第一章 计算机基础

一、计算机体系结构分类（Flynn 分类法）

体系结构类型	结构	关键特性	代表
单指令流单数据流 SISD	控制部分：一个 处 理 器：一个 主存模块：一个		单处理器系统
单指令流多数据流 SIMD	控制部分：一个 处 理 器：多个 主存模块：多个	各处理器以异步的形式执行同一条指令	并行处理机 阵列处理机 超级向量处理机 PS：GPU 属于 SIMD
多指令流单数据流 MISD	控制部分：多个 处 理 器：一个 主存模块：多个	被证明不可能，至少是不实际	目前没有，有文献称流水线计算机为此类
多指令流多数据流 MIMD	控制部分：多个 处 理 器：多个 主存模块：多个	能够实现作业、任务、指令等各级全面并行	多核处理器（SMP、BMP、MP） 多处理机系统（MPP） 多计算机

二、CISC 与 RISC

	多频率差别大变长	**多种**	**微程序控制技术**	
	少频率接近定长操作寄存器	**方式少**	**增加了通用寄存器硬布线逻辑控制为主适合采用流水线**	

复杂指令集计算机（CISC，Complex Instruction Set Computers）

精简指令集计算机（RISC，Reduced Instruction Set Computers

三、计算机系统的多层次结构

硬联逻辑级：这是计算机的内核，由门、触发器等逻辑电路组成。

微程序级：这一级的机器语言是微指令集，程序员用微指令编写的微程序一般直接由硬件执行。

传统机器级：这一级的机器语言是该机的指令集，程序员用机器指令编写的程序可以由微程序进行解释。

操作系统级：从操作系统的基本功能来看，一方面它要直接管理传统机器中的软硬件资源，另一方面它又是传统机器的延伸。

汇编语言级：这一级的机器语言是汇编语言，完成汇编语言翻译的程序称为汇编程序。

高级语言级：这一级的机器语言就是各种高级语言，通常用编译程序来完成高级语言翻译的工作。

应用语言级：这一级是为了使计算机满足某种用途而专门设计的，因此，这一级的机器语言就是各种面向问题的应用语言。

四、流水线

相关参数计算（流水线执行时间计算、流水线吞吐率、流水线加速比）

（1）流水线周期：执行时间最长的一段

（2）流水线执行时间（理论公式）：$(t_1+t_2+\cdots+t_k)+(n-1)* t_{max}$

（3）流水线执行时间（实践公式）：$k*t_{max}+(n-1)* t_{max}$

（4）流水线吞吐率：TP=指令条数/流水线执行时间

（5）流水线最大吞吐率

$$TP_{max}=\lim_{n\to\infty}\frac{n}{(k+n-1)t_{max}}=\frac{1}{t_{max}}$$

（6）流水线加速比：顺序执行时间/流水线执行时间

五、存储系统

1、Cache

（1）Cache 的相关概念

Cache 的功能：提高 CPU 数据输入输出的速率，突破冯·诺依曼瓶颈，即 CPU 与存储系统间数据传送带宽限制。

在计算机的存储系统体系中，Cache 是除寄存器以外，访问速度最快的层次。

使用 Cache 改善系统性能的依据是程序的局部性原理。

- 时间局部性：指程序中的某条指令一旦执行，不久以后该指令可能再次执行。
- 空间局部性：指一旦程序访问了某个存储单元，不久以后，其附近的存储单元也将被访问，即程序在一段时间内所访问的地址可能集中在一定的范围内。

Cache 对程序员来说是透明的。

（2）Cache 映像方式**【直接由硬件完成地址映像】**

直接相联映像：硬件电路较简单，但冲突率很高。

全相联映像：电路难于设计和实现，只适用于小容量的 cache，冲突率较低。

组相联映像：直接相联与全相联的折中。

（3）平均存取时间

如果以 h 代表对 Cache 的访问命中率，（1-h）称为失效率（未命中率），t_1 表示 Cache 的周期时间，t_2 表示主存储器周期时间，以读操作为例，使用“Cache+主存储器”的系统的平均周期为 t_3，则：$t_3 = h \times t_1 + (1-h) \times t_2$

（4）Cache 页面淘汰算法

随机算法（RAND）

先进先出算法（FIFO）

近期最少使用算法（LRU）-----**年龄计数器**

最不经常使用页置换算法（LFU）-----算法计数器位数多，实现困难。

（5）Cache 的读写过程

写直达：同时写 Cache 与内存

写回：只写 Cache，淘汰页面时，写回内存

标记法：只写入内存，并将标志位清 0，若用到此数据，需要再次调取

2、磁盘管理

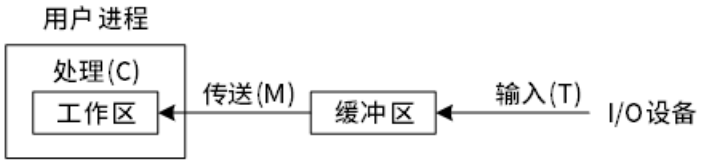
（1）存取时间=寻道时间+等待时间，寻道时间是指磁头移动到磁道所需的时间；等待时间为等待读写的扇区转到磁头下方所用的时间。

（2）读取磁盘数据的时间应包括以下三个部分：找磁道的时间、找块（扇区）的时间，即旋转延迟时间、传输时间。

（3）磁盘调度算法：先来先服务 FCFS（谁先申请先服务谁）；最短寻道时间优先 SSTF（申请时判断与磁头当前位置的距离，谁短先服务谁）；扫描算法 SCAN（电梯算法，双向扫描）；循环扫描 CSCAN（单向扫描）。

（4）单双缓冲区

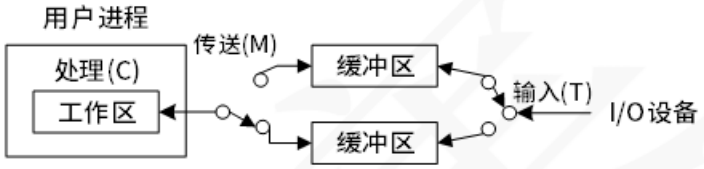
单缓冲区：在单缓冲情况下，每当用户进程发出一 I/O 请求时，OS 便在主存中为之分配一缓冲区。在块设备输入时，假定从磁盘把一块数据输入到缓冲区的时间为 T,OS 将该缓冲区中的数据传送到用户区的时间为 M，而 CPU 对这一块数据的处理时间为 C，T 和 C 是可以并行的。



双缓冲区：

由于缓冲区是共享资源，生产者与消费者在使用缓冲区时必须互斥。

如果消费者尚未取走缓冲区的数据，生产者又生产新的数据，也无法将它送入缓冲区，所以设置两个缓冲区。



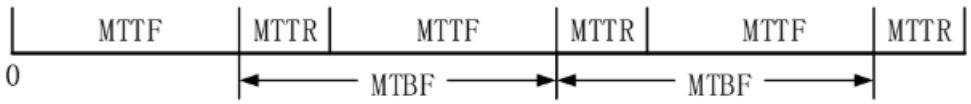
六、可靠性

1、串联系统计算： $R = R_1 \times R_2 \times \ldots \times R_n$

2、并联系统计算：$R = 1 - (1 - R_1) \times (1 - R_2) \times \ldots \times (1 - R_n)$

3、N 模混联系统：先将整个系统划分为多个部分串联 R1、R2…等，再计算 R1、R2 内部的并联可靠性，代入原公式。

4、可靠性指标



在实际应用中，一般 MTTR 很小，所以通常认为 MTBF≈MTTF。

平均无故障时间 → (MTTF) MTTF=1/λ，λ为失效率

平均故障修复时间 → (MTTR) MTTR=1/μ，μ为修复率

平均故障间隔时间 → (MTBF) MTBF = MTTR + MTTF

可靠性可以用 MTTF/（1+MTTF）来度量。

可用性可以用 MTBF/（1+MTBF）来度量。

可维护性可以用 1/（1+MTTR）来度量。

注：MTBF 大，MTTR 小表示系统具有高可靠性和高可用性。

七、性能指标

1、平均每条指令的平均时钟周期个数（CPI，Clock Per Instruction）

2、每（时钟）周期运行指令条数（IPC，Instruction Per Clock）

3、百万条指令每秒（MIPS，Million Instructions Per Second），MIPS = 指令条数/（执行时间×10^6）= 主频/CPI = 主频×IPC

4、每秒百万个浮点操作（MFLOPS，Million Floating-point Operations Per Second），**MFLOPS = 浮点操作次数/（执行时间×10^6）**

5、响应时间（RT，Response Time）

6、主频 = 外频 * 倍频

7、计算机系统的性能一般包括两个大的方面。

一方面是它的可靠性或可用性，也就是计算机系统能够正常工作的时间，其指标可以是能够持续工作的时间长度（例如，平均无故障时间），也可以是一段时间内，能正常工作的时间所占的百分比；

