permet de nouer plus de contacts.

Le prochain séminaire aura lieu les 10 et 11 septembre à l'hôtel du Signal de Chexbres, dans le canton de Vaud, sur les sujets suivants: dynamique moléculaire et calcul massivement parallèle. Les lecteurs figurant sur la liste de distribution de Speedup recevront l'invitation et le programme prochainement. Les autres sont cordialement invités à s'adresser à l'auteur afin d'avoir plus de renseignements. sawley@sic.epfl.ch (tél. 021/693.2242) ■

formation

Les cours ci-dessous sont ouverts à tous, membres ou non de l'EPFL en priorité et gratuitement pour le personnel de l'EPFL. Nous pouvons également organiser des cours à la demande, un logiciel vous intéresse plus spécialement, faites-le nous savoir !

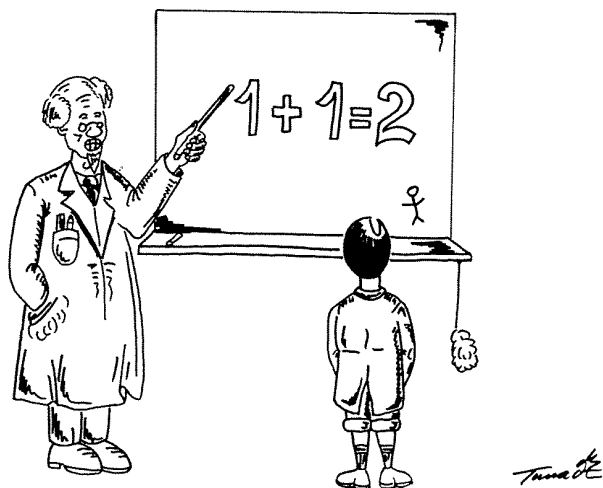
Inscriptions et renseignements :

Josiane Scalfo, SIC-EPFL, CP 121, 1015 Lausanne

☎ 693.2244 (le matin uniquement)

Fax : 021 693 22 20

QuickMail ou E-mail : scalfo@sic.epfl.ch



CRAY

Introduction au Cray-2 (1 demi-jour)

Ce cours permettra aux débutants d'acquérir des notions de base de l'utilisation du Cray2. Le programme sera :

- Introduction au Cray2
- Connexion à la machine
- Unicos
- Cartouches
- Aide
- Système de fichiers
- Langages et outils
- Travailler en batch
- Applications

Renseignements :

Jacqueline Dousson, SIC-SA, E-Mail : dousson@sic.epfl.ch

N° C44 07.05.92 08h30 - 12h30

MACINTOSH

N° Mac257 - Cycle de base complet (6 demi-jours)

Intro. Macintosh	04 & 07.05.92	13h30 - 17h30
Intro. MacDraw Pro	11.05.92	13h30 - 17h30
Intro. Excel 3.0	14.05.92	13h30 - 17h30
Intro. Word 4.0	18.05.92	13h30 - 17h30
Intro. FileMaker Pro	21.05.92	13h30 - 17h30

N° Mac258 - Cycle de base complet (6 demi-jours)		
Intro. Macintosh	03 & 05.06.92	08h15 - 12h00
Intro. MacDraw Pro	10.06.92	08h15 - 12h00
Intro. Word 4.0	12.06.92	08h15 - 12h00
Intro. Excel 3.0	17.06.92	08h15 - 12h00
Intro. FileMaker Pro	19.06.92	08h15 - 12h00

Word 4.0	avancé	(3 demi-jours)
N° Mac262	07, 14 & 21.05.92	08h15 - 12h00

FileMaker Pro	avancé	(3 demi-jours)
N° Mac276	01, 03 & 10.06.92	13h30 - 17h30

Excel 3.0	avancé	(3 demi-jours)
N° Mac263	18 & 25.05 & 01.06.92	08h15 - 12h00

Excel 3.0, macros		(1 demi-jour)
N° Mac271	06.05.92	08h15 - 12h00

Scanner et OmniPage		(2 demi-jours)
N° Mac270	02 & 04.06.92	08h15 - 12h00

Illustrator		(2 demi-jours)
Connaissances approfondies d'un logiciel de dessin indispensables !		
N° Mac266	16 & 16.06.92	08h15 - 12h00

