

10-B1

高级搜索树

B-树：大数据

640K ought to be enough for anybody.

- B. Gates, 1981

白嘉轩一听就不由得火了：“又是个百日忌讳！”仙草却说：“百日又不是百年。你权当百日后才娶我。你就忍一忍，一百天很快就过去了.....”

邓俊辉

deng@tsinghua.edu.cn

越来越小的内存

❖ RAM : 存储器？不就是无限可数个寄存器吗？

Turing : 存储器？不就是无限长的纸带吗？

❖ 但事实上

系统存储容量的增长速度

<< 应用问题规模的增长速度

1 Kilobyte = $2^{10} = 10^3$

1 Megabyte = $2^{20} = 10^6$

1 Gigabyte = $2^{30} = 10^9$

1 Terabyte = $2^{40} = 10^{12}$

1 Petabyte = $2^{50} = 10^{15}$

1 Exabyte = $2^{60} = 10^{18}$

2010 1 Zettabyte = $2^{70} = 10^{21}$

1 Yottabyte = $2^{80} = 10^{24}$

1 Nonabyte = $2^{90} = 10^{27}$

1 Doggabyte = $2^{100} = 10^{30}$

越来越小的内存

❖ 典型的 $1980 : 10\text{MB} / 1\text{MB} = 10$

数据库规模/内存容量 2010 : 1TB / 1GB = **1000**

❖ 今天典型的数据集以TB度量；相对而言，内存容量在不断减小！

❖ 内存为何不能更大？存储器容量越大/小，访问速度越慢/快

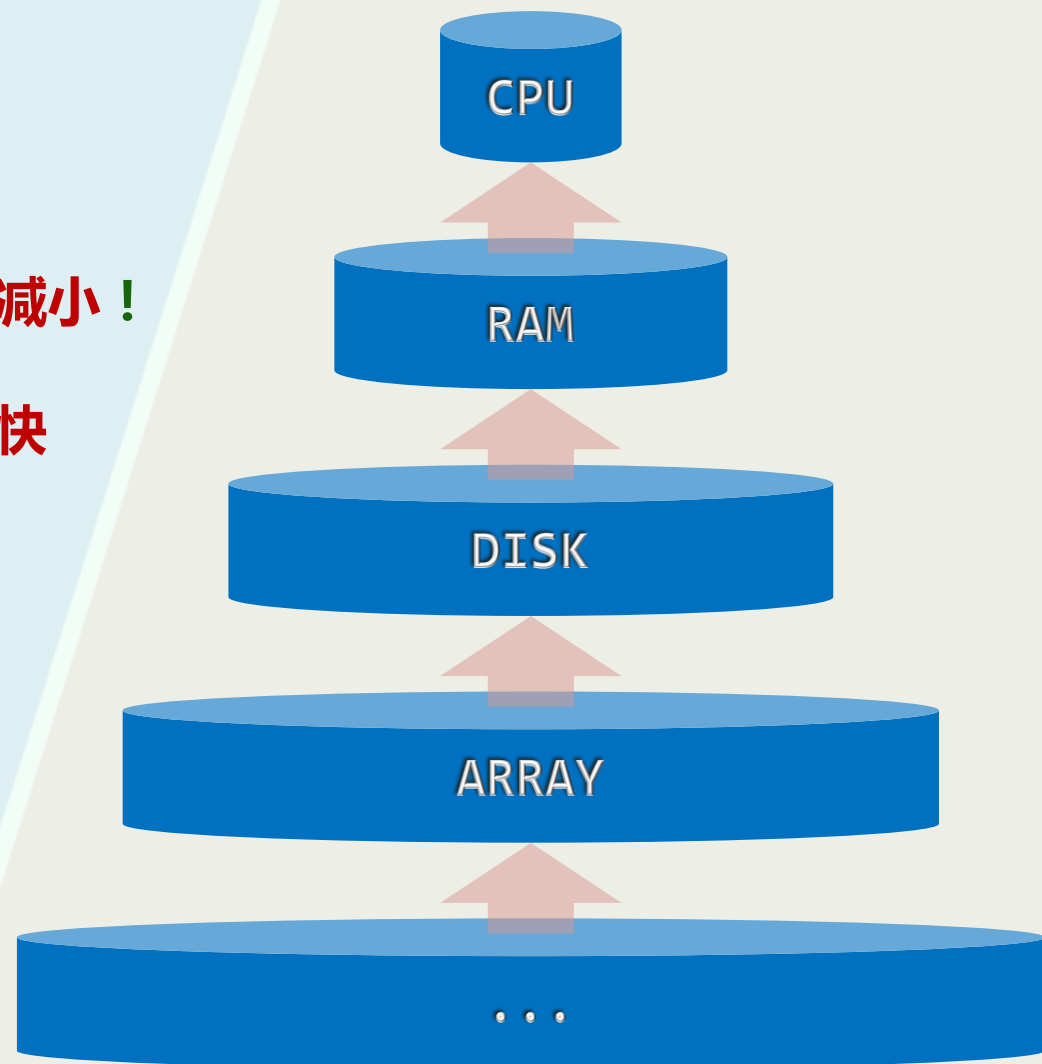
❖ 事实1：不同容量的存储器，访问速度差异悬殊

以磁盘与内存为例： $\text{ms/ns} > 10^5$

❖ 若一次内存访问需要一秒，则一次磁盘访问就需一天

为避免一次磁盘访问，我们宁愿访问内存1000次

❖ 实用的存储系统，多为分级结构...