另一方面是它的处理能力或效率，这又可以分为三类指标，第一类指标是吞吐率（例如，系统在单位时间内能够处理正常作业的个数），第二类是响应时间（从系统得到输入到给出输出之间的时间），第三类指标是资源利用率，即在给定的时间区间中，各种部件（包括硬件设备和软件系统）被使用的时间与整个时间之比。

当然，不同的系统对性能指标的描述有所不同，例如，计算机网络系统常用的性能评估指标为信道传输速率，信道吞吐量和容量、信道利用率、传输延迟、响应时间和负载能力等。

八、性能评价方法

1、时钟频率法：以时钟频率高低衡量速度。

2、指令执行速度法：表示机器运算速度的单位是 MIPS。

3、等效指令速度法 （ Gibson mix，吉普森混合法）：通过各类指令在程序中所占的比例（W_i）进行计算得到的。**特点：考虑指令比例不同的问题。**

4、数据处理速率法（PDR）：PDR 值的方法来衡量机器性能，PDR 值越大，机器性能越好。PDR = L/R　　**特点：考虑 CPU+存储**

5、综合理论性能法（CTP）：CTP 用 MTOPS（Million Theoretical Operations Per Second，每秒百万次理论运算）表示。CTP 的估算方法是，首先算出处理部件每个计算单元的有效计算率，再按不同字长加以调整，得出该计算单元的理论性能，所有组成该处理部件的计算单元的理论性能之和即为 CTP。

6、基准程序法：把应用程序中用得最多、最频繁的那部分核心程序作为评估计算机系统性能的标准程序，称为基准测试程序（benchmark）。基准程序法是目前一致承认的测试系统性能的较好方法。

【测试精确度排名】真实的程序->核心程序->小型基准程序->合成基准程序

（1）Dhrystone 基准程序：它是一个综合性的整数基准测试程序，是为了测试编译器和 CPU 处理整数指令和控制功能的有效性，人为地选择一些典型指令综合起来形成的测试程序。

（2）Linpack 基准程序：它是国际上最流行的用于测试高性能计算机系统浮点性能的测试。

（3）Whetstone 基准程序：它是用 Fortran 语言编写的综合性测试程序，主要由执行浮点运算、功能调用、数组变址、条件转移和超越函数的程序组成。

（4）SPEC 基准程序：一种是测试计算机完成单项任务有多快，称为速度测试；另一种是测试计算机在一定时间内能完成多少项任务，称为吞吐率测试。

（5）TPC 基准程序： TPC（Transaction Processing Council，事务处理委员会）基准程序用以评测计算机在事务处理、数据库处理、企业管理与决策支持系统等方面的性能。该基准程序的评测结果用每秒完成的事务处理数 TPC 来表示。

TPC-A 基准程序规范用于评价在 OLTP 环境下的数据库和硬件的性能；

TPC-B 测试的是不包括网络的纯事务处理量，用于模拟企业计算环境；

TPC-C 测试的是联机订货系统；

TPC-D、TPC-H 和 TPC-R 测试的都是决策支持系统，其中 TPC-R 允许有附加的优化选项；

TPC-E 测试的是大型企业信息服务系统。

TPC-W 是基于 Web 应用的基准程序，用来测试一些通过 Internet 进行市场服务和销售的商业行为，所以 TPC-W 可以看作是一个服务器的测试标准。

第二章 嵌入式系统

一、嵌入式基本概念

嵌入式系统是以应用为中心、以计算机技术为基础，并将可配置与可裁剪的软、硬件集成于一体的专用计算机系统【面向特定领域】，需要满足应用对功能、可靠性、成本、体积和功耗等方面的严格要求。

从计算机角度看，嵌入式系统是指嵌入各种设备及应用产品内部的计算机系统。它主要完成信号控制的功能，体积小、结构紧凑，可作为一个部件埋藏于所控制的装置中。

一般嵌入式系统由嵌入式处理器、相关支撑硬件、嵌入式操作系统、支撑软件以及应用软件组成。

嵌入式系统通常通过外部接口采集相关输入信息或人机接口输入的命令,对输入数据进行加工和计算，并将计算结果通过外部接口输出，以控制受控对象。

嵌入式系统初始化过程：片级初始化→板级初始化→系统级初始化

二、嵌入式微处理器体系结构

体系结构分类	定义	特点	典型应用
冯·诺依曼结构	冯·诺依曼结构也称普林斯顿结构，是一种将程序指令存储器和数据存储器合并在一起的存储器结构。	指令与数据存储器合并在一起。 指令与数据都通过相同的数据总线传输。	一般用于 PC 处理器，如 I3、I5、I7 处理器。 注：常规计算机属于冯·诺依曼结构
哈佛结构	哈佛结构是一种并行体系结构，它的主要特点是将程序和数据存储在不同的存储空间中，即程序存储器和数据存储器是两个独立的存储器，每个存储器独立编址、独立访问。	指令与数据分开存储，可以并行读取，有较高数据的吞吐率。 有 4 条总线：指令和数据的数据总线与地址总线。	一般用于嵌入式系统处理器。 注：DSP 属于哈佛结构

嵌入式微处理器主要用于处理相关任务。由于嵌入式系统通常都在室外使用，可能处于不同环境，因此选择处理器芯片时，也要根据不同使用环境选择不同级别的芯片。其主要因素是芯片可适应的工作环境温度。通常，我们把芯片分为民用级、工业级和军用级。

民用级器件的工作温度范围：0~70℃

工业级器件的工作温度范围：-40~85℃

军用级器件的工作温度范围：-55~150℃

当然，除了环境温度外，环境湿度、震动、加速度等也是应考虑的因素。

三、总线与接口

1、总线基本概念

总线是一组能为多个部件分时共享的信息传送线，用来连接多个部件并为之提供信息交换通路。(总线通常是半双工的)

2、总线的特点

（1）挂接在总线上的多个部件只能分时向总线发送数据，但可同时从总线接收数据。

（2）通过总线复用方式可以减少总线中信号线的数量，以较少的信号线传输更多的信息。

3、总线的分类

（1）从功能上来对总线进行划分数据总线【传数据】、地址总线【传地址】和控制总线【传控制信号】。

（2）从数据传输方式的角度划分为并行总线和串行总线

并行总线：将数据字节的各位用多条数据线同时进行传送。【短距离】

串行总线：数据是一位一位地进行传输的，在传输中每一位数据都占据一个固定的时间长度。【长距离，传输波特率可调整，正确性依赖于校验码，数据传输方式可以使用多种】

4、接口：I/O 接口，也称为 I/O 控制器，它是主机和外设（外部设备）直接的交接界面，通过接口可以实现主机和外设之间的信息交换。

接口的主要功能表现在以下 5 个方面：

（1）实现主机和外设的通信联络控制。

（2）进行地址译码和设备选择。

（3）实现数据缓冲。

（4）数据格式的变换。

（5）传递控制命令和状态信息。

接口的分类：串行接口和并行接口。

四、嵌入式系统软件

1、基本概念

嵌入式系统是一种以应用为中心，以计算机技术为基础，可以适应不同应用对功能、可靠性、成本、体积和功耗等方面的要求，集可配置可裁减的软、硬件于一体的专用计算机系统。（面向特定领域）

嵌入式系统具有以下特点：

规模较小、开发难度大、实时性和可靠性要求高、要求固化存储等。

2、嵌入式系统软件分类

根据系统对时间的敏感程度可将嵌入式系统划分为：

（1）嵌入式系统

（2）嵌入式实时系统：强实时系统、弱实时系统等。

从安全性要求看，嵌入式系统还可分为：

（1）安全攸关系统（2）非安全攸关系统

五、嵌入式操作系统

1、嵌入式操作系统特点：嵌入式操作系统具有一般操作系统的功能，同时具有嵌入式软件的特点，主要有：非通用型操作系统；在性能和实时性方面可能有严格的限制；能源、成本和可靠性通常是影响设计的重要因素；占用资源少；可剪裁、可配置；微型化；实时性；可靠性；可定制。从减少成本和缩短研发周期考虑，要求嵌入式操作系统能运行在不同 的微处理器平台上，能针对硬件变化进行结构与功能上的配置，以满足不同应用需要；易移植性。为了提高系统的易移植性，通常采用硬件抽象层（HAL）和板级支持包（BSP）的底层设计技术。

