

墙的另一面——图灵模型更深层次的思考

李奕权¹ 郑利龙²

(中山大学软件学院 广州 510006)¹ (城市大学电子工程学系 香港)²

摘要 最近几年,计算机体系迈向多处理器结构道路。然而,冯诺曼机器主导的多核结构,令我们处在存储墙错误的一面。地址参数引入的冗余,降低了处理器的效率,成为图灵-冯诺曼模型的致命要害。对图灵模型作更深一层思考,以信息变换统一了冯诺曼机器程序变换和神经网络变换,分析了两种变换的异同及其优劣。提出以微核为基础,并按变换的成熟程度,向灵活的可编程的冯氏机器或高速的神经网络分化。模拟生物神经系统的进化,构建为人们服务的智能机器。2010年9月15日,美国波士顿的高性能嵌入式计算(HPEC)研讨会上,耶鲁大学的欧亨尼奥·卡鲁塞伊罗教授发表了一个基于人类视觉系统的高性能计算机“神经流”(NeuFlow),其体系结构利用了与本文的仿生电脑十分相似的概念。

关键词 仿生电脑,体系结构,图灵机,冯诺曼机器,神经网络,信息变换

Another Side of the Wall——Deeper Thinking in Turing Model

LEE Yick Kuen CHENG Lee Lung

(School of Software, Sun Yat-sen University, Guangzhou 510006, China)¹ (Electronic Engineering Dept., City University of HK, HK, China)²

Abstract Computer has entered into the era of multi-processor structure. However, multicore puts us on the wrong side of the wall. The memory address parameters introduce extra-redundant, decrease system efficiency. This paper proposed a new kind of model by considering intelligent processes as a transformation on information. This model unifies the Von-Neumann machine program and neural network transformation. The paper analysed the similarities and differences of these two process and proposed a computer system based on micro-kernel architecture and diverted into Von-Neumann or neural transform according to the maturity and speed requirement. Imitating biological organism intellect, this kind of computer, serves as a tool for human beings. Professor Eugenio Culurciello of Yale, with Yann LeCun of NYU, presented a high performance Embedded computer, NeuFlow, on HPEC workshop 2010-9-15 in Boston, its system architecture concepts are quire similar to the Pseudo-Organic computer proposed in this paper.

Keywords Pseudo-organic computer, System architecture, Turing machine, Von neumann machine, Neural network, Information transfer

1 引言

1.1 高性能计算进入P级和E级

随着IBM的Roadrunner、中国的天河一号、曙光6000星云系统的诞生,计算机世界已经进入千万亿次的P级领域,百亿亿次的E级计算已开始提上议程。普遍认为,2019年以前,将会出现成功的E级计算机。要进入E级计算,下面几个难题必须克服^[1]:

1. 冯诺曼瓶颈;
2. 存储墙;
3. 主频速度与能耗效率;
4. 电互连性能极限;
5. 容错管理;
6. 海量处理器的管理。

1.2 图灵模型的冗余影响效率

图灵-冯诺曼模型,用机器替代人类的思维,对信息进行

变换。模型基本假设之一,是把系统状态用字符串序列或线性地址存储的数据来表达^[2]。

模型的状态数据被假定是独立的,数据间没有定义任何关系,数据由线性地址随机指定。

图灵-冯诺曼模型中,数据内容的含义和数据间的关系,由程序员在编程时,在程序中隐性地或显性地表达。

冯诺曼机器初始状态的数据间本来存在某些关系,但这些数据由输入装置搬到记忆空间的过程中被忽略了,反而引入与状态无关的地址参数,原文为:“Numbers are assigned to the parts of the store in which the various packets of information are stored,”这是一个引入冗余的变换,增大了系统的自由度,使系统变得臃肿^[2]。

数据地址用来区分数据,与系统的状态没有关系,但却用作指标参与每一段变换。地址信息不得不在处理器和记忆系统间搬来搬去,效率低下,是造成各种技术难题的主因。

1.3 改进图灵模型

虽然线性地址的数据表达方式一直深入人心,可分离数据和程序,可以对通用的数学计算编程,但这不是智能机器必需的内容。图灵模型带来的技术墙阻挡系统进步。特别是进入 E 级计算,更大的系统意味着更高的墙。

怎样去建立新的模型呢?

