

LA CONCEPTION DES SYSTÈMES REPARTIS

Nice, du 26 juin au 7 juillet 1978

COURS DE LA COMMISSION
DES COMMUNAUTÉS EUROPÉENNES

réalisé par

M. AMIRCHAHY et D. NÉEL

IRIA - SEFI / Formation

Édité par



INSTITUT DE RECHERCHE D'INFORMATIQUE ET D'AUTOMATIQUE

B.P. 105 - 78150 LE CHESNAY - Tél.: 954 90 20

LA CONCEPTION DES SYSTÈMES REPARTIS

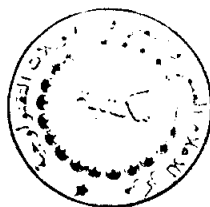
Nice, du 26 juin au 7 juillet 1978

COURS DE LA COMMISSION
DES COMMUNAUTÉS EUROPÉENNES

réalisé par

M. AMIRCHAHY et D. NÉEL

IRIA - SEFI / Formation



Édité par

IRIA

INSTITUT DE RECHERCHE D'INFORMATIQUE ET D'AUTOMATIQUE

B.P. 105 - 78150 LE CHESNAY — Tél.: 954 90 20

Édité par l'Institut de Recherche d'Informatique et d'Automatique

I.S.B.N. 2 - 7261 - 0213 - 1

Dépôt légal 300879/500

0724
0724

LA CONCEPTION DES SYSTEMES REPARTIS

Cours réalisé avec la coopération de :

CCE-CREST	Commission des Communautés Européennes Comité de la Recherche Scientifique et Technique
IRIA	Institut de Recherche d'Informatique et d'Automatique
NICE	Faculté des Sciences

Directeur : G. LE LANN

Conférenciers :

- E. HOLLER
- E.D. JENSEN
- D.B. MacQUEEN
- E.G. MANNING
- G. MAZARE
- R.W. PEEBLES
- C. WHITBY-STREVENS

Organisation et coordination :

IRIA - SEFI/Formation

Réalisation : M. AMIRCHAHY et D. NEEL

Secrétariat : O. GUILLON et G. PEREZ

BIBLIOTHEQUE DU CERIST

PREFACE

Ce cours qui était placé sous le patronage de la Commission des Communautés Européennes et organisé par l'IRIA-SEFI/Formation a réuni pendant deux semaines huit conférenciers et une centaine d'auditeurs venant de quinze pays. Le présent document rassemble les textes originaux écrits par les conférenciers à l'occasion de ce cours.

Les systèmes informatiques constitués de plusieurs processeurs physiques autonomes, interconnectés et travaillant de façon asynchrone à l'accomplissement de tâches communes posent des problèmes de conception et de réalisation relativement nouveaux en Informatique. Ces problèmes, tant sous leurs aspects formels que pratiques, ont été examinés par chacun des conférenciers à la lumière des réalisations expérimentales et des derniers résultats de recherche que l'on a pu comparer et évaluer.

Elmar HOLLER, Responsable Scientifique au Kernforschungszentrum de Karlsruhe (RFA), a décrit le système expérimental multiminiprocesseur DISCO qui permet de gérer des fichiers répartis sur plusieurs processeurs. Il a présenté la partie "Allocation de ressources" en comparant la technique développée au Kernforschungszentrum avec celle en cours de réalisation dans le projet SIRIUS de l'IRIA.

Douglas JENSEN, Senior Principal Research Engineer à Honeywell Inc., Minneapolis (USA) a traité principalement des problèmes d'architecture de systèmes répartis (taxonomie, mécanismes d'allocation de bus, transactions interprocesseurs). Il a présenté le système expérimental HXDP et s'est fait l'avocat du "tout-matériel".