2、按照系统对响应时间的敏感程度，可以分为：

硬实时系统：系统对响应时间有严格要求，若响应时间不能满足，是绝对不允许的，会引起系统的崩溃或致命的错误

软实时系统：系统对响应时间有要求，若响应时间不能满足，会带来额外可接受的代价

非实时系统：响应时间没有严格要求

如分时操作系统，基于公平性原则，各进程分享处理器，获得大致相同的运行时间

3、嵌入式实时操作系统实时性的评价指标：中断响应和延迟时间、任务切换时间、信号量混洗时间、系统响应时间。

4、嵌入式实时操作系统调度算法

优先级调度算法：系统为每个任务分配一个相对固定的优先顺序。

抢占式优先级调度算法：根据任务的紧急程度确定该任务的优先级。大多数 RTOS 调度算法都是抢占方式（可剥夺方式）。

最早截止期调度算法（EDF 算法）：根据任务的截止时间来确定其优先级，对于时间期限最近的任务，分配最高的优先级。

最晚截止期调度算法：指调度程序按每个任务的最接近其截止期末端的时间进行调度，本题描述的就是最晚截止期调度算法。

5、嵌入式操作系统的分类

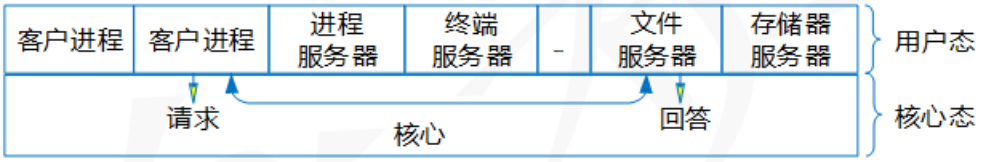
（1）内核是操作系统的核心部分，它管理着系统的各种资源。内核可以看成连接应用程序和硬件的一座桥梁，是直接运行在硬件上的最基础的软件实体。

目前从内核架构来划分，可分为宏内核（Monolithic Kernel）和微内核（Micro Kernel）

（2）微内核操作系统

现代操作系统大多拥有两种工作状态，分别是核心态和用户态。一般应用程序工作在用户态，而内核模块和最基本的操作系统核心工作在核心态。

将传统的操作系统代码放置到更高层，从操作系统中去掉尽可能多的东西，而只留下最小的核心，称之为微内核。（C/S 结构）



操作系统的内核服务：异常和中断、计时器、I/O 管理等。

微内核与单体内核对比：

	实质	优点	缺点
单体内核	将图形、设备驱动及文件系统等功能全部在内核中	减少进程间通信和状态切换的系统开销，获得较高的运行效率。	**内核庞大，占用资源较多且不易剪裁。**

	实现，运行在内核状态和同一地址空间。		系统的稳定性和安全性不好。
微内核【鸿蒙操作系统】	只实现基本功能，将图形系统、文件系统、设备驱动及通信功能放在内核之外。	**微内核系统结构相当清晰，有利于协作开发.** **内核精练，便于剪裁和移植（灵活性、可扩展性）。** **系统服务程序运行在用户地址空间，系统的可靠性、稳定性和安全性较高。** 可用于分布式系统。	用户状态和内核状态需要频繁切换，从而导致系统效率不如单体内核，性能偏低。

六、多核操作系统

1、多核操作系统设计的核心技术：核结构、Cache 设计、核间通信、任务调度、中断处理同步互斥等。

2、对于多核 CPU，优化操作系统任务调度算法是保证效率的关键。一般任务调度算法有全局队列调度和局部队列调度。

全局队列调度：指操作系统维护一个全局的任务等待队列，当系统中有一个 CPU 核心空闲时，操作系统就从全局任务等待队列中选取就绪任务开始在此核心上执行。这种方法的优点是 CPU 核心利用率较高。

局部队列调度：指操作系统为每个 CPU 内核维护一个局部的任务等待队列，当系统中有一个 CPU 内核空闲时，便从该核心的任务等待队列中选取恰当的任务执行，这种方法的优点是任务基本上无需在多个 CPU 核心间切换，有利于提高 CPU 核心局部 Cache 命中率。目前多数多核 CPU 操作系统采用的是基于全局队列的任务调度算法。

七、嵌入式设计与开发

1.嵌入式软件的开发也与传统的软件开发方法存在比较大的差异，主要表现在以下方面：

（1）嵌入式软件开发是在宿主机（PC 机或工作站）上使用专门的嵌入式工具开发，生成二进制代码后，需要使用工具卸载到目标机或固化在目标机储存器上运行。

（2）嵌入式软件开发时更强调软/硬件协同工作的效率和稳定性。

（3）嵌入式软件开发的结果通常需要固化在目标系统的储存器或处理器内部储存器资源中。

（4）嵌入式软件的开发一般需要专门的开发工具、目标系统和测试设备。

（5）嵌入式软件对实时性的要求更高。

（6）嵌入式软件对安全性和可靠性的要求较高。

（7）嵌入式软件开发时要充分考虑代码的规模。

（8）在安全攸关系统中的嵌入式软件，其开发还应满足某些领域对设计和代码的审定。

（9）模块化设计即将一个较大的程序按功能划分成若干程序模块，每个模块实现特定的功能。

2.低功耗设计：

基于硬件的低功耗	**基于软件的低功耗**
板级电路低功耗设计 选择低功耗处理器 总线的低功耗设计 接口驱动电路的设计 分区分时供电技术	编译优化技术 软件与硬件的协同设计 算法优化

第三章 数据库系统

一、数据库模式

1、体系结构

（1）三级模式：外模式对应视图，模式（也称为概念模式）对应数据库表，内模式对应物理文件。

（2）两层映像：外模式-模式映像，模式-内模式映像；两层映像可以保证数据库中的数据具有较高的逻辑独立性和物理独立性。

（3）逻辑独立性：数据的逻辑结构发生变化后，用户程序也可以不修改。但是为了保证应用程序能够正确执行，需要修改外模式和概念模式之间的映像。

（4）物理独立性：当数据的物理结构发生改变时，应用程序不用改变。但是为了保证应用程序能够正确执行，需要修改概念模式和内模式之间的映像。

2、视图

（1）数据库视图

它一个虚拟表（逻辑上的表），其内容由查询定义（仅保存 SQL 查询语句）。同真实的表一样，视图包含一系列带有名称的列和行数据。但是，视图并没有真正存储这些数据，而是通过查询原始表动态生成所需要的数据。

（2）视图的优点

视图能简化用户操作

视图使用户能以多种角度看待同一数据

视图对重构数据库提供了一定程度的逻辑独立性

视图可以对机密数据提供安全保护

（3）物化视图

它不是传统意义上虚拟视图，是实体化视图，其本身会存储数据。同时当原始表中的数据更新时，物化视图也会更新。

二、分布式数据库

（1）分布式数据库特点

数据独立性。除了数据的逻辑独立性与物理独立性外，还有数据分布独立性（分布透明性）。

集中与自治共享结合的控制结构。各局部的 DBMS 可以独立地管理局部数据库，具有自治的功能。同时，系统又设有集中控制机制，协调各局部 DBMS 的工作，执行全局应用。

适当增加数据冗余度。在不同的场地存储同一数据的多个副本，可以提高系统的可靠性和可用性，同时也能提高系统性能。

（提高系统的可用性，即当系统中某个节点发生故障时，因为数据有其他副本在非故障场地上，对其他所有场地来说，数据仍然是可用的，从而保证数据的完备性。）

（2）分布透明性

分片透明性：用户不必关心数据分不分片，怎么分片（水平分片：按记录分；垂直分片：按字段分；混合分片）

位置透明性：用户不必关心数据存放在何处

局部数据模型透明性（逻辑透明）：用户不必关心局部 DBMS 支持哪种数据模型、使用哪种语言

复制透明：用户不必关心各个结点数据的复制与同步更新

（3）两阶段提交协议 2PC

2PC 事务提交的两个阶段

表决阶段，目的是形成一个共同的决定

执行阶段，目的是实现这个协调者的决定

两条全局提交规则

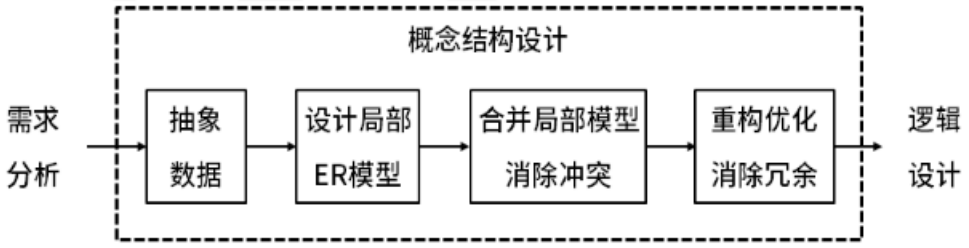
只要有一个参与者撤销事务，协调者就必须做出全局撤销决定

只有所有参与者都同意提交事务，协调者才能做出全局提交决定

三、数据库设计过程

（1）需求分析：根据当前和未来应用的数据要求以及数据处理要求，对用户数据维度需求进行分析，获得产物包括数据流图、数据自定、需求说明书。

（2）概念结构设计过程：根据需求分析结果以及用户要求进行概念结构设计，获得产物 ER 模型。



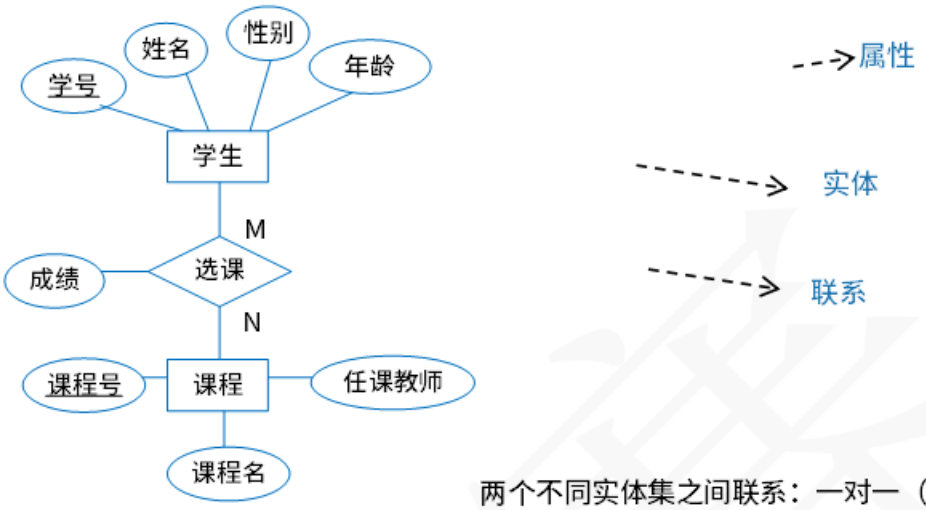
集成的方法：多个局部 E-R 图一次集成；逐步集成，用累加的方式一次集成两个局部 E-R。

