第二部分 分布式算法

汪炀

中国科学技术大学计算机系

国家高性能计算中心（合肥）

Ch.1 导论

§1.1 分布式系统

- **Def：一个分布式系统是一个能彼此通信的单个计算装置的集合（计算单元：硬——处理器；软——进程）**

  包括：紧耦合系统----如共享内存多处理机

  松散系统-----cow、Internet

- **与并行处理的分别(具有更高程度的不确定性和行为的独立性)**
  - **并行处理的目标是使用所有处理器来执行一个大任务**
  - **而分布式系统中，每个处理器一般都有自己独立的任务，但由于各种原因（为共享资源，可用性和容错等），处理机之间需要协调彼此的动作。**

- **分布式系统无处不在，其作用是：**

  ①共享资源

  ②改善性能：并行地解决问题

  ③改善可用性：提高可靠性，以防某些成分发生故障

§1.1 分布式系统

分布式系统软件实例简介

- **ElcomSoft Distributed Password Recovery是一款俄罗斯安全公司出品的分布式密码暴力破解工具**
- **能够利用Nvidia显卡使WPA和WPA2无线密钥破解速度提高100倍**
- **还允许数千台计算机联网进行分布式并行计算**

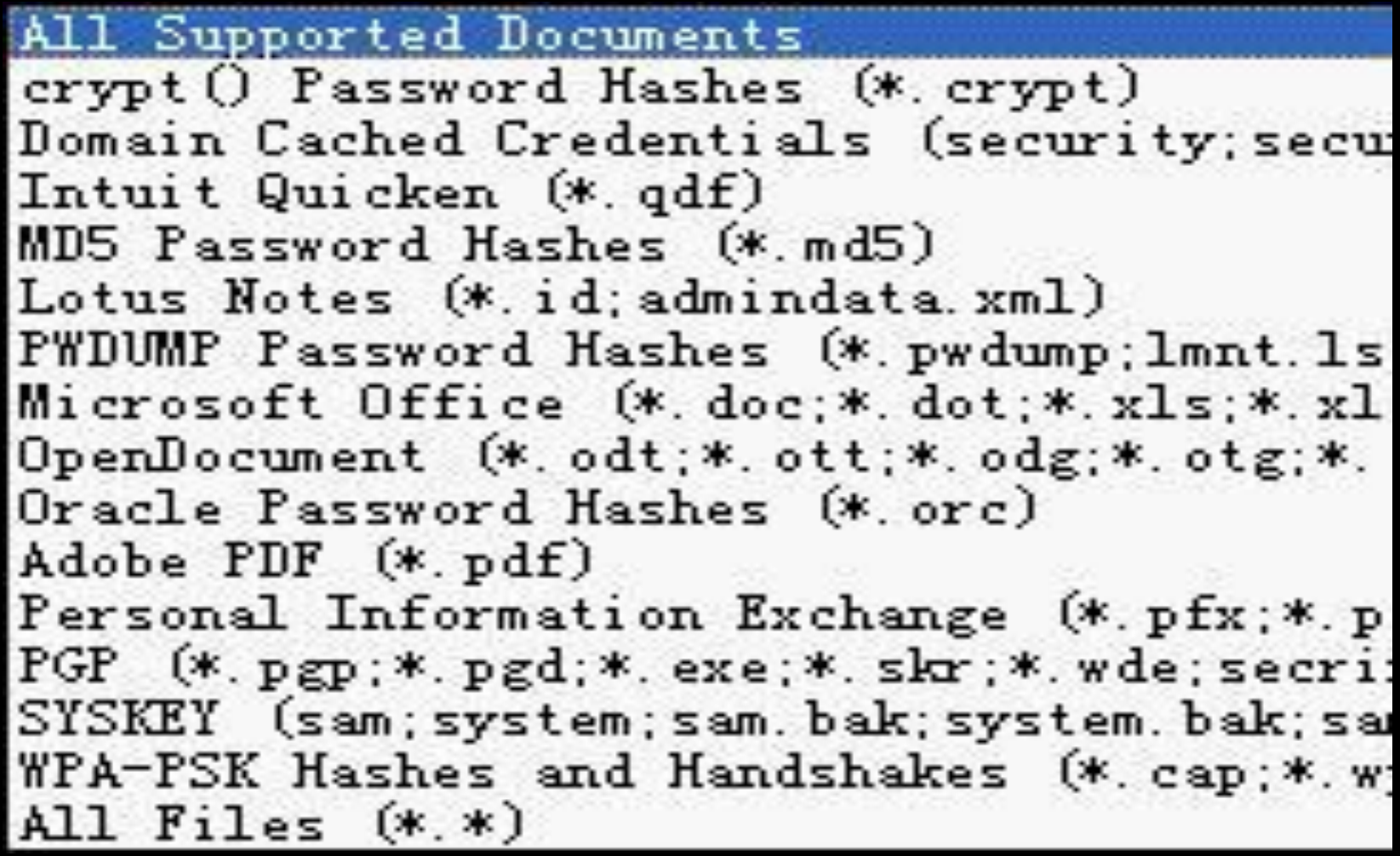
§1.1 分布式系统
系统适用范围

- ElcomSoft 的密码恢复软件主要是面向Office，包括（Word, Excel, Access, Outlook, Outlook Express, VBA, PowerPoint and Visio)

- 其他的面向微软的产品有（Project, Backup, Mail, Schedule+), archive products (including ZIP, RAR, ACE and ARJ files)等