Gérard LE LANN, Ingénieur de Recherche à l'IRIA (France), a analysé les aspects fondamentaux des systèmes répartis en étudiant les conséquences des hypothèses faites sur les délais de propagation, en particulier pour ce qui concerne les techniques permettant d'assurer le contrôle dans les systèmes répartis. Plusieurs de ces techniques, reflétant l'état de l'art actuel, ont été étudiées et commentées.

David MacQUEEN, Senior Researcher, University of Edinburgh (GB), a proposé différents modèles de description de réseaux de processus ainsi qu'un langage de haut niveau permettant d'exprimer les liens de coopération existant entre processus. Il a montré comment appliquer aux réseaux informatiques certains résultats de la théorie des graphes.

Eric MANNING, Director of CCNG, University of Waterloo (Canada), a assuré l'ouverture du cours par la présentation des objectifs des systèmes répartis à travers quelques études de cas (DCS, Citibank, Mininet, CM*).

Guy MAZARE, Ingénieur au Centre Scientifique CII-HB de Grenoble (France), s'est attaché à décrire les problèmes de détection et d'expression du parallélisme dans les programmes. Deux autres approches ont été développées par D. MacQUEEN et C. WHITBY-STREVENS.

Richard PEEBLES, Senior Engineer, DEC Research Center, Maynard (USA), a examiné les problèmes soulevés par les systèmes de bases de données réparties dont les contraintes sont sans doute parmi les plus difficiles à satisfaire (contraintes de cohérence des données et de temps de réponse). Il a terminé par la présentation du projet expérimental en cours de réalisation chez DEC.

Colin WHITBY-STREVENS, Lecturer, University of Warwick, Coventry (GB), a présenté le langage EPL, basé sur la notion d'acteur et destiné à permettre l'expression des structures de processus parallèles. Il a parlé en outre des problèmes d'assignation automatique d'objets dans les systèmes répartis.

La qualité des exposés et les échanges entre conférenciers et participants ont donné à ce cours efficace une note attrayante. Les auditeurs ont pu acquérir une vision d'ensemble approfondie des travaux actuellement entrepris dans ce domaine dont l'évolution est rapide. Pour beaucoup d'entre eux, en particulier pour les auditeurs en provenance des pays de la Communauté Européenne, ce fut également l'occasion d'établir des liens tant professionnels qu'amicaux.

Ce document, qui est tout d'abord un support de cours, présente un panorama des techniques les plus récentes destinées aux systèmes informatiques répartis.

G. LE LANN



CONTENTS

(E. HOLLER)	ALLOCATION OF DISTRIBUTED RESOURCES	1
1.	Introduction	2
2.	The basic coordination protocol	4
3.	Implementation of controllers	9
3.1.	Conventional approach	9
3.2.	The secretary concept	11
4.	Resiliency properties	16
5.	Conclusion	20
	References	21
	DISCO : A DISTRIBUTED DATA BASE SYSTEM BASED ON LOGICAL FILES	23
1.	Introduction	24
2.	Determination of a suitable concept	24
3.	Distributed file system architecture	30
4.	Operational model of the distributed file system	34
5.	Conclusion	38
	References	39
(E.D. JENSEN)	DECENTRALIZED CONTROL	41
	References	53
	INTERPROCESSOR COMMUNICATION TRANSACTIONS	55
1.	Introduction	56
2.	Major transaction decisions	57
2.1.	Which processor initiates a transaction	57
2.2.	The number of destinations each message transmission may have	59
2.3.	Number of messages per transaction	60
2.4.	Types of responses to messages	60
2.5.	Timing of responses to messages	61
2.6.	Response content	62
2.7.	Which processor terminates the transaction	63
	References	63
	BINDING OF NAMES TO MESSAGES, SOURCES, AND DESTINATIONS	65
	References	71

BUS ALLOCATION -- A TAXONOMY 73

PARTITIONING AND ASSIGNMENT OF DISTRIBUTED
PROCESSING SOFTWARE 79

1.	Introduction	80
2.	The role of the programmer	81
3.	The role of the translator	84
4.	The role of the configurator	85
5.	The role of the assigner	88
6.	The role of the dispatcher	88
7.	The role of the processor	91
8.	Conclusion	91
	References	92