集成产生的冲突及解决办法：

- 属性冲突：包括属性域冲突和属性取值冲突。
- 命名冲突：包括同名异义和异名同义。

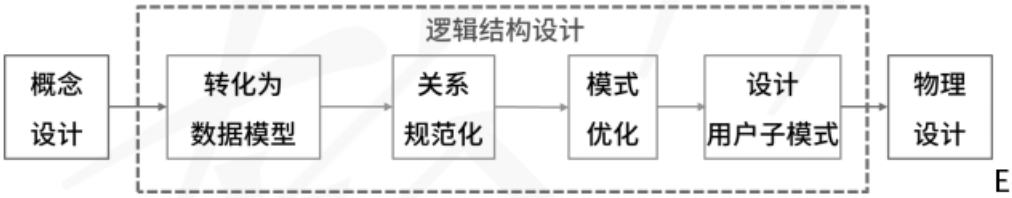
- 结构冲突：包括同一对象在不同应用中具有不同的抽象，以及同一实体在不同局部 E-R 图中所包含的属性个数和属性排列次序不完全相同。

E-R 模型



两个不同实体集之间联系：一对一（1：1）、一对多（1：n）、多对多（m：n）

（3）逻辑结构设计过程：根据用户需求、ER 模型、转换规则、规范化理论等进行逻辑结构设计，获得产物数据模型，典型的数据模型有关系模式、对象模型等，其中关系模式可以理解为二维表。



E-R 图向关系模式的转换：

- 实体向关系模式的转换
- 联系向关系模式的转换

关系模式的规范化

确定完整性约束（保证数据的正确性）

用户视图的确定（提高数据的安全性和独立性）

- 根据数据流图确定处理过程使用的视图
- 根据用户类别确定不同用户使用的视图

应用程序设计

数据模型三要素：数据结构、数据操作、数据的约束条件。

四、关系代数

1、并（结果为二者元组之和去除重复行）

2、交（结果为二者重复行）

3、差（前者去除二者重复行）

以元组行作为整体进行判断，类似于集合运算。

4、笛卡尔积

结果列数为二者属性列数之和，行数为二者元组行数的乘积。

两个表做笛卡尔积，结果表的元组由前表与后表的元组拼接而成，不同的排列组合形成不同的结果元组。

5、投影（筛选符合条件的属性列）

6、选择（筛选符合条件的元组）

属性名可以依次标序号，直接以数字形式出现在表达式中。

7、自然连接

结果列数为二者属性列数之和减去重复列,行数为二者同名属性列其值相同的结果元组。笛卡尔积、选择、投影的组合表示可以与自然连接等价。

普通连接的条件会写出，没有写出则表示为自然连接。

五、规范化理论

1、非规范化存在的问题

规范化过程是为了解决数据冗余、删除异常、插入异常、更新异常（修改操作一致性问题）等问题。

数据冗余：重复存储数据较多，浪费存储空间。

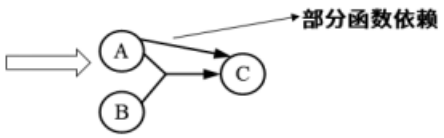
更新异常（引起修改操作的不一致性）：若不注意，会使一些数据被修改，另一些数据未被修改，导致数据修改的不一致性。

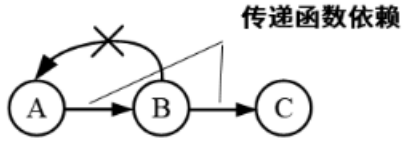
插入异常：未提供主键，当主键为空时，不能进行插入操作。

删除异常：删除部分信息时会删除整条记录，找不到原记录。

2、规范化

（1）函数依赖

部分函数依赖	传递函数依赖
	



由：A -> B，B -> C，可得出A->C。
此为传递函数依赖

（2）Armstrong 公理

关系模式 R <U，F >来说有以下的推理规则：

A1.自反律（Reflexivity）：若 Y⊆X⊆U，则 X →Y 成立。

A2.增广律（Augmentation）：若 Z⊆U 且 X→Y，则 XZ→YZ 成立。

A3.传递律（Transitivity）：若 X→Y 且 Y→Z，则 X→Z 成立。

根据 A1，A2，A3 这三条推理规则可以得到下面三条推理规则：

合并规则：由 X→Y，X→Z，有 X→YZ。 （A2， A3）

伪传递规则：由 X→Y，WY→Z，有 XW→Z。 （A2， A3）

分解规则：由 X→Y 及 Z ⊆ Y，有 X→Z。 （A1， A3）

（3）键与属性

候选键（候选码）是能够唯一标识元组却无冗余的属性组合，可以有多种不同的候选键，在其中任选一个作为主键。候选键的求取可以利用图示法找入度为 0 的属性集合，并在此基础上进行扩展，最终找到能够遍历全图的最小属性组合作为候选键，对于入度为 0 在关系依赖集中可以理解为从未在箭线右侧出现。

组成候选码的属性就是主属性，其他为非主属性。

外键是其他关系模式的主键。

【解题技巧：判断主键和外键】

找主键：使用图示法找主键。

找外键：外键是其它关系模式的主键。

全码：关系模型的所有属性组是这个关系模式的候选码，称为全码。

完整性约束相关：了解主键、外键相关的概念，根据题干，做出相关判断。

（4）范式：规范化是为了解决数据冗余、删除异常、插入异常、更新异常等问题。

第一范式（1NF）：在关系模式 R 中，当且仅当所有域只包含原子值，即每个属性都是不可再分的数据项，则称关系模式 R 是第一范式。

第二范式（2NF）：当且仅当关系模式 R 是第一范式（1NF），且每一个非主属性完全依赖候选键（没有不完全依赖）时，则称关系模式 R 是第二范式。

第三范式（3NF）：当且仅当关系模式 R 是第二范式（2NF），且 R 中没有非主属性传递依赖于候选键时，则称关系模式 R 是第三范式。

BC 范式（BCNF）：设 R 是一个关系模式，F 是它的依赖集，当且仅当其 F 中每个依赖的决定因素必定包含 R 的某个候选码时，R 属于 BCNF。【当且仅当关系模式 R 是第三范式（3NF）】

3、模式分解

规范化过程-拆表即分解关系模式。

（1）无损分解

无损联接分解：指将一个关系模式分解成若干个关系模式后，通过自然联接和投影等运算仍能还原到原来的关系模式

【公式法】

定理：如果 R 的分解为ρ＝{ R1，R2}，F 为 R 所满足的函数依赖集合，分解ρ具有无损联接性的充分必要条件是：

$$R1 \cap R2 \rightarrow (R1 - R2) \text{ 或 } R1 \cap R2 \rightarrow (R2 - R1)$$

（2）保持函数依赖

设数据库模式ρ={R1，R2，…，Rk}是关系模式 R 的一个分解，F 是 R 上的函数依赖集，ρ中每个模式 Ri 上的 FD 集是 Fi。如果{F1，F2，…，Fk}与 F 是等价的（即相互逻辑蕴涵），那么称分解ρ保持 FD。

六、数据控制

1、数据控制功能包括：①安全性（security）；②完整性（integrality）；③并发控制（concurrency control）；④故障恢复（recovery from failure）

并发控制（concurrency control）是指在多用户共享的系统中，许多用户可能同时对同一数据进行操作。DBMS 的并发控制子系统负责协调并发事务的执行，保证数据库的完整性不受破坏，避免用户得到不正确的数据。

2、数据安全性控制

安全性（security）是指保护数据库受恶意访问，即防止不合法的使用所造成的数据泄漏、更改或破坏。这样，用户只能按规定对数据进行处理，例如，划分了不同的权限，有的用户只能有读数据的权限，有的用户有修改数据的权限，用户只能在规定的权限范围内操纵数据库。

