分散處理 아키텍쳐

宋 榮 宰

慶熙大學校 電子工學科 助敎授(工博)

1. 序 論

지금까지의 데이터處理技術은 汎用大型計算機에 의한 集中處理를 중심으로 발전하여 왔다.그 主要한 이유는 高性能의 데이터處理能力을 확보하는 일이 중요하였고, 그것을 실현하기 위한 가장 좋은 방법은 性能/價格比가 우수한 汎用大型計算機를 效率좋게 사용하는 것이라고 생각하였기 때문이다.

그러나, 第四世代의 計算機技術을 생각하는 단계와 와서 여러 가지 입장으로 부터 集中處理에 대한 의문이 생겨나서 分散處理(distributed processing) 시스템의 모색이 개시되어 지금은 分散處理가 다음 世代의 대표적인 處理形態가 되리라 예상된다. 分散處理를 指向한 계산기의 商品化는 小形機의 分野로 부터 침투하고 있으나, 지금까지의 계산기 기술의 中心 思想에 대한 大變革이라 말할 수 있다. 어쨌든 分散處理는 앞으로 당분간 情報處理技術을 代表하는 中心 課題의 하나임은 틀림없는 사실이다.

分散處理시스템이라고 부르는 것을 크게 보면

1) 計算機 室內에서 시스템 資源이 가장 적합하게 分散化를 시도한 것

2) 通信네트워크에 의하여 지역적으로 시스템 資源을 가장 적합하게 分散化를 시도한 것

의 2種類가 있다.[1] 本稿에서는 前者를 分散形머시인, 後者를 分散形(計算機)네트워크라고 부르도록 한다. 分散形머시인은 종래 1臺의 프로세서로서 구성된 계산기를 여러 臺의 프로세서로 구성함에 의하여 시스템을 효율적으로 構造化시킨 것이다. 여기에는 각 구성 프로세서에 응용 업무의 負荷를 均等하게 分散시키고저 하는 負荷分散形의 것과, 각 구성 프로세서를 실행機能別로 專用化하여 응용업무의 특정 기능을 각 프로세서에 分散시키도록 하는 機能分散形이 있다.[2] 한편, 分散形네트워크(network)에는 그 크기의 大小에 따라서 狹域네트워크 (예를 들면 同一敷地內), 廣域네트워크(지역적으로 흩어진 것)等이 있으나 시스템 構造의 관점으로 보면 각 構成 프로세서(計算機)를 終形으로 배치하는 것과 橫形으로 배치하는 것 等으로 크게 나눌 수 있다.[3] 이 경우에, 시스템을 이용하는 사업체의 組織構成, 業務構造에 적합한 시스템을 구성함에 의하여 效率을 올릴 수 있다. 이들 시스템에 대하여는 다음 절에서 상세히 논하기로 한다.

2. 分散處理의 背景

分散處理를 오늘 날 중요시 하고 있는 要因을 최근의 모든 計算機 技術을 고려하여 정리하여 보면 다음과 같다.[4][5]

(1) 從來부터 集中處理를 유지해 왔던 Grosch 法則 즉, "計算機의 throughput는 그 價格의 2乘에 비례한다"고 하는 것이 성립하지 않게 되었다. 이 법칙은 1950 ~ 1970年의 시대에 근사적으로 성립하였다고 말할 수 있으나, 최근의

LSI 기술등의 경이적인 진보에 의하여 데이터處理 가격의 低下가 대량생산 위주의 小型計算機에 특히 현저하게 나타났고, 또한 大型計算機의 運營体制(operating system)가 너무 복잡하게 되는 등의 이유 때문에 이와 같은 상태변화가 일어났다. 이 결과로 데이터 處理 機能을 集中시킨 대형계산기가 가격/성능비가 반드시 유리하다고 말할 수 없게 되었고, 때로는 시스템을 分割하여 複數의 계산기를 사용하는 방법이 가격/성능비의 개선 방법이라고 생각하는 경우도 나타나고 있다.[6]

(2) 데이터 處理 가격의 현저한 低下에 비하여 通信回線에 의한 데이터 轉送 가격의 低下는 훨씬 적다. 따라서 시스템을 분할한다면, 分割된 데이터 處理 機能을 실제로 데이터가 발생하여서 이용되는 end user(말단 사용자) 부문에 배치하는 것이 통신코스트의 削減에 유리하다.[7]

(3) 인건비는 데이터 處理 가격의 저하와는 반대로 높이 올라가고 있다. 그 결과로 인건비를 종래 이상으로 유효하게 사용한다는 관점으로부터 다소 여분의 비용을 소비한다 해도 end user부문에 따라서 사용하기 쉬운 현장 업무용 시스템을 개발하는 것이 좋다.[7]

(4) 실제의 현장 업무는 여러 가지이고, 때로는 시간의 경과와 함께 변화할지도 모른다. 따라서 시스템은 이와 같은 현장의 요구의 多樣性/變化性에 견디는 고도의 융통성을 가져야 한다. 이 때문에 시스템은 명확한 階層構造/모듀울 構造를 가진다.