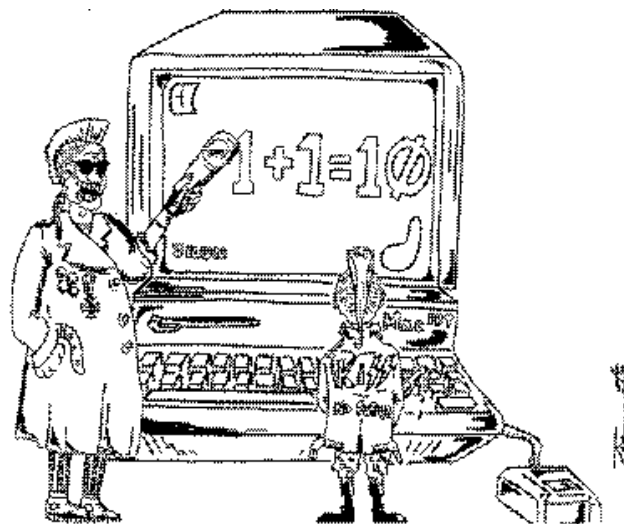
ClarisCAD		(2 demi-jours)
N° Mac272	13 & 15.05.92	08h15 - 12h00

QuickMail		(1 demi-jour)
N° Mac268	20.05.91	08h15 - 12h00

Facilités du Mac (système 7.0)		(2 demi-jours)
N° Mac274	05 & 12.05.92	13h30 - 17h30

Système 7.0, partage de fichiers		(1 demi-jour)
N° Mac273	13.05.92	13h30 - 17h30

X sur Macintosh		(1 demi-jour)
N° Mac275	26.05.92	08h15 - 12h00



PC

Introduction au DOS	(2 demi-jours)
N° PC102	19 & 21.05.92
13h30 - 17h30	
Introduction au Windows 3.0	(3 demi-jours)
N° PC103	26.05 & 02, 04.06.92
13h30 - 17h30	
Introduction à Word for Windows 1.1	(2 demi-jours)
N° PC99	04 & 06.05.92
N° PC104	09 & 11.06.92
13h30 - 17h30	
Introduction à Excel 3.0	(2 demi-jours)
N° PC100	11 & 13.05.92
08h15 - 12h00	
Introduction à Designer 3.01	(2 demi-jours)
N° PC101	18 & 20.05.92
08h15 - 12h00	
Introduction à VisualBasic	(2 demi-jours)
N° PC107	05 & 12.05.92
08h15 - 12h00	
Introduction aux réseaux	(1 demi-jour)
N° PC118	29.04.92
13h30 - 17h30	

Perfectionnement

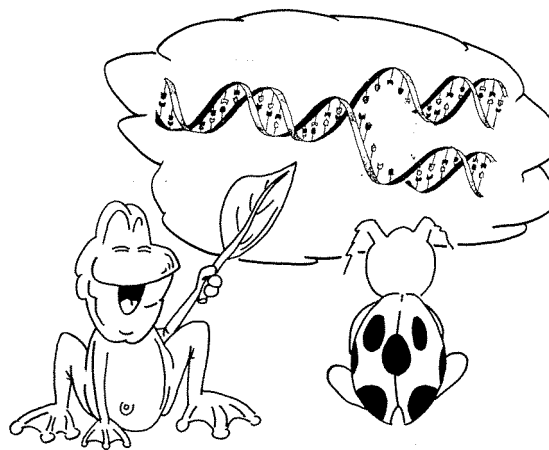
Excel 3.0	avancé	(2 demi-jours)
N° PC113	21 & 28.04.92	13h30 - 17h30
Excel 3.0, base de données		
(1 demi-jour)		
N° PC114	07.05.92	13h30 - 17h30
Excel 3.0, macros		(1 demi-jour)
N° PC115	14.05.92	13h30 - 17h30