THE HONEYWELL EXPERIMENTAL DISTRIBUTED PROCESSOR
AN OVERVIEW 95

1.	Motivations for distributed computers	96
1.1.	Extensibility	96
1.2.	Integrity	97
2.	Research approach	100
3.	Application characteristics and requirements	100
4.	The HXDP philosophy	102
5.	HXDP Hardware architecture	104
5.1.	Physical structure	105
5.2.	Logical structure	106
5.3.	Errors	107
6.	Status and conclusion	109
	Acknowledgements	109
	References	110

(G. LE LANN)

	SOME FUNDAMENTAL ISSUES IN DISTRIBUTED PROCESSING	113
1.	Definitions	114
2.	System properties	115
3.	System classification	116
4.	Basic characteristics and principles	116
5.	What is new regarding control ?	118
	References	119

	AN OVERVIEW OF DISTRIBUTED CONTROL TECHNIQUES	121
1.	Introduction	122
2.	Objectives of distributed synchronization techniques	124
2.1.	Redundant computing	124
2.2.	Partitioned computing	124
3.	A classification of some current techniques	126
3.1.	Utilization of physical clocks	126
3.2.	Explicit utilization of control privileges	127
3.3.	Utilization of counters	128
3.4.	Utilization of sequencers	128
4.	An example	129
5.	Evaluation criteria	132
6.	Conclusion	135
	References	136
(D.B. MacQUEEN)	MODELS FOR DISTRIBUTED COMPUTING	139
1.	Introduction	140
2.	Theoretical issues	140
2.1.	Virtualization	141
2.2.	Nature of computing agents	141
2.3.	Communication	141
2.4.	Structure of systems	142
2.5.	Nondeterminism	143
2.6.	Language design	143
3.	Traditional models	144
3.1.	Automata models	144
3.2.	PETRI nets	144
3.3.	Operating system theory	145
4.	The actor model	146
4.1.	Actors, messages, and events	146
4.2.	Computations as event orderings	150
4.3.	Fork and join behavior	153
4.4.	More on subcomputations	154
4.5.	Nondeterminacy	155
4.6.	Summary	155
5.	Communicating behaviors	155
5.1.	Ports, labels, and sorts	156
5.2.	Expressing elementary behaviors	156
5.3.	Nets and compound behaviors	159

	5.4.	Algebra of behaviors	163
	5.5.	Remarks	164
	6.	Stream processing systems	165
	6.1.	Process networks	165
	6.2.	A denotational point of view	166
	6.3.	Semantics	170
	6.4.	Nondeterminacy	171
		Bibliography	172
(E.G. MANNING)		DISTRIBUTED DATA PROCESSING	175
	1.	Introduction	176
	1.1.	What is it ?	176
	1.2.	Motivations	177
	1.3.	Summary	183
	1.4.	References for chapter 1	183
(G. MAZARE)		PARALLEL AND DISTRIBUTED SOFTWARE	185
	1.	Introduction	186
	2.	Parallelism necessity and existence	186
	2.1.	Parallelism or functional distribution	186
	2.2.	Parallelism existence	187
	3.	Parallelism detection techniques	187
	3.1.	Independent instructions	187
	3.2.	Parallel evaluation of arithmetic expressions	188
	3.3.	Loops	189
	3.4.	Amount of parallelism detected in this way	189
	3.5.	Conclusion	189
	4.	Expression of parallelism and distribution	190
	4.1.	Background	190
	4.2.	PASTORAL : a parallel programming language	191
	4.3.	Functional distribution : some existing tools	193
	4.4.	Future trends	194
	5.	References	194
(R.W. PEEBLES)		DISTRIBUTED DATA MANAGEMENT : AN INTRODUCTION	197
	1.	Introduction	198
	2.	Distributed system structure	199
	2.1.	Physical networks	199
	2.2.	Amorphous distributed systems	200
	2.3.	Logical networks	201