生物大脑的智能给我们带来提示。

2 仿生电脑模型

我们制造智能机器,目的是对信息作变换,主要是用作预测未来。

生物通过感觉器官,测度外界环境的信息,经过大脑神经系统变换,预测未来环境变迁,作出适当反应,从而延续个体和种族的生命。

仿生电脑就是人造的智能机器,模拟生物的大脑,协助我们对信息作变换。

2.1 仿生信息论基础

信息是对环境系统的描述^[3],它减小了系统的不定性。信息用传感器采集,经过编码,用符号表达出来。

2.1.1 通信和计算

通信是信息从一个位置传送到另一位置的空间变换,计算是对信息从一种表达方式到另一方式的形式变换。

在信息的变换过程中,不失真变换是使信息量不会减小,其中一个判别标准是,变换后的信息是否可以还原成为原来的信息。

电子通信中,信息经过编码变成电信号,通常以空间某点上一连串电波的形式表达。能表达一个编码意义的电波串称为一组(packet)。

在每一组信息内,有时会加入冗余,检查错误,特别在通信和存储的时候。

图灵-冯诺曼模型中,信息以机器的状态,或者说是一组寄存器和记忆体的状态表达^[2]。

状态信号通常是并联的(如多个双稳电路),但也不排除是一组或多组串联信号的形式(例如在水银延迟线记忆中)。

2.1.2 静态和实时数据

图灵模型的程序把固定的初始状态逐步变换,等到变换终止,得到终止状态,变换只要能在有限的时间内完成,便算作完成一个作业,不管变换的时间有多长。

在现实环境中,有很多信息会随时间改变,计算机对初始状态数据作变换,需要在某一个时间段内完成,否则输出数据会失去时效,称为实时数据。图 1 给出静态和实时数据。

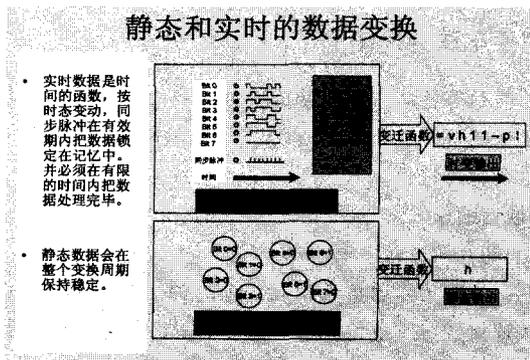


图 1 静态和实时数据

2.1.3 仿生电脑的信息

仿生电脑模型中,信息是指空间的多个点上,一连串的时变信号。能表达一个完整编码意义的多个信息串称为一组(packet)。

每组信息都是时间的函数,在一时段内保持不变的信息称为静态数据。

数码化的信息会有一个时钟周期,周期开始后,组内每个点都会在下一周期前的某个时间内达到数值为 0 或 1 的稳定状态,从这时间到下一周期,点内的信息可以看成静态数据,称为一个位数 Bit。

组内只有一个点,通过位数序列表达的信息称为串行信息,多个点共同表达的信息称为并行信息。

并行信息各个点,表达一个共同编码,点与点之间必定存在某些关系。数码化的串行信息,每组串行的 Bit 间也必定存在某些关系。这些关系在编码时已经确定。

正如图灵模型的数据一样,信息间的关系可以用依赖性、重用性、相似性、亲和性、一致性和生存性来表达^[4]。

2.2 仿生电脑对信息作变换

2.2.1 逻辑变换

最简单的信息变换是逻辑网络,读取输入端各点的信息后,在输出端产生与、或、及其组合的输出。

没有反馈的变换,只反映同一时钟周期内信息间的依赖性。各时钟周期间,信息互相独立。

反馈使后续的输出可以依赖前期信息,可以反映全组信息间的依赖关系。

反馈网络的输出,可能有发散、振荡和收敛几种状态。收敛是指给定任意的容许误差,在一定时间后,输出信息误差必会少于容许误差。

对于时变的输入信息,输出也是一组时变的信息。由于反馈的时延,输出常常会滞后多个时钟周期才能反映全组输入信息的影响。

2.2.2 记忆变换

记忆变换是一种特别的信息变换,一组或多组信息经过串行化,转换成一组以线性地址表达的字符所描述的静态数据,称为写入。时变的信息需定时取样,每个样本写在另一组地址上。