(5) 이와 같은 시스템의 構造化는 종래 극도로 복잡화 되었던 OS와 user 프로그램의 관계를 간소화하여 user 시스템 전체의 개발기간을 단축하여야 한다.[6]

(6) 分散處理 시스템에서는 通信回線등에 支障을 가져와도 user부문의 分散프로세서를 단독으로 동작 시키므로서 user 부문에 대한 서비스를 계속하여야 한다.

(7) 종래의 on-line 시스템의 확장과 함께 集中處理 方式으로서는 對處할 수 없는 시스템 throughput가 요구되는 경우가 생기고 있다. 특히 온라인 은행업무의 database(데이터의 불필요한 중복을 피하여 상호 연관이 있는 데이터

표 1. 集中處理와 分散處理의 利害損失

項目	集中處理	分散處理
機能	• 획일적인 모든 기능의 제공에 머물기 쉽다.	• 보다 좋은 end user 기능을 제공할 수 있다.
構成	• 시스템 구성상의 융통성에 한계 있다.	• 시스템 구성상의 융통성이 높다.
性能	• 性能面에 한계 있다.	• 시스템 throughput 및 응답성이 개선된다.
信賴性	• 신뢰성에 한계가 있으나 障害時의 대응기술이 확립되어 있다.	• 신뢰성의 향상이 기대되지만 보수 및 회복순서등의 검토필요
運用性	• 시스템 운용이 중앙에서 이루어지므로 비교적 쉽다.	• 사용자마다 end user 機器의 운용관리가 필요하다. 요원 교육에 배려 필요
價格	• 데이터處理 기능의 集中에 의하여 가격低減 기대.	• End user 위주의 中小形 각종장치의 가격저하에 기대
開發期間	• 대형 온라인 시스템에서는 3년이상 필요하다.	• 개발기간의 단축이 기대되지만 분산처리의 모든 기술이 확립되어야 한다.

를 모은 것)化는 그 전형적인 例로서, 이러한 경우에는 分散處理시스템이 좋다.[5],[6]

이상과 같이 分散處理는 여러 가지의 利點을 가지고 있으나 동시에 몇 가지의 난점이 있으므로 실제적인 배려가 필요하게 된다. 분산처리의 利害損失을 集中處理와 비교하여 표1에 정의한다.

3. 分散處理의 定義와 分類

일반적으로, 分散處理는 여러 臺의 프로세서를 포함시킨 計算機 시스템上에서 시스템內에 분산된 각종 資源을 가장 적합하게 이용하면서 응용프로그램을 실현시키는 計算機 技術의 총칭이다.

그러나, 分散處理라는 말은 매우 넓은 의미로 사용하고 있으나 사람마다 각기 다르다. 따라서 이를 체계적으로 定義할 필요가 있다.

우선, 분산처리를 정의하기 위하여는 그림 1과 같이 ① 무엇을 ② 어떻게 나눌까, ③ 왜 나눌까, 를 고찰하여 보면 좋으리라 생각된다.

- 무엇을? — Computation { Processors, Programs, Data } Processes
- 어떻게? { Geographical, Load, Functional }
- 왜? { Performance/cost, Extensibility, Integrity (Realiability, Availability) }

그림 1. 分散處理의 定義

- Modularization { Programs (Software), Data, Computation, Processors (Hardware) }
- Distribution { Transaction (Users), Data, Computation }
- Distribution의 레벨, 크기 { Instructions, Processes, Jobs }

그림 2. 分散處理를 위한 條件

또한, 분산처리가 성립하는 조건은 그림 2와 같이 각 요소의 모듀울化가 진행되어 transaction이 본질적으로 분산하고 있는 경우이다.

분산의 레벨로서는 命令, 處理 및 作業(jobs) 등의 각 요소를 생각하여야 하는데, 그것은 case by case로서 교환을 하면서 분산처리가 실시되게 된다. 실제로 실용화된 것을 분류하면, 地理的 分類와 非地理的 分類의 2가지로 나누어서 그림 3과 같이 정의할 수 있다.