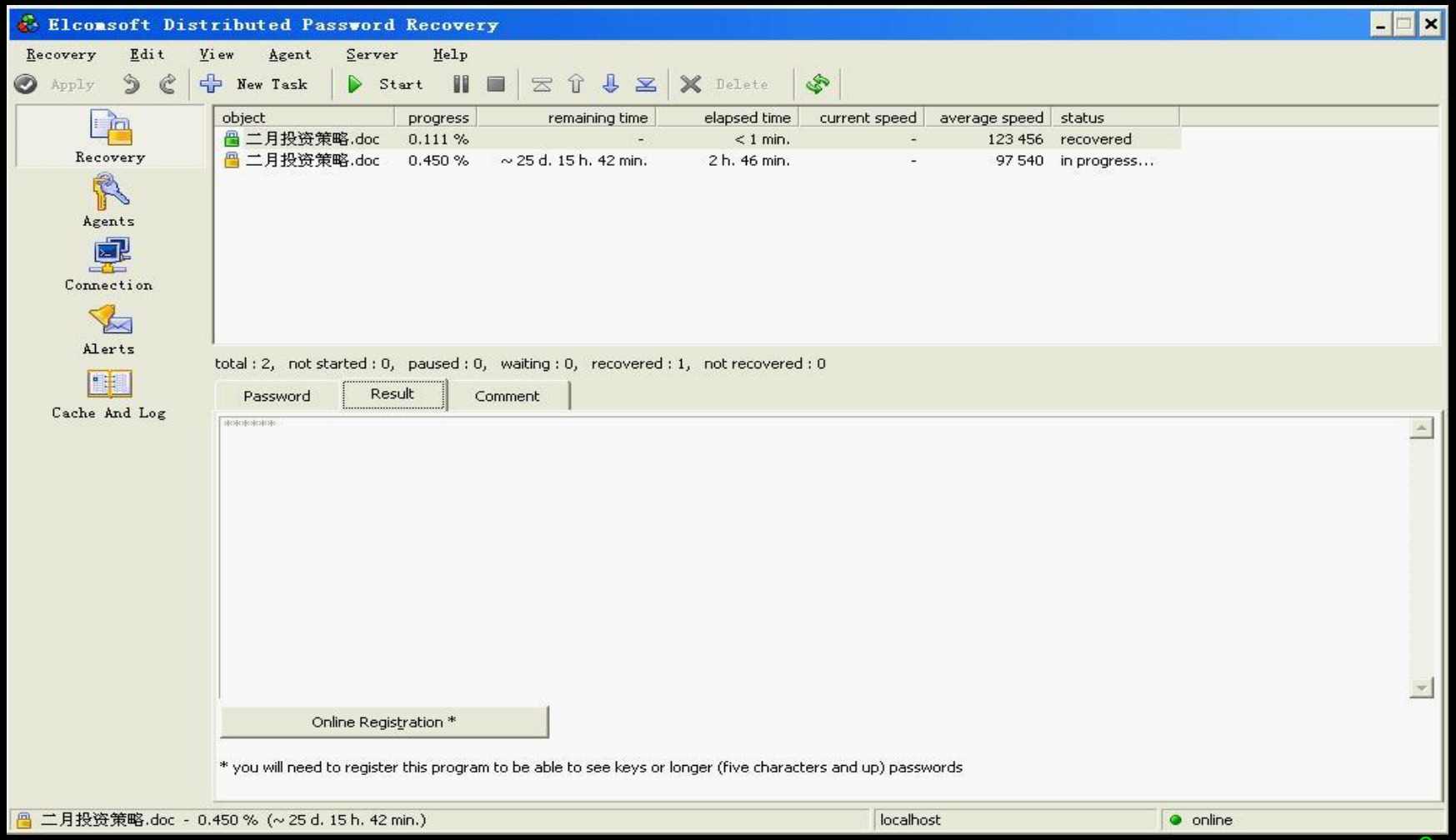
§1.1 分布式系统

演示界面-支持的文件类型



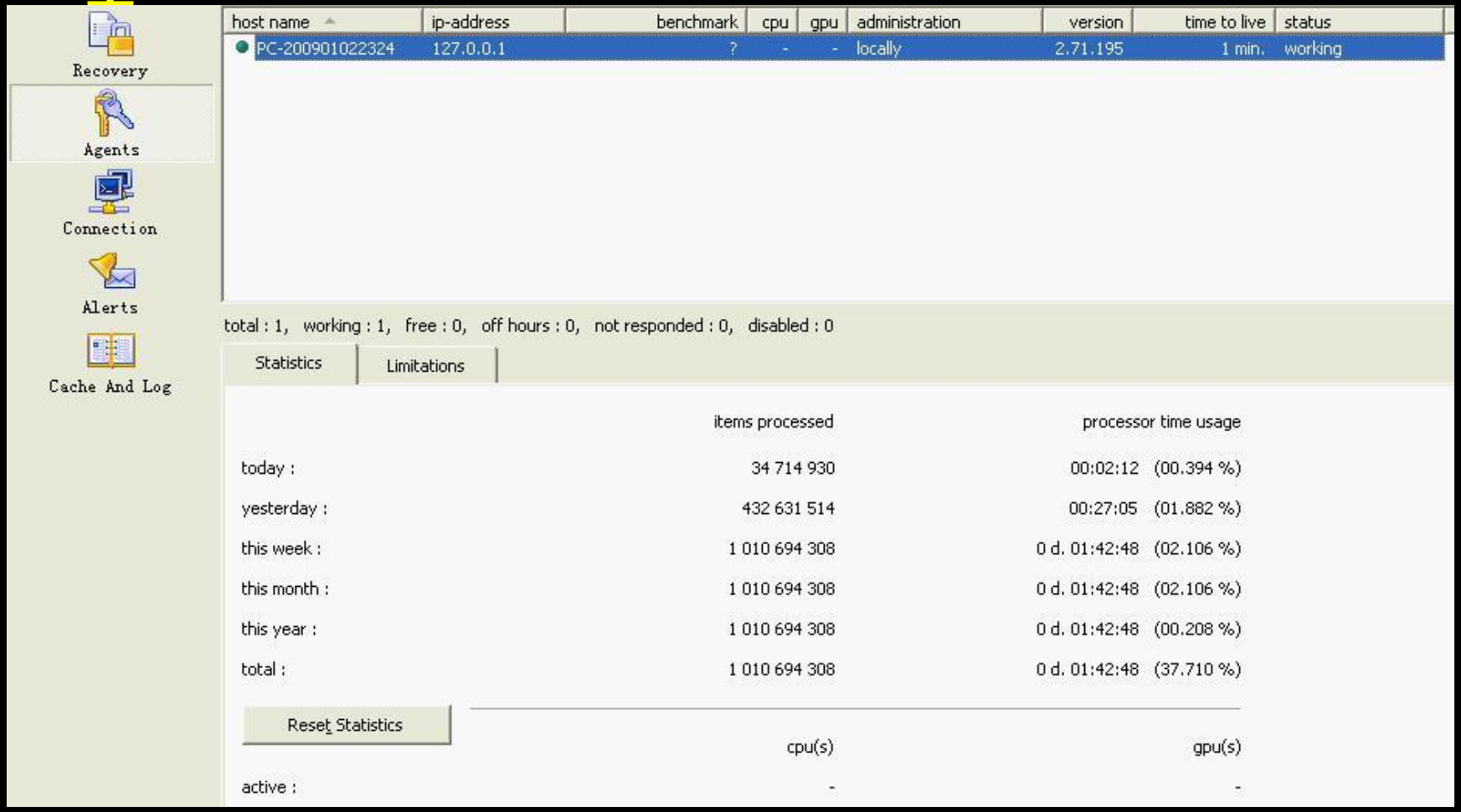
§1.1 分布式系统

演示一主界面