串行化时加入的地址是冗余的参数,同一信息可以对应多种串行化规律。除了线性序列外,地址被假定成独立的,写入数据时会失去原始信息各点间的关系信息。

为了保证写入数据不失真,通常会同时有一个对应的反变换,把串行化了的数据转回原来的形式,称为读出,反变换包含原始信息各点间的关系信息。

记忆是静态数据,写入后,数据不随时间变化,直到在同一地址上写入新的数据,旧数据才会消失。

某些硬件的数据在断电后会消失,这些硬件称为易失性存储器件,反之称为永久性存储器件。

数据从写入到消失的期间,称为生存期。

2.2.3 抽象变换

抽象变换是另一种特别的信息变换,多组信息很多时候会有相似的性质。抽象变换把这些相似性提取出来,寻求其中的不变量,用一组单独的信息表达。

抽象变换是减少信息量的变换,抽象的信息只表达多组数据的共性,不能还原成个别的信息组。

为了描述每组的全部信息,个别的信息组可以保留个性的变换。把抽象的共性和个性信息合起来,便可还原成原来的信息。

经过抽象变换,大量原始信息的共性可以用单独的信息组表达,连同每组的个性,表达信息需要的资源通常可以减少。实际上是对信息实行压缩。如果把个性信息抛弃,需要的资源便会更少。