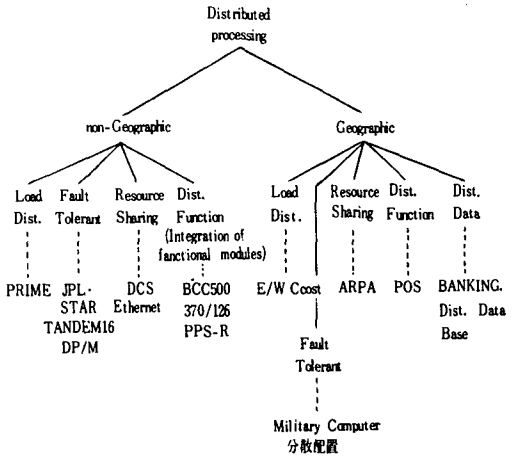


그림 3. 分散處理의 分類

4. 分散形머시인

여기에서는 分散處理시스템 가운데에서 分散形머시인(machine)을 들어서, 그 특징과 분류, 實例, 그리고 문제점등에 대하여 설명한다.

4 - 1. 分散形머시인의 考慮點

반도체 기술의 발달로 인하여 小形機의 가격/성능비가 大形機보다 훨씬 우수하다면, 小形機를 여러 臺 사용하여 大形機를 구성하고저 하는 發想이 생긴 것은 당연하다. 그러므로 미니컴퓨터나 마이크로컴퓨터를 사용한 分散形시스템의 연구가 시작되었으나 우선 이들 시스템에 있어서 고려해야 될 점을 들어 보면,[9] [10]

① OS에 대한 多樣프로세서의 制御方法
② User job에 대한 프로세서의 割當方法
③ 프로세서, 主記憶등의 모듀울 結合方式
④ 프로세서 사이의 通信 및 通信方式
⑤ 시스템 資源의 平均的 使用과 競合(conflict)의 制御方法

⑥ 시스템의 信賴性 확보방법

⑦ 시스템의 확장성 제공의 방법 등이다.

4 - 2. 分散形머시인의 分類

일반적으로 여러 관점으로 부터 분류할 수 있지만 여기에서는 앞서의 ①②의 고려점을 중시하여 다음과 같이 크게 분류한다.

1) 負荷分散形 : 각 프로세서가 同一機能을 가지고 있어서 user job나 OS가 똑같이 어느 프로세서에서도 走行할 수 있는 形態의 시스템

2) 機能分散形 : 각 프로세서가 각각 특정의 計算機 機能을 實行하도록 專用化 되어 있는 형태의 시스템

前者는 大形機에 있어서 이른바 密結合形(tightly-coupled type)小數 멀티프로세서 (보통 1~4臺)의 思想을 이어 받은 것으로 응용업무의 負荷가 多數(예를 들면 10臺)의 프로세서에 똑같이 分散시키기 때문에 負荷分散形이라 부른다.[10] 한편, 後者는 계산기의 모든 기능이 특정 機能 專用의 프로세서에 분산되므로 보통 機能分散形이라 부른다.[2]

4 - 3. 負荷分散形머시인

이런 종류의 계산기로서 유명한 것은 1971년에 연구 개발한 미국의 Carnegie-Mellon 大學의 C.mmp 시스템이다.[12] [13]

이 multi-processor는 다음과 같은 특징을 가지고 있다.

1) 16臺까지의 프로세서가 共有메모리에 액세스하는 密結合멀티프로세서이다.

2) 시스템에 부여된 부하를 複數의 프로세서로서 分散處理하는 負荷分散形머시인이다.

3) 프로세서로서는 미니컴퓨터의 프로세서를 사용하고 있다.

C.mmp의 設計 目的은 다음과 같다.[12]

1) 複數의 프로세서가 共有메모리에 액세스하는 形式의 멀티프로세서技術을 개발하는 일

2) 멀티프로세서에 대하여 가능한 많은 것을 배우는 일.

1971년 당시에는 멀티프로세서의 과학적 성질, 성능, 문제점이 거의 알려지지 않았으나 실제로 만들므로서 이들이 명확하게 되었다. 1977년에는 시스템이 거의 완성상태에 이르렀다. 현재까지 판명된 멀티프로세서의 利點은 다음의 4가지이다.[14], [15]

1) 시스템의 처리능력, 특히 throughput 의 向上

1臺의 프로세서의 能力으로는 限界가 있으나, 이것을 여러 臺 竝列로 動作시킴으로서 處理能力을 向上시킬 수 있다.