措施	说明
用户标识和鉴定	最外层的安全保护措施，可以使用用户账户、口令及随机数检验等方式

存取控制	对用户进行授权，包括操作类型（如查找、插入、删除、修改等动作）和数据对象（主要是数据范围）的权限。（Grant 和 Revoke）
密码存储和传输	对远程终端信息用密码传输
视图的保护	对视图进行授权
审计	使用一个专用文件或数据库，自动将用户对数据库的所有操作记录下来

3、数据完整性控制

完整性（integrality）是指数据库正确性和相容性，是防止合法用户使用数据库时向数据库加入不符合语义的数据。保证数据库中数据是正确的，避免非法的更新。

- 实体完整性约束：规定基本关系的主属性不能取空值。
- 参照完整性约束：关系与关系间的引用，其他关系的主键或空值。
- 用户自定义完整性约束：应用环境决定。

4、并发控制

（1）事务的特性（ACID）

原子性（Atomicity）是指事务包含的所有操作要么全部成功，要么全部失败回滚。这些操作是一个整体，不能部分地完成。

一致性（Consistency）是指事务必须使数据库从一个一致性状态变换到另一个一致性状态，也就是说一个事务执行之前和执行之后都必须处于一致性状态。

隔离性（Isolation）是指一个事务的执行不能被其他事务干扰，即一个事务内部的操作及使用的数据对并发的其他事务是隔离的。

持久性（Durability，永久性）是指一个事务一旦被提交了，那么对数据库中的数据的改变就是永久性的，无论发生何种故障，都不应对其有任何影响。

（2）并发产生的问题

◆ 丢失更新

T1	T2
①读A=10 ② ③A=A-5写回 ④	读A=10 A=A-8 写回

◆ 不可重复读

T1	T2
①读A=20 读B=30 求和=50 ② ③读A=70 读B=30 求和=100 (验算不对)	 读A=20 A←A+50 写A=70

◆ 读“脏”数据

T1	T2
①读A=20 A←A+50 写回70 ② ③ROLLBACK A恢复为20	 读A=70

（3）锁的分类

- 共享锁（S 锁）：又称读锁，若事务 T 对数据对象 A 加上 S 锁，其他事务只能再对 A 加 S 锁，而不能加 X 锁，直到 T 释放 A 上的 S 锁。
- 排他锁（X 锁）：又称写锁。若事务 T 对数据对象 A 加上 X 锁，其他事务不能再对 A 加任何锁，直到 T 释放 A 上的锁。

（4）封锁协议

一级封锁协议。事务 T 在修改数据 R 之前必须先对其加 X 锁，直到事务结束才释放。可防止丢失修改

T1	T2
① 对A加写锁 ② ③ 读A=10 ④ A=A-5写回 ⑤ 释放对A的写锁 ⑥ ⑦ ⑧	 对A加写锁 等待 等待 等待 得到通知，读A=5 A=A-8 写回 释放对A的写锁

二级封锁协议。一级封锁协议加上事务 T 在读取数据 R 之前先对其加 S 锁，读完后即可释放 S 锁。可防止丢失修改，还可防止读“脏”数据

T1	T2
① 对A加写锁 ② 读A=20 ③ A←A+50 ④ 写回70 ⑤ ⑥ ROLLBACK ⑦ A恢复为20 ⑧	 对A加读锁 等待 等待 得到通知，读A=20 释放对A的读锁

三级封锁协议。一级封锁协议加上事务 T 在读取数据 R 之前先对其加 S 锁，直到事务结束才释放。可防止丢失修改、防止读“脏”数据与防止数据（不可）重复读

T1	T2
① 对A与B加S锁（读锁） 读A=20 读B=30 求和=50 ② ③读A=20 读B=30 求和=50 释放对A和B的读锁	 对A加X锁（写锁） 注：由于A已加了读锁，所以等待 等待 等待 等待 等待 得到通知，读A=20 A←A+50 写A=70 释放对A的写锁

两段锁协议。可串行化的。可能发生死锁

5、故障恢复

故障恢复（recovery from failure）。数据库中的 4 类故障是事务内部故障、系统故障、介质故障及计算机病毒。故障恢复主要是指恢复数据库本身，即在故障引起数据库当前状态不一致后，将数据库恢复到某个正确状态或一致状态。恢复的原理非常简单，就是要建立冗余（redundancy）数据。

（1）冷备份也称为静态备份，是将数据库正常关闭，在停止状态下，将数据库的文件全部备份（复制）下来。

（2）热备份也称为动态备份，是利用备份软件，在数据库正常运行的状态下，将数据库中的数据文件备份出来。

（3）完全备份：备份所有数据

（4）差量备份：仅备份上一次完全备份之后变化的数据

（5）增量备份：备份上一次备份之后变化的数据

（6）日志文件 ： 事务日志是针对数据库改变所做的记录，它可以记录针对数据库的任何操作，并将记录结果保存在独立的文件中。

（7）数据库故障与恢复

撤销事务（UNDO）：故障发生时未完成的事务，放入 Undo 撤销。

重做事务（REDO）：故障发生前已提交的事务，放入 Redo 重做。

七、NoSQL

（1）概念

NoSQL（Not-only SQL）：不仅仅只是 SQL，泛指非关系型的数据库。

（2）与关系数据库对比

对比维度	关系数据库	NoSQL
应用领域	面向通用领域	特定应用领域
数据容量	有限数据	海量数据
数据类型	结构化数据【二维表】	非结构化数据
并发支持	支持并发、但性能低	高并发
事务支持	高事务性	弱事务性
扩展方式	向上扩展	向外扩展

（3）分类

分类	典型应用场景	数据模型
键值(key-value)	内容缓存，主要用于处理大量数据的高访问负载，也用于一些日志系统等等	Key 指向 Value 的键值对，通常用 hash table 来实现
列存储数据库	分布式的文件系统	以列簇式存储，将同一列数据存在一起
文档型数据库	Web 应用（与 Key-Value 类似，Value 是结构化的，不同的是数据库能够了解 Value 的内容）	Key-Value 对应的键值对，Value 为结构化数据
图形数据库(Graph)	社交网络，推荐系统等。专注于构建关系图谱	图结构

分类	优点	缺点	举例
键值(key-value)	查找速度快	数据无结构化，通常只被当作字符串或者二进制数据	Redis, Tokyo Cabinet/Tyrant, Voldemort, Oracle BDB
列存储数据库	查找速度快，可扩展性强，更容易进行分布式扩展	功能相对局限	HBase, Cassandra, Riak
文档型数据库	数据结构要求不严格，	查询性能不高，而且缺乏统一	CouchDB, MongoDb

	表结构可变，不需要像关系型数据库一样需要预先定义表结构	的查询语法	
图形数据库(Graph)	利用图结构相关算法。比如最短路径寻址，N度关系查找等	很多时候需要对整个图做计算才能得出需要的信息，而且这种结构不太好做分布式的集群方案	Neo4J, InfoGrid, Infinite Graph

八、反规范化

（1）概念

由于规范化会使表不断的拆分，从而导致数据表过多。这样虽然减少了数据冗余，提高了增、删、改的速度 但会增加查询的工作量。系统需要进行多次连接，才能进行查询操作，使得系统的效率大大的下降。

（2）技术手段

增加冗余列是指在多个表中具有相同的列，它常用来在查询时避免连接操作。

增加派生列指增加的列来自其它表中的数据，由它们计算生成。

重新组表指如果许多用户需要查看两个表连接出来的结果数据，则把这两个表重新组成一个表来减少连接而提高性能。

分割表：水平分割与垂直分割。水平分割，有些记录常要查询，有些记录不常用，如历史记录。垂直分割，把主码与某些常用的字段组成一个表，把主码与另一些字段组成另一个表。

（3）优缺点

反规范化的优点：连接操作少，检索快、统计快；需要查的表减少，检索容易。

九、数据库索引与数据库分区

1、数据库索引：提升查询效率，降低添加、修改、删除效率。采用 B 树，B+树等。

2、分区分表

分区的常见方式：

分区策略	分区方式	说明
范围分区【RANGE】	按数据范围值来做分区	例：按用户编号分区，0-999999 映射到分区 A；1000000-1999999 映射到分区 B。

散列分区【HASH】	通过对 key 进行 hash 运算分区	例：可以把数据分配到不同分区，这类似于取余操作，余数相同的，放在一个分区上。
列表分区【LIST】	根据某字段的某个具体值进行分区	例：长沙用户分成一个区，北京用户分成一个区。