抽象变换是生物进化,形成智能的一个重要机制,也是面向对象软件设计、数据挖掘和人工智能的基本手段。

抽象产生新的概念,概念又能同语言接合,多层抽象可以把复杂的世界用简洁的语言描述,使不同的个体互相沟通。

2.2.4 冯诺曼变换

冯诺曼机器把输入数据写入记忆体中,通过编程,描述输出数据和输入数据的关系,转成需要的输出数据,再输出到相关的设备中。

冯诺曼机器的数据,是指记忆体内的静态数据,数据间的关系在编程时要用数据结构去描述。

冯诺曼变换把初始信息分成程序和数据,通过编程,把程序员要求的变换方式存入记忆体,控制数据变换。编程使人可以通过语言控制机器变换。

结构化模式编程,设定数据结构,自上而下,最终分解成为顺序选择和循环3种结构运算的组合。

面向对象的编程,考虑整个变换的目标,把问题分解,让高内聚、低耦合的数据和变换方法结成类,在高层次处理数据的共性,数据的个性则让下层较小的对象去处理。

无论结构化或面向对象的编程,冯诺曼机器最终都是构建一个把输入转换成输出的变换函数。

2.2.5 多处理器和GPU的并行变换

早期冯诺曼机器把很多资源放在增长时钟的速率。后来发觉多个处理器并行运算可以加快数据处理,特别是对称的或不相关的数据集合。

多处理器的并行机和GPU正是为这类问题而设^[5]。多级存储结构更可减轻相关性带来的问题。但数据的相关性多式多样,很难有一个通用的结构模式。

这两年流行混合模式,带来了一些进展,但也不能彻底解决问题。

2.2.6 神经网络变换

神经网络是一个并行、分布的处理结构,它由处理单元及称为联接的信号通道互连而成^[6]。

神经元输入端点的信号,乘上一个反映该信号对下级信号相关程度的权值,信号间逻辑关系和权值决定处理单元是否触发,向下级单元发出0或1的信号。

和逻辑网络一样,神经网络的输出也可能有发散、振荡和收敛几种状态。也会有反馈带来的时延。

与图灵模型相反,神经网络用联接明确地表达信息间的关系,处理信息的速度和效益都很高,是生物大脑的主要部件,也是仿生电脑提高效益的目标。

2.2.7 专用信息变换

逻辑和神经网络都有极高的效率,早期也有人将其用来

制造计算机。但是网络中各级信号很难找到对应的物理意义,没有和机器沟通的语言,不容易设计和编程,只能为专门的变换定做,换代要改变硬件结构,不如冯诺曼机器方便。

只有成熟的信息变换,才能造成直接变换的逻辑器件,例如浮点处理器、声音处理器、图像处理、网络处理器等专用信息变换器。

硬件描述语言使软硬件的区别变得模糊,种种设计硬件的软件日趋成熟,使设计硬件变得更容易。

2.2.8 现场可编程逻辑变换

现代芯片技术,可以把逻辑阵列和记忆元件组合起来,构建各种粒度的可编程逻辑阵列FPGA。

可编程阵列拥有直接变换网络的效率,又可以随时编程,按需求定义变换函数,是一种很有前途的信息处理器件。

近日更有把FPGA与ARM处理器建立在同一片芯片的报告^[7]。

2.2.9 变换器的分化

冯诺曼机器可以进行通用信息变换。程序经串行化后,存储在磁盘、磁带、光碟等廉价的媒体上。使用时再读入,同一机器可配有多种程序。

仿生电脑包含通用的冯诺曼机器,也包含另一个极端的专用信息变换器,它只执行单一的变换。其牺牲了以通用性来换取更高的效率。

可编程逻辑阵列FPGA介乎二者之间,内置的动态内存或闪存控制开关阵列,保留器件的可塑性,也不会损失反应速度。

GPU处理器按特定的需要,组织多个处理器和多级缓存,用特定的通信网络连接,对指定类型的变换有较高的效率。

按结构命令分配的通信网络,灵活性比GPU更高,也可以按需要配置通用或专用的变换器,使粗粒度结构重组更具可塑性。

2.2.10 仿生电脑的构件

逻辑网络、记忆元件、抽象变换、冯诺曼变换、并行变换、神经网络、专用变换器件可编程逻辑器件等都是现代计算技术常用的信息变换器件。这些都被认为是基本构件,连同收集数据的传感器,用通信线路把它们组合起来,加上适当的管理,便构成能为我们服务的仿生电脑。

传感器、信息变换器和通信通道都是仿生电脑可用的资源。

3 仿生电脑的通信

通信是信息从一个位置传送到另一位置的空间变换,是信息变换的一种特殊形式,通常会伴随物理载体的转换,也涉及信号传递的时延。

3.1 通信通道

3.1.1 芯片间通信

同一片芯片内通常会用同一个时钟。在一个周期内,用金属线连接的点其电位会达到稳定平衡。可以将其当成单个信息,不算通信。

不同的芯片其片内有不同的时钟,信息从一个位置传送

到另一位置,便是信息作空间变换的通信。

芯片内一般用电平高低表达 0 和 1,但在通信的时候,会转换成不同的物理载体,例如不同的电平、差分信号、频差、相位差、时差和光信号等。

由于传播速度有限,一个芯片时钟周期内,信息传播的最远距离也有限。300MHz 为 1m,3GHz 为 100cm。超过这个距离,便会产生时延,特别是,要得到目的地的反应,最快也要这个时延的一倍。

若要增加物理通道的数量或增加信号的频率,只能增加信息的数量,不能减少时延。

信号传输,可以是单向的、点对点的、一对多的和多对多的。

高性能计算和多核计算,如果用点对点传输通信,线路数目会按处理器的数量指数性地激增。计算程序中,多对多的通信,是软件和硬件中的一个重要瓶颈。仿生电脑会模拟脑神经元,尽量使用一对多的通信。

3.1.2 仿生通信的逻辑通道

生物的脑神经连接可以动态地产生和死亡^[8],物理连接通道也就是逻辑通道。

仿生电脑的物理通道不容易增加和减少,数量又比较少,但速度比较高,可以用分时、分频、编码等方式划分成更多虚拟的逻辑通道,应用时再动态地分配。

3.1.3 神经结和神经脊

仿生电脑的神经脊^[9],是一种芯片间通信通道,神经结堆叠成为脊索。把芯片连接成依靠通信连结的网络,为多处理单元并行运算提供一个通信骨架。

神经结是物理通道和芯片(或电路板)间的接口,结内的多路开关(见图 2)受上级称为邮局的仲裁机构控制,按结构命令和芯片的状态,拮取物理通道上的信息,转换(路由)到连接芯片的逻辑通道上,也用物理通道向邮局和网络上其它芯片传送芯片信息。