分级存储

❖ 机制与策略

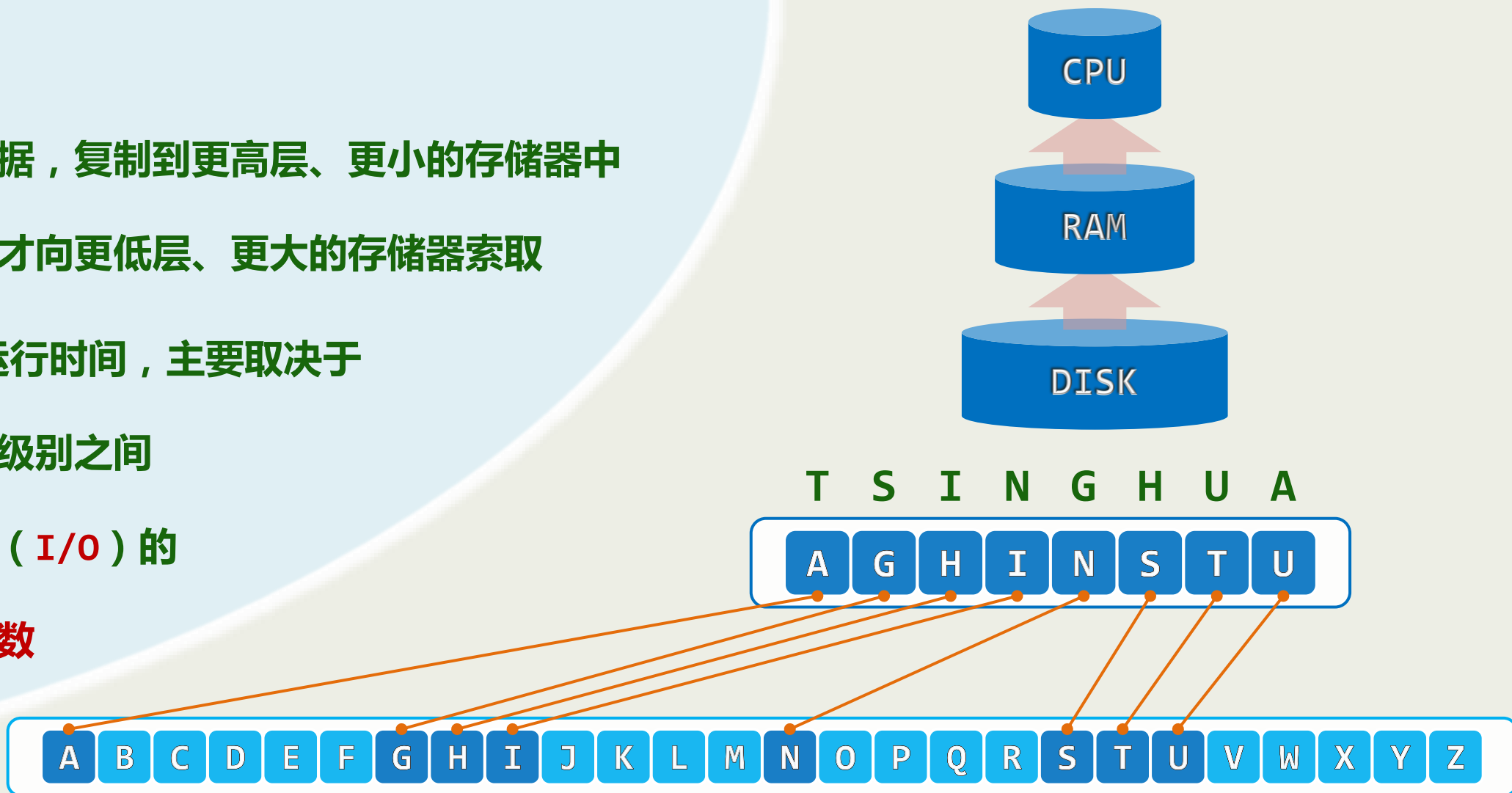
- 常用的数据，复制到更高层、更小的存储器中
- 找不到，才向更低层、更大的存储器索取

❖ 算法的实际运行时间，主要取决于

相邻存储级别之间

数据传输 (I/O) 的

速度与次数



缓存：从磁盘中读写1B，与读写1KB几乎一样快

❖ 批量式访问：以页（page）或块（block）为单位，使用缓冲区

```
// <stdio.h>
```

```
#define BUFSIZ 512 //缓冲区默认容量
```

```
int setvbuf( //定制缓冲区
```

```
FILE* fp, //流
```

```
char* buf, //缓冲区
```

```
int _Mode, //_IOFBF | _IOLBF | _IONBF
```

```
size_t size); //缓冲区容量
```

```
int fflush( FILE* fp ); //强制清空缓冲区
```

❖ 效果：单位字节的平均访问时间大大缩短