2) 性能/價格比의 向上

비교적 싼 프로세서를 여러 臺 사용하여 요구하는 性能을 실현할 수 있다.

3) 信賴性의 向上

똑 같은 종류의 프로세서가 여러 臺 있으므로 어느 것이 고장했을 경우에도 다른 프로세서로 代用할 수 있다.

4) 性能의 증가가 容易

시스템에 요구되는 처리능력이 시스템의 도입시정에는 不明하기도 하고, 시스템 運用時에 성능을 증가하고 싶은 경우에 프로세서의 臺數를 증가하여 점차 시스템을 확장할 수 있다.

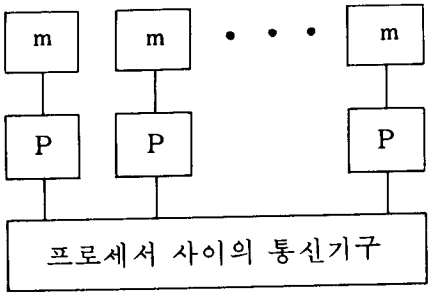
이와 같은 利點을 실현 시키기 위하여 C.mmp에서는 Hydra 라고 하는 OS를 개발하고 있다. 최근에 카네기·멜론대학에서는 C.mmp의 경험을 살려서 개별 主메모리를 가진 多數의 마이크로프로세서에 의한 Cm* 시스템[13]의 개발에 연구를 계속하고 있다.

4 - 4 機能分散形머시인[1],[2]

機能分散形 시스템의 기본 구성은 그림4와 같이 機能을 分擔한 프로세서가 프로세서 사이의 通信/結合機構를 통하여 相互 연락을 취하면서 독립된 처리를 하며, 전체로서 하나의 계산기시

스템을 구성하는 것으로서 利用目的에 따라서 機能의 分割과 시스템의 구성이 이루어진다.

이런 종류의 計算機로서 최초의 본격적인 시스템이라고 말할 수 있는 것은 1969년경에 개발이 시작된 미국의 BCC(berkeley computer



M : 메모리 P : 프로세서

그림 4. 機能分散시스템

corp)社의 BCC-500이다. 이 시스템은 상업적으로 실패하여 그 후에 하와이대학에서 1臺 가동하고 있으나, scheduling, 入出力制御, 보조메모리制御, 通信制御를 관리하는 專用프로세서를 포함하고 있다. 또 하나의 例로서는 Fair-child社의 SYMBOL계산기[16]도 高級言語를 직접 실행하는 機能分散形 計算機로서 유명하다. 또한 IBM 370/115,125등도 기능분산화의 경향이 강하다.

이들 시스템에서는 각 구성프로세서는 각각 特定 機能에 專用化시키므로 프로세서 機能과 그 實行 機能사이의 부적합한 사태를 피하기 쉽다. 따라서 가격성능 특성을 改善하기 쉽고, 商用機로서도 기능분산형이 비교적 성공하고 있다. 그러나 다음과 같은 문제점이 있다.[17]

① 각 프로세서의 기능 割當이 적절하지 않으면 job 實行時에 프로세서 사이의 通信의 over-head가 커지게 된다.

② 각 프로세서의 선택방식에 따라서 機能別 프로세서 사이에서의 負荷의 不均衡이 생긴다.

③ 構成프로세서의 專用化 技術의 사용 경험이 적으므로 충분히 專用프로세서로서의 가격/성능비를 발휘할 수 없을지도 모른다.

특히 처음의 것은 중요하여서, 이와 같은 시스템에서의 만일 각 프로세서 사이의 通信回線數가 많아지도록 기능을 할당하면 아무리 結合 인터페이스가 高速이라도 性能上의 커다란 문제를 가져오게 된다. 다음은 시스템內의 각 프로세서를 똑같은 종류로 할 것인가, 각 기능의 특수성에 맞추어서 서로 다른 종류로 할 것인가의 문제가 있다. 각 構成프로세서에 걸리는 기능負荷의 크기는 일반적으로 實行 job의 성질에 크게 의존하기 때문에 프로세서 사이의 負荷가 어느 정도 불균형하게 되는 것은 어쩔 수 없다. 그러나 각 프로세서에의 기능할당은 할당된 각 기능이 가능한 독립되고, 각 프로세서 사이에서의 부하의 균형이 어느 정도 성립하여야 한다.