Designer 3.01	avancé	(2 demi-jours)
N° PC116	04 & 11.06.92	08h15 - 12h00
SuperBase	intensif	(5 demi-jours)
N° PC117	06, 07, 08, 09 & 10.07.92	08h15 - 12h00
Télécom. PC	avancé	(1 demi-jour)
N° PC119	10.06.92	13h30 - 17h30
VisualBasic	avancé	(2 demi-jours)
N° PC108	03 & 10.06.92	08h15 - 12h00
Word for Windows 1.1	avancé	(2 demi-jours)
N° PC112	17 & 24.06.92	08h15 - 12h00

UNIX

Introduction à Unix (1 demi-jour)

Renseignements :
 Franck Perrot, SIC-SA, E-Mail : perrot@sic.epfl.ch
 N°U42 19.05.92 08h30 - 12h30



Toma

FORMULE D'INSCRIPTION POUR LES COURS

merci de ne pas grouper sur le même bulletin, les cours Mac, PC, Unix etc.

A retourner à Josiane Scalfò, SIC-EPFL, 1015 Lausanne

N° du cours	N° cours de remplacement	Nom du cours
.....
.....
Nom :	Prénom :	E-mail :
Institut :	Dépt :	Adresse :
s'engage à suivre les cours dans leur intégralité et à respecter l'horaire.		
Tél. :	Date :	Signature :
Autorisation du chef hiérarchique (nom lisible et signature) :		

calendrier

Date	Heure	Lieu	Réunion
<i>AVRIL 1992</i>			
Mardi 28	14h00	Salle Conférences SIC	Covax — Réunion des Vax-Managers <i>P. Lachaize, ☎ 021-693.22.54, E-Mail : lachaize@sic.epfl.ch</i>
<i>MAI 1992</i>			
Vendredi 1er	14h15	Salle Conférences SIC	Matlab V4.0 (lire en page 20) <i>J.-J. Dumont, ☎ 021-693.2224, E-Mail : dumont@sic.epfl.ch</i>
Mercredi 6	10h00	Salle Conférences SIC	APOTEC — Groupe des utilisateurs de stations Apollo <i>Ion Cionca, ☎ 021-693.45.86, E-Mail : cionca@sic.epfl.ch</i>
Jeudi 7	12h00		DÉLAI DE RÉDACTION DU FI N° 5
	14h15	Salle Conférences SIC	PolyPC — Groupe des utilisateurs d'IBM PC et compatibles <i>Ch. Zufferey, ☎ 021-693.45.98+3, E-Mail : zufferey@sic.epfl.ch</i> <i>Serveur : sicsoft, polypc, polypc</i>
Jeudi 21	14h15	Salle Conférences SIC	PolyPC — Groupe des utilisateurs d'IBM PC et compatibles
Vendredi 22	09h00	Auditoire CE2	Journée d'études sur les réseaux de neurones (voir page 21)
<i>JUIN 1992</i>			
Mardi 2	14h15	Salle Conférences SIC	CTI — Commission Technique Informatique <i>M. Reymond, ☎ 021-693.22.10, E-Mail : reymond@sic.epfl.ch</i>
Mercredi 3	10h15	Salle Conférences SIC	HPLine — Groupe des utilisateurs de Hewlett-Packard <i>Ion Cionca, ☎ 021-693.45.86, E-Mail : cionca@sic.epfl.ch</i>
Jeudi 4	12h00		DÉLAI DE RÉDACTION DU FLASH INFORMATIQUE N° 6
	14h15	Salle Conférences SIC	PolyPC — Groupe des utilisateurs d'IBM PC et compatibles
Jeudi 18	14h15	Salle Conférences SIC	PolyPC — Groupe des utilisateurs d'IBM PC et compatibles
Mardi 23	16h00	Salle Conférences SIC	CI — Commission Informatique <i>M. le Prof. Koelbl, ☎ 021-693.27.81</i>

OFFRE SPÉCIALE

HIGHLAND DISQUETTES 5 1/4"

Double face – Double densité

Fr. 0,30/pièce

Offre valable dès le 28 avril 92 jusqu'à épuisement du stock

Presses polytechniques et universitaires romandes

EPFL-Centre Midi