分区的优点：

（1）相对于单个文件系统或是硬盘，分区可以存储更多的数据。

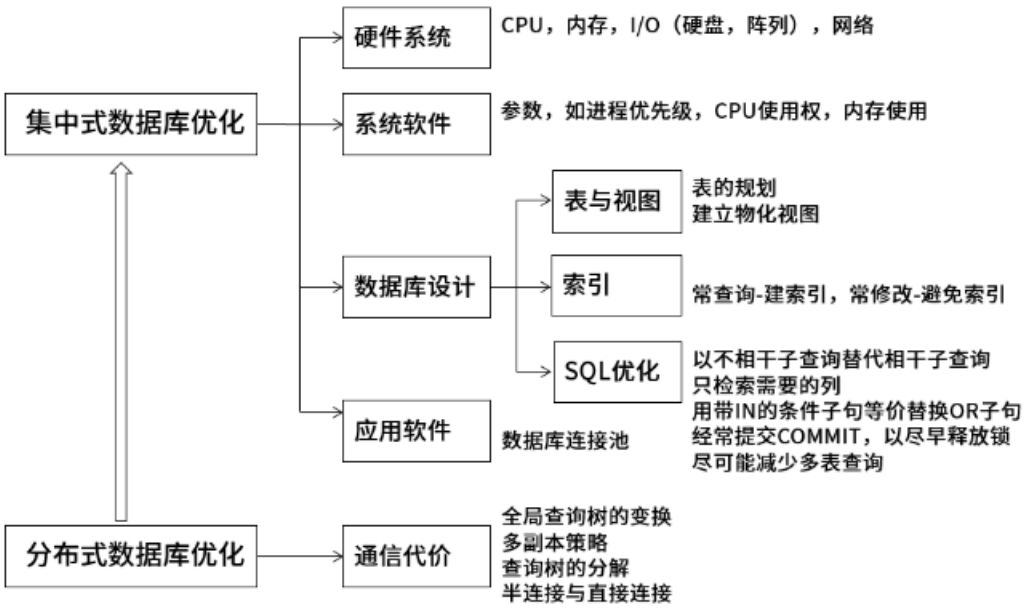
（2）数据管理比较方便，比如要清理或废弃某年的数据，就可以直接删除该日期的分区数据即可。

（3）精准定位分区查询数据，不需要全表扫描查询，大大提高数据检索效率。

（4）可跨多个分区磁盘查询，来提高查询的吞吐量。

（5）在涉及聚合函数查询时，可以很容易进行数据的合并。

十、数据库性能优化



十一、大数据

1、大数据的概述

大数据是指其大小或复杂性无法通过现有常用的软件工具，以合理的成本并在可接受的时限内对其进行捕获、管理和处理的数据集。这些困难包括数据的收入、存储、搜索、共享、分析和可视化。

5 个 V ：大规模(Volume)、高速度(Velocity)、多样化(Variety)、价值密度低（Value）、真实性(Veracity)

大数据的应用领域：制造业的应用、服务业的应用、交通行业的应用、医疗行业的应用等

大数据面临着 5 个主要问题，分别是异构性(Heterogeneity)、规模(Scale)、时间性(Timeliness)、复杂性(Complexity) 和隐私性(Privacy)。

大数据的研究工作将面临 5 个方面的挑战：

挑战一：数据获取问题。

挑战二：数据结构问题。

挑战三：数据集成问题。

挑战四：数据分析、组织、抽取和建模是大数据本质的功能性挑战。

挑战五：如何呈现数据分析的结果，并与非技术的领域专家进行交互。

建议采用现有成熟技术解决大数据带来的挑战，并给出了大数据分析的分析步骤，大致分为数据获取/记录、信息抽取/清洗/注记、数据集成/聚集/表现、数据分析/建模和数据解释 5 个主要阶段。

2、大数据处理系统概述

Hbase：分布式、面向列的开源数据库，适合于非结构化数据存储。【实时数据和离线数据均支持】。

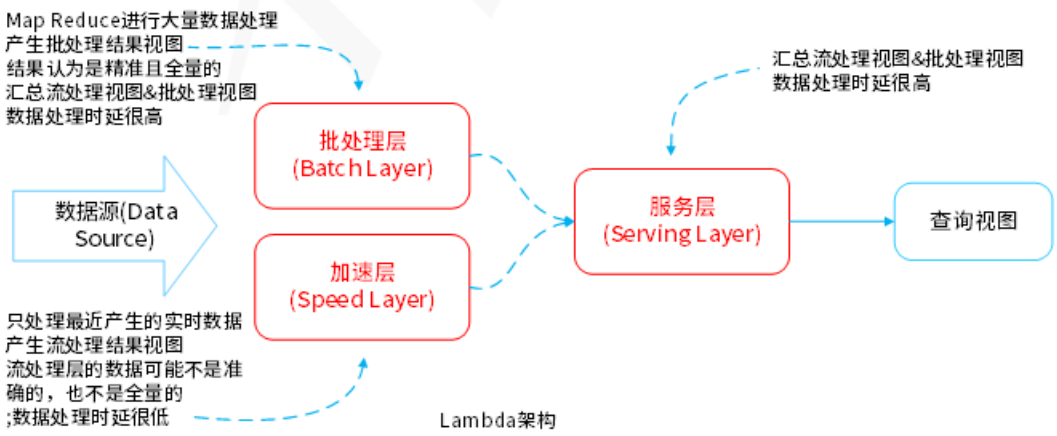
HDFS（Hadoop 分布式文件系统）：适合运行在通用硬件上的分布式文件系统（Distributed File System）。HDFS 是一个高度容错性的系统，适合部署在廉价的机器上。HDFS 能提供高吞吐量的数据访问，非常适合大规模数据集上的应用，【通常用于处理离线数据的存储】。

Flume：高可用/可靠，分布式海量日志采集、聚合和传输的系统，Flume 支持在日志系统中定制各类数据发送方，用于收集数据；同时，Flume 提供对数据进行简单处理，并写到各种数据接受方（可定制）的能力。

Kafka：一种高吞吐量的分布式发布订阅消息系统，它可以处理消费者在网站中的所有动作流数据。

ZooKeeper：开放源码的分布式应用程序协调服务，是 Hadoop 和 Hbase 的重要组件。它是一个为分布式应用提供一致性服务的软件，提供的功能包括：配置维护、域名服务、分布式同步、组服务等。

3、Lambda 架构



Lambda架构

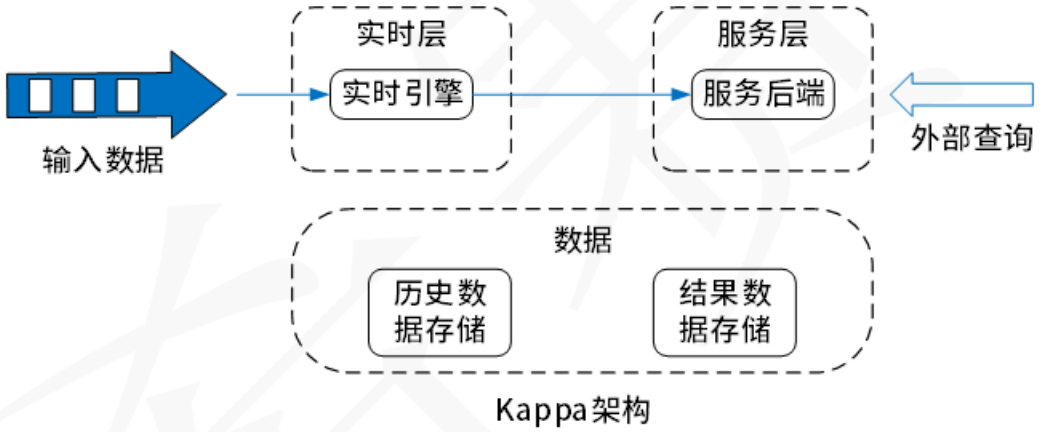
批处理层(Batch Layer)：两个核心功能：存储数据集和生成 Batch View。

加速层(Speed Layer):存储实时视图并处理传入的数据流，以便更新这些视图。

服务层(Serving Layer):用于响应用户的查询请求，合并 Batch View 和 Real-time View 中的结果数据集到最终的数据集。

优点	缺点
(1) 容错性好。 (2) 查询灵活度高。 (3) 易伸缩。 (4) 易扩展。	(1) 全场景覆盖带来的编码开销。 (2) 针对具体场景重新离线训练一遍益处不大。 (3) 重新部署和迁移成本很高。

4、Kappa 架构



Kappa架构

输入数据直接由实时层的实时数据处理引擎对源源不断的源数据进行处理；

再由服务层的服务后端进一步处理以提供上层的业务查询。

而中间结果的数据都是需要存储的，这些数据包括历史数据与结果数据，统一存储在存储介质中。

优点	缺点
(1)将实时和离线代码统一起来了。 (2)方便维护而且统一了数据口径。 (3)避免了 Lambda 架构中与离线数据合并的问题。	(1) 消息中间件缓存的数据量和回溯数据有性能瓶颈。 (2) 在实时数据处理时，遇到大量不同的实时流进行关联时，非常依赖实时计算系统的能力，很可能因为数据流先后顺序问题，导致数据丢失。 (3) Kappa 在抛弃了离线数据处理模块的时候，同时抛弃了离线计算更加稳定可靠的特点。