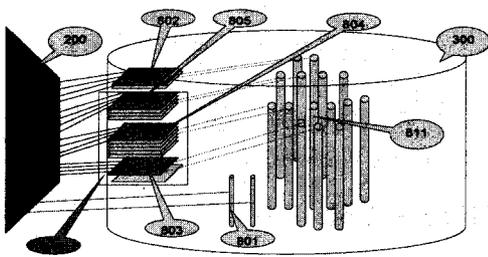


图 2 神经结的多路开关

神经结内集合多个物理通道,包括专用和通用的逻辑通道,并带有一定的智能,能完成物理通道和芯片间的信息变换。

3.1.4 金属线通信的极限

金属线路以电场波前传送信号,多少也有电磁辐射。传输线的阻抗不高,发送端需要有足够的功率。接收端采用非谐振传感器,它会吸收一定的功率,传送的扇出系数(Fan out)不高,通常做成点对点的信号通道。

用点对点传输通信,线路数目会按处理器的数量 $n * (n - 1)$ 激增。计算程序中,多对多的通信仍然是软件和硬件中的一个重要瓶颈。

3.2 光电通信

3.2.1 光电信息通道

一般光纤传送信号,传感器件的接收面积比通道截面大,接收器吸收了信号的全部能量,大多数做成点对点的通信。

无线电波在空间中传播,天线面积比传播空间小得多,天线和信道间的耦合很弱,接收器只吸收信号能量的小部分,容易做成多对多的通信^[10]。

以 1 微米波长的光计算,频率高达 3 PHz,理论带宽可达 1.5 PHz。现时做到约 40 GHz 通信带宽,不到三万分之一。原因是现在没有用相干信号。

相干光通信和无线电通信一样利用共振线路,同一信道内可以选择多个频段。几微瓦的功率就足以推动多个接收器件,得到极高的扇出。多处理器系统中,使用高扇出,可以使通道数量与处理器成线性关系而不是平方关系,简化了系统的结构。

最近几年,垂直腔面激光器件的研究,使得相干光通信可以建立在芯片规模的器件上,光信号可以用压电效应调制^[11],微机械可做成连续可调的光谐振器件^[12,13]。有人曾质疑多层反射膜激光器件工艺上的困难,但最近在强光 LED 中应用的复晶技术^[14,15],使生产廉价的光通信阵列(见图 3)成为可能。

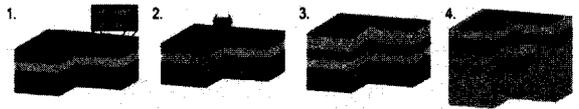


图 3 复晶技术生产廉价的光通信阵列

4 仿生电脑的软件

计算机从单核到上亿个核,软件系统怎样管理信息的变换,使系统资源无缝地配合工作,已构成一个从量变到质变的过程,编程的手段和语言都需要有革命性的突破。与此同时又要包容遗留的工作成果,要使软件在稳定中求进,更是一件艰巨的工作。