2.4.	Combined structures	203
3.	Data management in homogeneous environments	203
3.1.	Data structures	203
3.2.	Mapping information structures to distributed storage structures	204
3.3.	Concurrent access control	206
3.4.	Operator implementation strategies	207
3.5.	Security	211
3.6.	Fault tolerance	212
3.7.	Operating system implications	213
4.	Heterogeneous systems	214
5.	Summary	215
	References	216

(C. WHITBY-STREVS)	DISTRIBUTED COMPUTING : COMPUTER SCIENCE REVISITED	219
1.	Prologue	220
2.	Introduction	220
3.	Computer architecture	220
4.	Programming techniques	220
4.1.	Algorithms	221
4.2.	Systems analysis	221
4.3.	Automatic detection of parallelism	221
4.4.	Concurrent sequential process	221
4.5.	Operating system structures	222
4.6.	Network protocols	222
4.7.	Efficiency	222
4.8.	The LSI revolution	223
4.9.	What is left ?	223

	A "NESTED PARALLELISM" APPROACH TO DISTRIBUTED COMPUTING	225
1.	Introduction	226
2.	Existing methods	226
3.	New techniques	227
4.	The concept of process	227
5.	Data abstraction	228
6.	EPL - An experimental language for distributed computing	229
6.1.	Acts and actors	229

6.2.	Messages	229
6.3.	Procedures	230
6.4.	Multiple values	231
6.5.	General remarks	231
6.6.	Examples of the use of EPL and of 'nested parallelism'	232
6.7.	A distributed implementation of EPL	235
	References	235
	PROBABILISTIC PERFORMANCE MODELLING	237
	DYNAMIC OBJECT ASSIGNMENT	243
1.	Introduction	244
2.	Aims of dynamic object assignment	245
3.	Problems of dynamic object assignment	245
4.	Case studies	247
5.	Optional file allocation in a multiple computer system	247
6.	Probabilistic modelling of file allocation .	247
7.	Models for dynamic load balancing in a heterogeneous multiple processing system	247
8.	The Edinburgh distributed domain system	248
9.	Conclusion	250
	References	251

BIBLIOTHEQUE DU CERIST



ALLOCATION OF DISTRIBUTED RESOURCES

Elmar Holler

Kernforschungszentrum Karlsruhe

Postfach 3640

7500 Karlsruhe Bundesrepublik Deutschland

Mechanisms providing for the allocation of distributed resources are a basic requirement for distributed database systems and distributed process control. In cases where decentralization of control is required because of strong availability and security demands, resource allocation may be achieved by means of distributed resource allocation controllers, intercommunicating on the basis of specialized high level protocols. In this paper an example of a resource allocation protocol, allowing for resilient control structures, will be presented.

Les mécanismes qui assurent l'allocation des ressources réparties sont nécessaires aux systèmes de bases de données réparties et de contrôle de processus. Dans le cas où la décentralisation du contrôle est exigée pour améliorer la disponibilité et la sécurité, l'allocation des ressources est réalisée au moyen de contrôleurs d'allocation de ressources répartis qui communiquent suivant des protocoles spécialisés. Cet article présente un exemple de protocole d'allocation de ressources qui conduit à des structures de contrôle tolérant les défaillances.

1. INTRODUCTION

Distributed applications require mechanisms for allocation of distributed resources in all cases, where user processes need coordinated access to such resources, like e.g.

- in a distributed data base environment where certain transactions need exclusive access to a set of distributed data base components, in order to provide for the consistency of stored and retrieved information
- in a process control environment where complex technical processes require coordinated activation of peripherals distributed over several realtime computers e.g. when transferring the proper set of NC programs to all tool controllers involved in a special production process to be started.

Solutions to this problem are sketched in fig.1:

Conventionally a centralized resource allocation controller, C in fig. 1a, coordinates requests of user processes for exclusive use of subsets of distributed resources.