5. 分散形네트워크

지역적으로 흩어진 分散處理시스템, 즉 分散形(計算機)네트워크에 대하여 그 특징과 분류에 대하여 논하고저 한다.

5-1. 分散形네트워크의 특징

최근에는 계산기를 사용하는 데에 멀리 떨어진 곳으로부터 端末을 거쳐서 중앙의 계산기에 액세스하는 on-line 이용이 널리 일반화되어 있다. 그 결과로 계산기의 user는 중앙의 계산기를 관리하는 데이터處理 부분에서 떨어져 각 부문에 散在하고 있는 것이 보통이다. 1967년 이래 미국의 ARPA(advanced research project agency)[18] 네트워크의 연구에 이어 계산기 네트워크 기술에 관한 연구가 계속되고 있으나, 分散形네트워크에서는 이들의 새로운 기술을 이용하여 지역적으로 散在하는 多數의 user에 대하여 종래의 온라인 시스템 이상으로 機能性能面에서 한층 효과적인 데이터 처리 서비스를 제공하려 하고 있다.

일반적으로 분산형네트워크에서는 user의 조

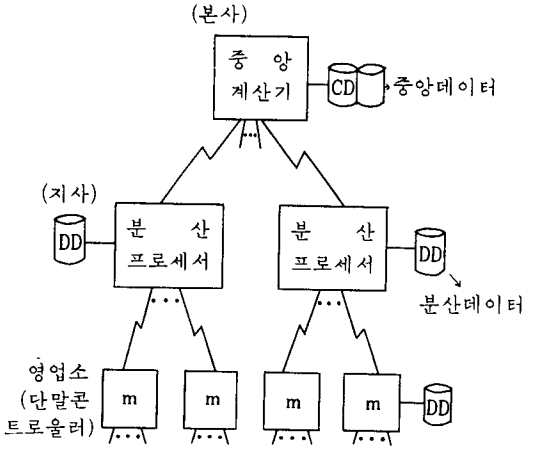
직 및 업무의 전체 구조에 가장 적합한 시스템의 구조가 追求된다. 이 결과로 각 user 부문에 실무적인 遠隔計算機를 배치하여 이들을 중앙의 데이터 處理部門의 중앙계산기와 通信結合하는 경우가 많다. 앞서와 같이 싸고, 가격성능비가 우수한 소형기가 입수 가능한 현재, 대량의 업무 데이터를 발생, 사용하는 user부문에 원격계산기를 도입하여 실무의 수행효율을 올리는 것은 폭등하는 인건비와 그다지 싸지않은 通信回線코스트에 앞으로 잘 대처하여 갈 유효한 방법이라 말하고 있다.

5 - 2. 分散形 네트워크의 分類

分散形 네트워크로서 구성되는 계산끝시스템은 다음과 같이 크게 분류한다.[5],[8]

1) 獨立形 分散시스템
2) 縱形 分散시스템(垂直形, 階層形)
3) 橫形 分散식스템(水平形)
4) 縱橫形 分散시스템(複合形)

獨立形 分散시스템이란 組織內의 각 user 부문



M : 미니컴퓨터, T : 단 말

그림 5. 縱形 分散시스템의 一例

에서 서로 독립적으로 小型計算機 (미니컴퓨터, office 컴퓨터, 개인용 계산기등)를 이용하는 형태이다. 이것에 대하여 縱形 分散시스템은 그림 5에 표시한 바와 같이 말단 user 위주의 特定機能을 가진 각종 데이터 處理裝置를 각 user 부문에 배치하고, 이것을 조직 및 업무의 구조에 맞추어서 通信手段으로 統合하는 것이다. 전체로서는 user부문에서의 會話形 處理를 主體로 한 현장업무의 실행과 本社 데이터처리 부문에서의 batch처리를 주체로 한 經營情報의 處理(data-base 처리等)을 가능하게 한다. 그리고, 橫形 分散處理시스템은 그림 6에 표시하는 바와 같이 각각 논리적으로 대등한 계산기 사이를 情報資

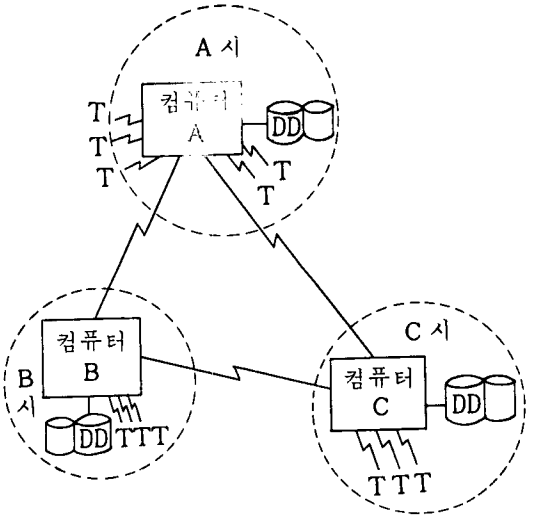


그림 6. 橫形 分散시스템의 一例

源等의 共用을 목적으로 하여 대등한 관계에서 通信結合하는 것이다. 끝으로 縱橫形 分散시스템은 複數의 縱形시스템을 橫形에 결합한 것이다.