4.1 层次式的系统

鉴于系统的规模,软件可分为 3 个不同的层次,分别处理不同性质的工作。

顶层应用软件由用户提出需求,要将那些已知的信息转换成用户需要得到的信息。

只有用户才知道这些信息间的关系,所以最好是用户自己,或者在熟悉电脑性能又了解用户需求的程序员帮助下,把需求分解成较简单的对象和任务。

对象把高内聚、低耦合的信息集中在一起,组成相应的概念类。对象的属性是描述对象的信息,任务是对象可能执行的行为,可对自身和其他有关联的对象信息作变换。

对象间的属性和行为最好是无关系的,相关的行为在上一级处理效率最高。但也可以利用通信,协调各对象行为同步运作。

对象也可以从已知的原始信息出发,组合成任务要使用的信息,由下而上构建。

经过多重分解或构建,应用软件被分拆为对服务软件的呼叫。

仿生电脑的应用软件和现有的面向对象的软件差别不大,基本上可以重用。但这里的信息不单是记忆元件内的数据,而是更广泛的时变信息。新概念会有更大的发展空间。

4.2 面向服务的中间件

第二层是服务性软件。

服务也是一个信息变换,服务变换的原始信息是呼叫时带人的参数和接收到其它对象传来的信息,服务把原始信息变成用户需要的信息。

服务起源于应用软件,成熟的应用软件使用多了,便可以构成为更复杂的应用软件的低层服务。

服务着重信息的变换,与怎样完成变换的方式无关。无论是逻辑网络、冯诺曼机器、神经网络,或者任何硬件/软件的结合,都能提供同样的变换。

服务代表电脑的功能,功能越多,能完成的任务也越多。然而无论是用户或者是专业的程序员,能充分理解并灵活使用的概念总会有限,任务库就是电脑的知识库,须按不同的领域分门别类,程序员只能在自己熟悉的几个领域中选出一部分来构建应用程序。

如果把客户需求比作要攀登的山峰,服务软件就像登山的石阶,搭建石阶的方法多式多样,怎样才能以最有效的方法铺出登山的道路,就是软件行业最大的挑战。

提供服务的软件,称为中间件。中间件可以由语言指令表达,也可以由应用接口 API 集合表达,也可以由一组对象表达。服务本身也有很多层次,客户需求最终会分解成操作系统所能完成的基本服务。

4.3 仿生电脑的操作系统

第三层是操作系统的基本服务。仿生电脑的硬件与冯诺曼机器有较大差异,故操作系统有很多不同角度提供的服务。

4.3.1 资源列表

传感器、信息变换器和通信通道都是仿生电脑可用的资源。

传感器感受外界信息,由设计人员编码,变成一个或多个点上的电信号,仿生电脑内的每个传感器,都编上一个独有的名称。

信息变换器能将信息从一种形式转成另一种形式。转换过程中可能失去部分信息而不能还原,也可能加入了冗余信息而须要指定还原方案。仿生电脑内的每个变换器也都编上一个独有的名称。

专用的通信通道在硬件设计时已经确定,它把信息从固定的传感器或变换器传到另一固定的器件。

通用通道则受结构命令控制,连接指定的器件。仿生电脑内的每个通用通信通道也都编上一个独有的名称。

仿生电脑系统中维持着一组树状的硬件资源列表,其中一部分会放在持久性记忆中,在启动时再经自检确认,另一部分通过硬件搜索服务在运行中找出,也有一部分由硬件接入时向服务器报告。

4.3.2 自治系统

在加电或重置系统时,这些资源中最少有一组,也可能有很多组,经过专用通信检测到系统内其它资源的存在,并控制和管理其它资源,称为自治系统,自治系统有自己的名称。自治系统之间,可以通过通信、交换资源使用信息,也可以交换

资源的管理权。自治系统在创建时,会建立一个命名服务器,记录系统内所有服务的名字、占有的资源和外部通信方式。命名服务器组成树状的命名空间。

4.3.3 微核和线程

微核是仿生电脑里面最小的对象单位。它由电脑内的资源组成一个包含信息结构和操作方式的整体。微核有自己的名称,在微核被创建时就向命名服务器注册。通过外部的通信接口,微核能接受指令信息,按指定的操作方式,对内部信息作变换,并向其它接口转发经过变换的信息,包括向原来接口返回信息。

冯诺曼系统中的线程,通常是某时间片内占用一段记忆空间运行的软件,与微核很相似,但微核的含义比较广,只要是能接受指令信息,按既定的操作方式,对内部信息作变换的结构,不管是硬件或软硬件组合,都可以算是微核。

线程由运行中的一段记忆体的资源、前后文和处理器的中断状态组成。微核则是由传感器、信息变换器、信息存储器 and 通信通等资源组成的服务。

4.3.4 操作系统服务

微核或者微核的组合构成的服务中一部分是系统运行必需的,例如前述的命名服务器,资源管理器,控制微核组成、运行、睡眠、串行化、死亡、解体等状态的微核控制器,控制自治系统的分裂、组合、资源重分配的系统控制器,信息串行化和存储的文件系统,通信控制邮局、路由开关,硬件搜索服务,应用软件开发接口等。