5、Lambda 架构与 Kappa 架构对比和设计选择

对比内容	Lambda 架构	Kappa 架构
复杂度与开发、维护成本	需要维护两套系统（引擎），复杂度高，开发、维护成本高	只需要维护一套系统（引擎），复杂度低，开发、维护成本低
计算开销	需要一直运行批处理和实时计算，计算开销大	必要时进行全量计算，计算开销相对较小
实时性	满足实时性	满足实时性
历史数据处理能力	批式全量处理，吞吐量大，历史数据处理能力强	流式全量处理，吞吐量相对较低，历史数据处理能力相对较弱
使用场景	直接支持批处理，更适合对历史数据分析查询的场景，期望尽快得到分析结果，批处理可以更直接高效地满足这些需求。	不是 Lambda 的替代架构，而是简化，Kappa 放弃了对批处理的支持，更擅长业务本身为增量数据写入场景的分析需求
选择依据	根据两种架构对比分析，将业务需求、技术要求、系统复杂度、开发维护成本和历史数据处理能力作为选择考虑因素。 计算开销虽然存在一定差别，但是相差不是很大，所以不作为考虑因素。	

第四章 企业信息化战略与实施

一、信息和信息化的概念

1、什么是信息

维纳(Norbert Wiener)：信息就是信息，既不是物质也不是能量，但信息可转换为物质或能量。

香农(Claude E.Shannon)：信息就是不确定性的减少。

2、信息的基本属性

信息具有如下基本属性：

真伪性：真实是信息的中心价值，不真实的信息价值可能为负。

层次性：信息一般和管理层一样，可以为战略层、策略层和执行层 3 个层次。

不完全性：客观事实的全部信息是不可能得到的。我们需要正确滤去不重要的信息、失真的信息，抽象出有用的信息。

滞后性：信息是数据加工的结果，因此信息必然落后于数据，加工需要时间。

扩压性：信息和实物不同，它可以扩散也可以压缩。

分享性：信息可以分享，这和物质不同，并且信息分享具有非零和性。

动态性：信息随着时间的变化而变化。

传递性：信息在时间上的传递即是存储；在空间上的传递即是转移或扩散。

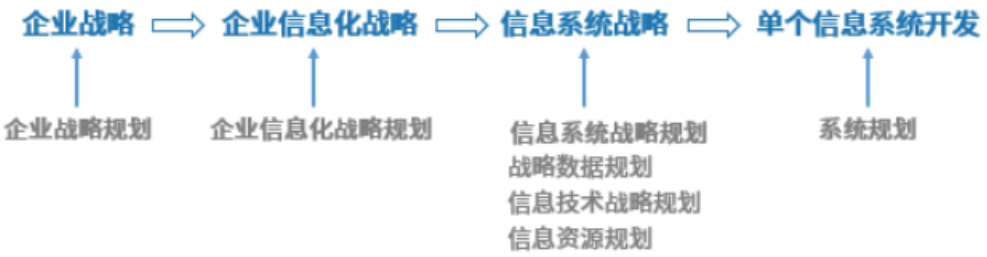
3、信息化的概念

信息化是从工业社会到信息社会的演进与变革。

信息化的主体是全体社会成员（政府、企业、团体和个人），时域是一个长期过程，空域是经济和社会的一切领域，手段是先进社会生产工具。

二、信息化战略体系

1、信息化战略体系规划



企业战略规划是用机会和威胁评价现在和未来的环境，用优势和劣势评价企业现状，进而选择和确定企业的总体和长远目标，制定和抉择实现目标的行动方案。

希赛解读：确定企业未来发展的大方向。

信息系统战略规划关注的是如何通过信息系统来支撑业务流程的运作，进而实现企业的关键业务目标，其重点在于对信息系统远景、组成架构、各部分逻辑关系进行规划。

希赛解读：为企业战略开发支撑系统。

信息技术战略规划通常简称为 IT 战略规划，是在信息系统规划的基础上，对支撑信息系统运行的硬件、软件、支撑环境等进行具体的规划，它更关心技术层面的问题。

希赛解读：为支撑系统运行环境做规划。

信息资源规划是在以上规划的基础上，为开展具体的信息化建设项目而进行的数据需求分析、信息资源标准建立、信息资源整合工作。

希赛解读：数据与标准相关的规划

系统规划是信息系统生命周期的第一个阶段，其任务是对企业的环境、目标及现有系统的状况进行初步调查，根据企业目标和发展战略，确定信息系统的发展战略，对建设新系统的需求做出分析和预测，同时考虑建设新系统所受的各种约束，研究建设新系统的必要性和可能性。

希赛解读：单个项目的立项分析。

2、企业战略与信息化战略集成方法：

业务与 IT 整合（BITA）：重心是找业务与现有 IT 系统之间的不一致，并给出转变计划。【业务路线】

企业 IT 架构（EITA）：帮助 IT 企业建立 IT 的原则规范、模式和标准。【IT 技术路线】

3、信息系统战略规划（ISSP）

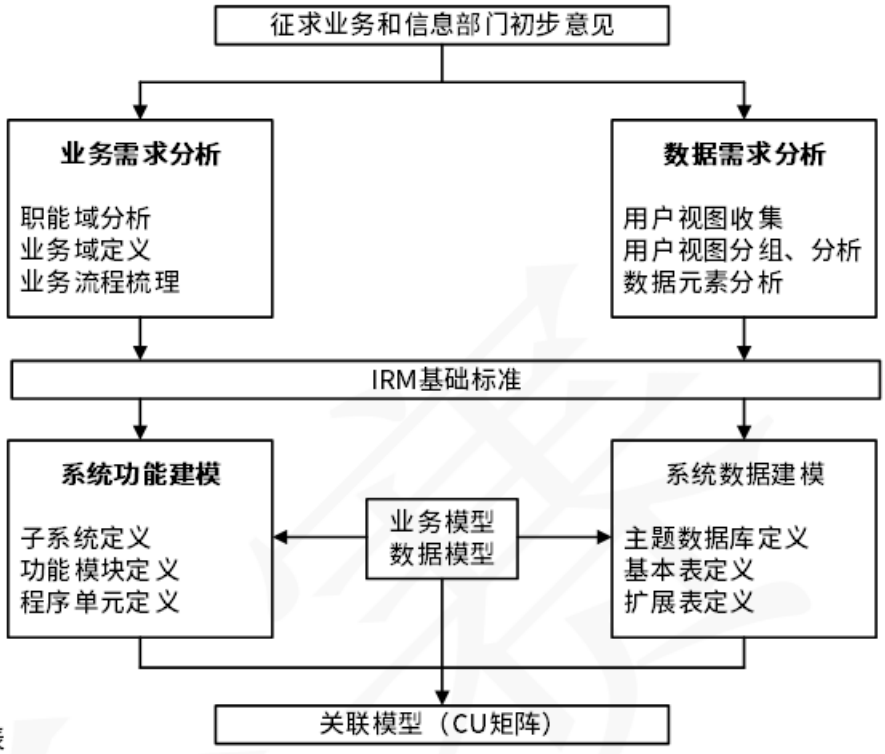
信息系统战略规划（Information System Strategic Planning，ISSP）是从企业战略出发，构建企业基本的信息架构，对企业内、外信息资源进行统一规划、管理与应用，利用信息控制企业行为，辅助企业进行决策，帮助企业实现战略目标。

ISSP 方法经历了三个主要阶段，各个阶段所使用的方法也不一样。

第一个阶段主要以数据处理为核心，围绕职能部门需求的信息系统规划，主要的方法包括企业系统规划法（BSP）--CU 矩阵、关键成功因素法（CSF）和战略集合转化法（SST），此外还有：投资回收法、征费法、零线预算法、阶石法；

第二个阶段主要以企业内部管理信息系统为核心，围绕企业整体需求进行的信息系统规划，主要的方法包括战略数据规划法（SDP）：主题数据库、信息工程法（IE）和战略栅格法（SG）；

第三个阶段的方法在综合考虑企业内外环境的情况下，以集成为核心，围绕企业战略需求进行的信息系统规划，主要的方法包括价值链分析法（VCA）和战略一致性模型（SAM）。



4规范化表

三、业务流程分析方法

1、概念

业务流程分析的目的是了解各个业务流程的过程，明确各个部门之间的业务关系和每个业务处理的意义。企业业务流程包含三个要素，分别是实体、对象和活动。

2、分析方法

价值链分析法：找出或设计出那些能够使顾客满意，实现顾客价值最大化的业务流程。

客户关系分析法：把 CRM 用在业务流程的分析上。

供应链分析法：从企业供应链的角度分析企业的业务流程。

基于 ERP 的分析法：将企业的业务流程看作是一个紧密联接的供应链，将供应商和企业内部的采购、生产、销售，以及客户紧密联系起来，对供应链上的所有环节进行有效管理，实现对企业的动态控制和各种资源的集成和优化。

业务流程重组：通过重新审视企业的价值链，从功能成本的比较分析中，确定企业在哪些环节具有比较优势。

3、业务流程建模

标杆瞄准【“拷贝”优秀企业成功经验】

IDEF【一系列建模、分析和仿真方法的统称】

- IDEF0：业务流程（功能）建模；
- IDEF1：信息建模；
- IDEF1X：数据建模（如 ER 模型）；
- IDEF4：面向对象设计；
- IDEF8：用户界面建模；