6. 結 論

以上, 최근 각 방면에서 화제가 되고 있는 分散處理시스템에 대하여 간단히 現狀을 중심으로 설명하였으나, 앞으로 分散化의 경향은 더욱 증대되리라 생각된다. 또한 分散處理시스템을 용이하게 실현하기 위하여는 電話를 거는 것과 같이 간단히 쓸 수 있는 네트워크가 필요하게 되고 인터페이스의 표준화가 시급하다. 그리고 앞으로 현재 연구되고 있는 分散處理 가운데에서

가장 성행되리라 생각되는 것은 multi-microprocessors 시스템에 의한 機能分散 處理가 되리라 확신한다. 끝으로, 本稿가 分散處理를 이해하는 데에 조금이라도 도움이 되었으면 筆者는 多幸으로 생각하는 바이다.

參 考 文 獻

1. E. C. Joseph, "Innovation in Heterogeneous and Homogeneous Distributed-Function Architecture", Computer, Vol. 7, No. 3, pp. 17-24 (1974).
2. 相磯秀夫, "機能分散形計算機システム, 情報處理, Vol. 18, No. 4, pp. 325-333 (1977).
3. 八木 曉, "分散處理システム構築のたぬの 技術とえの現狀, 日經エレクトロニクス別册「コンピユータ」, 1979-1980, pp. 60-76.
4. Bruce Gladston, et al., "Distributed Processing slashes development system's cost", Electronics, pp. 89-94, August 17, (1978).
5. A. 1. Scherr, "Distributed Data Processing, IBM Syst. J., Vol. 17, No. 4, pp. 324-343 (1978)
6. J. D. Foster, "The development of a concept for distributive processing,", Proc. Compcon Spring, pp. 28 (1976).
7. F. V. Wangner, "Is decentralization inevitable?", Datamation, Vol. 22, No. 11, pp. 86 (1976).
8. G. M. Booth, "Distributed Information Systems", Proc. NCC, pp. 789 (1976).
9. 飯塚肇, "マイクロセツサ複合体の技術", 電子通信學會誌, Vol. 60. No. 2, pp. 125-135 (1977).
10. 村上國男, "ユンピュータ アーキテクチヤの最近の進步, 電子通信學會誌, Vol. 60, No. 6, pp. 669-680 (1977).
11. Y. C. Chow, et al., "Models for dynamic Load Balancing in a Heterogeneous Multiple Processor System", IEEE Trans. Computers, Vol. 18, No. 5, pp. 354-361 (1979).
12. W. Wulf, et al., "C. mmp-A Multi-Mini Processor", FJCC, Vol. 41, pp. 765-777 (1972).
13. D. P. Siewiorkek, et al., "A Case study of C. mmp, Cm*, and C. vmp: Part 1-Experiences with Fault Tolerance in Multiprocessor Systems", Proc. IEEE, Vol. 66, No. 10, pp. 1178-1199.
14. G. Reyling, "Performance and Control of Multiple Microprocessor Systems", Computer Design, Vol. 13, No. 3, pp. 81-86 (March 1974).
15. P. H. Enslow, "Multiprocessors and Parallel Processing" John Wiley & Sons, 1974.
16. Rex Rice, et al., "SYMBOL-A major departure from classic software dominated von Neumann computing systems, AFIPS SJCC, pp. 575-587 (1971).
17. R. N. Nilson, et al., "Distributed-Function Computer Architecture," Computer, Vol. 7, No. 3, pp. 15-35 (March 1974).
18. N. Abramson, et al., "Communication Networks," Prentice-Hall, pp. 537-554 (1973).
19. S. E. Scrupski, "Distributed processing grows as its hardware and software develop", Electronics, Vol. 49, No. 11, pp. 91 (1976).
20. V. G. Cerf, et al., "The future of computer communications", Datamation, Vol. 23, No. 5, pp. 105 (1977).