操作系统服务可以由语言指令表达,也可以由应用接口 API 呼叫集合表达,也可以由一组对象表达。系统服务通常只占系统资源的小部分。

与中间服务相反,系统服务组成一个近乎正交的指令集。集合越小越好。指令构成一个控制硬件资源的虚拟机器,使不同的硬件集合也能用同样的指令构建中间服务。

系统中的信息变换器和通信通道资源大部分都处于备用状态,等待微核控制器编配用途。应用程序向系统服务取用资源,直到需要资源的峰值。必要时,操作系统会重新分配资源,使系统发挥最高效率。

操作系统尤如生物的基因决定的结构,设计时就已确定,让电脑提供基本运作的服务。

5 现实的仿生电脑

虽然高性能的仿生电脑不容易一步到位,但是,现实中的数字计算已慢慢向这方向转移。浮点计算、声音和多媒体的专用处理器、图型计算器多层次的费米结构、ARM 系列的哈佛结构、专用的外围设备、嵌入式器件的种种专用设计、手持通信设备、苹果的 A4 处理器等,都使用硬件线路代替了部分较慢的软件;网络搜索和百科全书强调概念间的联接都开始在有意识和无意间模拟生物神经网络信息变换模式。

2010年9月15日,美国波士顿的高性能嵌入式计算(HPEC)研讨会上,耶鲁大学的欧亨尼奥·卡鲁塞伊罗教授发表了一个基于人类视觉系统的高性能计算机“神经流”(NeuFlow)^[16,17]。

系统使用纽约大学(NYU)燕利勤(Yann LeCun)教授的视回路神经网络算法(Convolutional neural networks)^[18,19]模

拟人脑视神经区域, V1, V2, V4 产生与位置和尺度无关的不变量, 确认抽象的目标, 每一步骤都设有过滤层、非线性层和特性池层, 每个网络回路会由一到三步骤和跟随的分类器组成。

在 Virtex6 FPGA 的硬件上, 以 7W 功耗模拟 500 万个神经元, 得到 30 fps 的速度, 轻易地把 100 万像素的二维图像实时地识别并转换成三维目标对象, 计算结果送入手提电脑, 用在自动驾驶导航, 达到了高性能计算的效果。

我们也在研发一种实时数据采集系统(见图 4), 使用嵌入式计算机, 通过 CAN 网络和互联网, 管理分散在多个地点的数百台机器。大量的实时数据在收集的过程中逐步转换成简洁的管理资讯。可惜资源有限, 投入的人力物力以及个人的精力都不够, 除了部分通信软件外, 暂时未有可以报告的成绩。

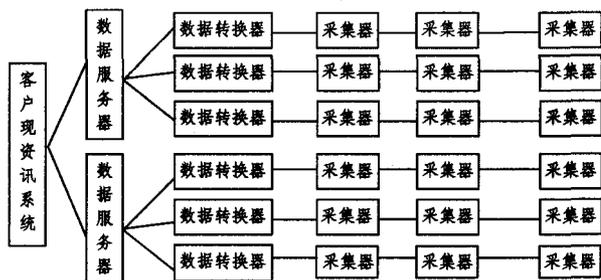


图 4 实时数据采集系统

仿生电脑的范围很庞大, 可具体研究的项目很多, 如果有其他单位或学校有兴趣, 我们很乐意与大家分享经验和无偿地共用取得的国内外知识产权。

结束语 图灵-冯诺曼模型用电子机械替代人类的思维, 创造了智能机器, 打开了资讯时代的大门。

在对算法抽象的时候, 冯诺曼体系把数据和程序分离, 引入冗余的地址参数, 降低了变换效率。

神经网络以联结表达数据间的关系, 牺牲系统的通用性, 换回变换速度。

仿生电脑就要在这两个极端之间取得平衡。

参 考 文 献

[1] 漆锋滨. E 级高性能计算机面临的挑战[R]. 全国高性能计算大会报告, 2009

[2] Turing A. MComputing machinery and intelligence[J]. Mind, 1950, 59: 433-460. <http://www.loebner.net/Prizef/TuringArticle.html>,