DEMO【行为角色之间的通信方式】

Petri 网【业务流：并行、异步、分布式和随机性】

基于服务的 BPM

4、业务流程重组 BPR

BPR 是对企业的业务流程进行根本性的再思考和彻底性的再设计，从而获得可以用诸如成本、质量、服务和速度等方面的业绩来衡量的显著性的成就。

5、业务流程管理 BPM

BPM 是一种以规范化的构造端到端的卓越业务流程为中心，以持续的提高组织业务绩效为目的的系统化方法。

PDCA 闭环的管理过程：明确业务流程所欲获取的成果；开发和计划系统的方法，实现以上成果；系统地部署方法，确保全面实施；根据对业务的简称和分析以及持续的学习活动，评估和审查所执行的方法，并进一步提出计划和实施改进措施。

四、电子政务

电子政务主要有 3 类角色：政府（Government）、企（事）业单位（Business） 及 公民（Citizen）。如果有第 4 类就是公务员（Employee）。

类型	应用
G2G	基础信息的采集、处理和利用，如：人口信息、地理信息 各级政府决策支持
G2E	政府内部管理系统

G2B	政府给企业单位颁发【**各种营业执照、许可证、合格证、质量认证**】
B2G	企业向政府**缴税** 企业向政府供应各种商品和服务【含竞/投标】 企业向政府**提建议，申诉**
G2C	社区公安和水、火、天灾等与公共安全有关的信息 **户口、各种证件和牌照的管理**
C2G	个人应向政府缴纳的各种税款和费用 个人向政府反馈民意【**征求群众意见**】 报警服务【盗贼、医疗、急救、火警等】

五、企业信息化与电子商务

1、企业资源计划（ERP）

（1）发展过程

（2）ERP 结构

ERP 是将企业所有资源（企业三大流：物流、资金链、信息流）进行集成整合，全面一体化管理的管理信息系统。

包括三方面：生产控制（计划、制造）、物流管理（分销、采购、库存管理）和财务管理（会计核算、财务管理）。这三个系统本身就是一个集成体，它们相互之间有相应的接口，能够很好地整合在一起。

管理思想：他是管理思想的变革。

软件产品：但不是直接买来就用，需要个性化的开发与部署。

管理系统：存在众多的子系统，这些子系统有统一的规划，是互联互通的，便于事前事中监控。

2、客户管理（CRM）

(1)CRM 理念：将客户看作资产；客户关怀是中心，CRM 的目的是提高收入。CRM 的核心思想就是以客户为中心。

(2) CRM 的主要模块：

- 销售自动化；
- 营销自动化；
- 客户服务与支持；
- 商业智能。

CRM 的价值：

提高工作效率，节省开支；

提高客户满意度；

提高客户的忠诚度。

3、供应链管理（SCM）

SCM 理念：强强联合，整合与优化“三流”（信息流-需求信息流、供应信息流、资金流、物流），打通企业间“信息孤岛”，严格的数据交换标准。将制造商、供应商、分销商、零售商，在计划（策略性）、采购、制造、配送、退货等各方面联系起来。

信息流需要进一步了解其分类：

需求信息流：如客户订单、生产计划、采购合同等

供应信息流：如入库单、完工报告单、库存记录、可供销售量、提货发运单等

4、商业智能（BI）

BI=数据仓库+数据挖掘+OLAP，用途：决策分析【分析历史数据预判未来】

(1)数据仓库

数据仓库的特点：面向主题；集成的；相对稳定的（非易失的）；反映历史变化（随着时间变化）。

（2）数据湖：

数据湖是一个存储企业的各种各样原始数据的大型仓库，其中的数据可供存取、处理、分析及传输。

数据仓库和数据湖的区别：

数据仓库仅支持数据分析处理。

数据湖既支持数据分析处理，也支持事务处理。

维度	数据仓库	数据湖
数据	清洗过的数据 结构化的数据	原始数据 结构化，半结构化数据
模式	数据存储之前定义数据模式 数据集成之前完成大量工作 数据的价值提前明确	数据存储之后定义数据模式 提供敏捷、简单的数据集成 数据的价值尚未明确
存取方法	标准 SQL 接口	应用程序，类 SQL 的程序
优势	多数据源集成、干净、安全的数据、 转换一次，多次使用	无限扩展性、并行执行、支持编程框架、 数据经济

(3)数据挖掘

数据挖掘是从数据本身出发，挖掘人类所未知的知识内容。

数据挖掘方法分类：

关联分析：挖掘出隐藏在数据间的相互关系。

序列模式分析：侧重点是分析数据间的前后关系（因果关系）。

分类分析：为每一个记录赋予一个标记再按标记分类。

聚类分析：分类分析法的逆过程。

（4）数据治理

数据治理是对数据资产管理行使权力和控制的活动集合。

数据治理模型包括三个框架：范围、促成因素和执行及评估：

范围：展示了我们应该关注什么

促成因素：展示了数据治理的推动因素

执行和评估：展示了如何实现治理的方法。

数据治理的流程：

数据规划：梳理业务流程，规划数据资源，如需要采集哪些数据，放哪里，如何放等方面的规划。

数据采集：ETL 采集、去重、脱敏、转换、关联、去除异常值。

数据存储管理：大数据高性能存储及管理。

数据应用：即时查询、报表监控、智能分析、模型预测。

5、知识管理

（1）知识分类

显性知识（explicit knowledge）：以文字与数字来表达，以资料、科学法则、特定规格及手册等形式展现者。利于传承、传播。

隐性知识（tacit knowledge）：个人主观的洞察力、直觉与预感等。不利于传承、传播。如：个人经验。

显性知识：规范，系统，结构化，明确

隐性知识：未规范，零星，非正式，未编码，不易保存、传递

（2）知识管理工具

可以把知识管理工具分为知识生成工具、知识编码工具和知识转移工具三大类

知识生成工具：知识获取、知识合成、知识创新

知识编码工具：通过标准形式表现知识

知识转移工具：使知识能在企业内传播和分享

六、企业应用集成

不同维度的集成划分

	集成点	效果	解题关键点
界面集成	界面层	统一入口，产生“整体”感觉	“整体”感觉 最小代价实现一体化操作
数据集成	数据层	不同来源的数据逻辑或物理上“集中”	其他集成方法的基础
控制集成	应用逻辑层	调用其它系统已有方法，达到集成效果	
业务流程集成（过程集成）	应用逻辑层	跨企业，或优化流程而非直接调用	企业之间的信息共享能力
门户集成		将内部系统对接到互联网上	发布到互联网上

	特点
消息集成	数据量小，交互频繁，立即地，异步
共享数据库	交互频繁，立即地，同步
文件传输	数据量大，交互频度小，即时性要求低（月末，年末）

七、企业门户

企业信息门户（EIP，Enterprise Information Portal）：使员工/合作伙伴/客户/供应商都能够访问企业内部网络和因特网存储的各种自己所需的信息。

企业知识门户（EKP，Enterprise Knowledge Portal）：在企业网站的基础上增加知识性内容。

企业应用门户（EAP，Enterprise Application Portal）：以商业流程和企业应用为核心，把商业流程中功能不同的应用模块通过门户技术集成在一起。

垂直门户：为某一特定的行业服务的，传送的信息只属于人们感兴趣的领域。

八、电子商务分类

电子商务主要有 2 类角色：企业（Business） 及 个人（Customer）。

主要了解电子商务的类型：

企业对消费者（B2C）：京东，当当，天猫，亚马逊。

企业对企业（B2B）：阿里巴巴，慧聪网。

消费者对消费者（C2C）：闲鱼。

消费者对企业（C2B）：个人给企业提供咨询服务

线上对线下（O2O）：团购

九、数字化转型

1、数字化

（1）数字化是新一代信息技术真正的实现推动整个商业模式的变革，推动产业链的重构，推动改进企业与消费者之间的关系，以及企业与合作伙伴之间的关系。

（2）企业数字化转型的五个发展阶段

初始级发展阶段【数码化】：信息的数字化，记录、储存、传输数码化。

单元级发展阶段【数量化】：提升单项业务的运行规范性和效率。

流程级发展阶段【数字化】：关键业务流程及关键业务与设备设施、软硬件、行为活动等要素间的集成优化。

网络级发展阶段【数模化】：组织（企业）级数字化和产业互联网级网络化，实现以数据为驱动的业务模式创新。

生态级发展阶段【数用化】：生态级数字化和泛在物联网级网络化，推动与生态合作伙伴间资源、业务、能力等要素的开放共享和协同合作。

2、智能制造体系

系统层级

设备层：传感器、仪器仪表、机器、装置等

单元层：企业内处理信息、实现监测和控制物理流程的层级

车间层：面向工厂或车间的生产管理的层级

企业层：面向企业经营管理的层级

协同层：其内部和外部信息互联和共享，实现跨企业间业务协同的层级

制作于 23 年 12 月 适用于第 1 版教材