A decentralized approach would install several controllers which coordinate user requests by means of communicating according to a special protocol. Once a mutual agreement on the request to be honored is achieved, the subset of resources in question is allocated to the requesting user process as shown in fig. 1b.

Protocols for decentralized allocation of distributed resources are based on interprocess communication [ABS75] and allow the synchronization of parallel processes in a distributed system with respect to their resource usage despite the absence of a centralized communication area.

The number of known approaches tackling this problem is limited. The paper will take up solution presented in [Hol74], the so called "basic coordination protocol", which has been extended to provide for fault tolerance and recovery [Dro76]. Another solution showing resiliency properties was proposed by Le Lann [LeL77]. It uses the concept of a token traveling on a "virtual ring" of controllers.

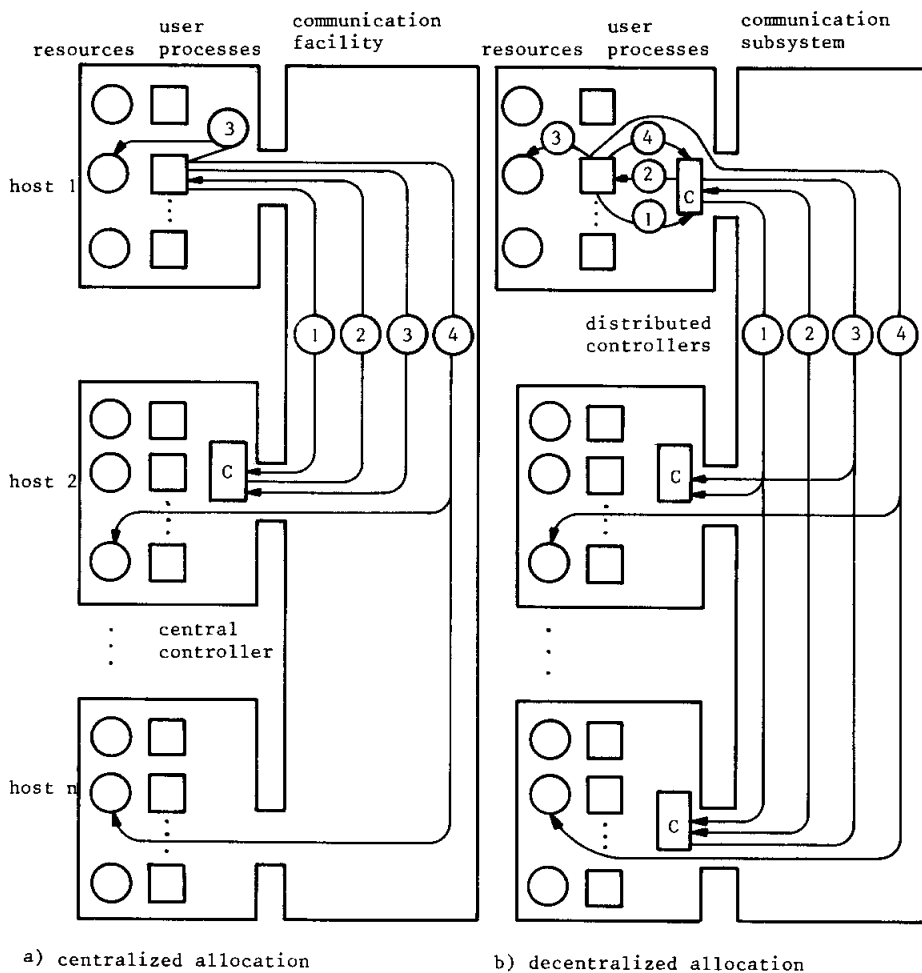


Fig. 1: Centralized versus decentralized allocation of distributed resources. Interactions of processes and controllers are labeled to indicate ① request, ② grant, ③ use and ④ deallocation.