[3] Shannon C E. A mathematical theory of communication[J]. Bell System Technical Journal, 1948, 27: 379-423, 623-656

[4] 杨学军. 存储墙问题的思考[R]. HPCC 2009 年全国高性能计算大会报告, 2009

[5] Peter N. Glaskowsky NVIDIA's_Fermi; The_First_Complete_GPU_Architecture [OL]. http://www.nvidia.com/content/PDF/fermi_white_papers/

[6] 蒋宗礼. 人工神经网络导论 [M]. 北京: 高等教育出版社, 2003

[7] Fabio Benfenati Synaptic plasticity and the neurobiology of learning and memory [J]. ACTA BIOMED, 2007, 78(Suppl 1): 58-66

[8] Brent Przybus FPGA Families[OL]. <http://china.xilinx.com/support/documentation/>

[9] 李奕权. 仿生电脑的神经脊[P]. CN101339627

[10] 柯熙正, 席晓莉. 无线激光通信概论[M]. 北京: 北京邮电大学出版社, 2004

[11] Piazza G, et al. Design of a monolithic piezoelectrically actuated, microelectromechanical tunable, vertical-cavity surface-emitting laser[J]. OPTICS LETTERS, 2005, 30(8)

[12] Huang M C Y, Zhou Ye, Chang-Hasnain C J. Nano electro-mechanical optoelectronic tunable[J]. VCSEL. OPTICS EXPRESS, 2007, 1222. <http://www.eecs.berkeley.edu/Research/Projects/Data July 21, 2010>

[13] Huang M C Y, et al. Monolithic Piezo-Electric Actuated MEMS Tunable VCSEL[OL]. <https://buffy.eecs.berkeley.edu/PHP/resabs/resabs.php>

[14] VISIT - Vertically Integrated Systems [OL]. <http://www.visit.tu-berlin.de> 18. 08. 09

[15] SUBTUNE [OL]. <http://www.subtune.eu> Last update 01/02/2010 11:33:17

[16] 基于人类视觉系统的超级计算机[EB]. <http://it.zaobao.com/pages9/itech100917.shtml>, 2010-9-17

[17] NeuFlow-Supercomputer Can 'See' Well Enough To Navigate Roads [OL]. http://www.science20.com/news_articles/neu-flow_supercomputer_can_see_well_enough_navigate_roads, 2010-9-24

[18] A Dataflow Computer[OL]. <http://www.eng.yale.edu/elab/research/svision/svision.html>

[19] Yann LeCun. <http://yann.lecun.com/>

(上接第 278 页)

[5] Plimpton S. Fast parallel algorithms for short-range molecular dynamics[J]. Journal of Computational Physics, 1995, 117(1): 1-19

[6] Allen M P, Tildesley D J. Computer Simulation of Liquids[M]. Oxford: Clarendon Press, 1994

[7] Allen F, et al. BlueGene: A vision for protein science using a petaflop computer[J]. IBM Systems Journal, 2001, 40(2): 310-327

[8] Gara A, et al. Overview of the Blue Gene/L System Architecture [J]. IBM Journal of Research and Development, 2005, 49(2/3):

195-212

[9] Phillips C, et al. Scalable Molecular Dynamics with NAMD[J]. J. Comput. Chem., 2005, 26(16): 1781-1802

[10] Humphrey W, Dalke A, Schulten K. VMD-Visual Molecular Dynamics[J]. J. Molec. Graphics, 1996, 14(1): 33-38

[11] Yang Jue-kuan, Wang Yu-juan, Chen Yun-fei. GPU accelerated molecular dynamics simulation of thermal conductivities[J]. J. Comp. Phys., 2007, 221(1): 799-804

[12] Liu Wei-guo, et al. Accelerating molecular dynamics simulations using Graphics Processing Units with CUDA [J]. Journal of Computational Physics, 2008, 179(9): 634-641