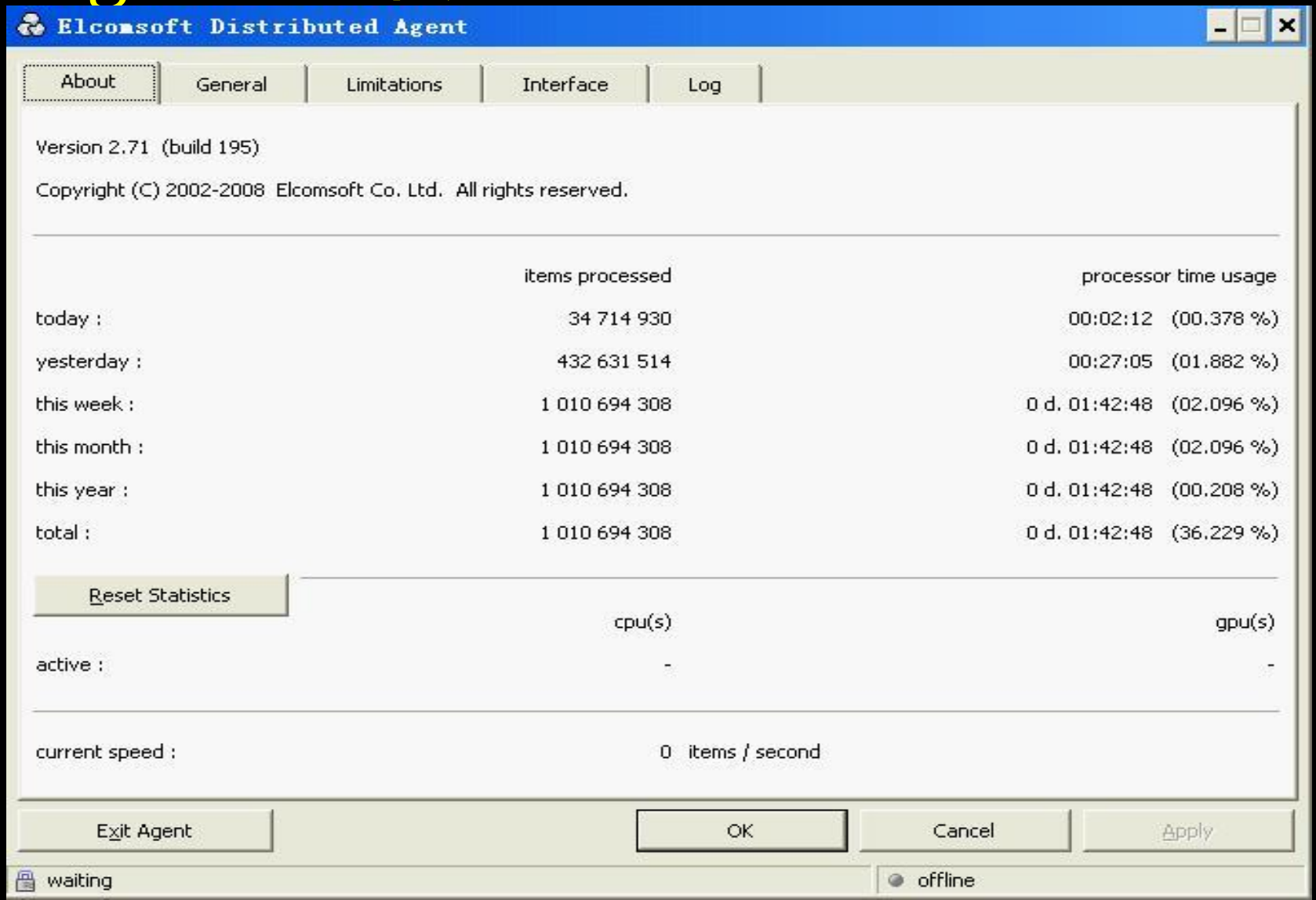
§1.1 分布式系统
最终破解效果

- DOC加密的文档，8位数字型密码小于1分钟即可成功解



§1.1 分布式系统

Agents工作界面



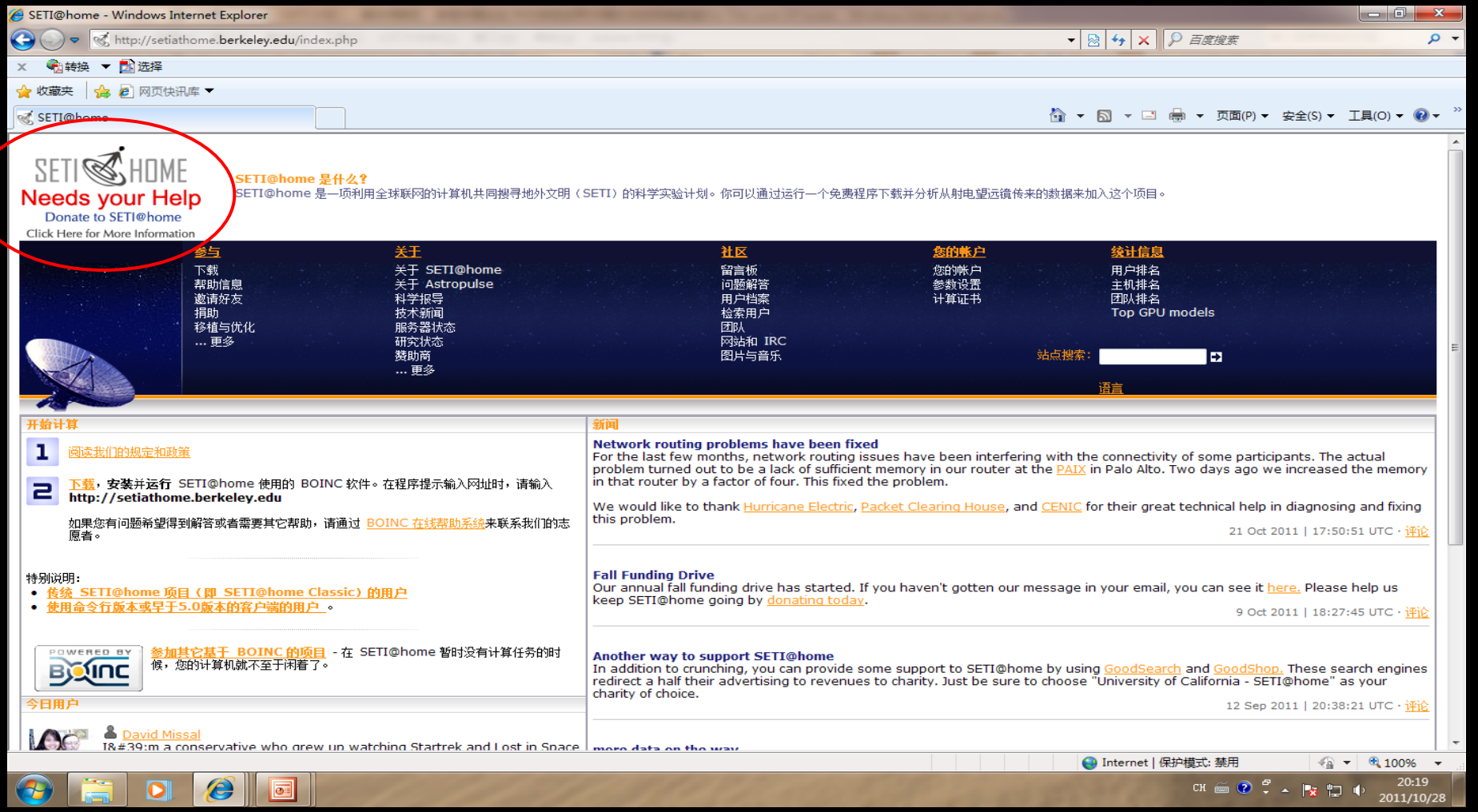
§1.1 分布式系统

NASA SETI寻找外星人计划

- SETI (搜寻外星智慧) 是一个寻找地球外智慧生命的科学性实验计划，使用射电望远镜来监听太空中的窄频无线电讯号。假设这些讯号中有些不是自然产生的，那么只要我们侦测到这些讯号就可以证明外星科技的存在。
- 射电望远镜讯号主要由噪声 (来自天体的发射源与接收者的电子干扰) 与像电视转播站、雷达和卫星等等的人工讯号所组成。现代的 Radio SETI 计划会分析这些数字信息。有更强大的运算能力就可以搜寻更广泛的频率范围以及提高灵敏度。因此，Radio SETI 计划对运算能力的需求是永无止尽的。
- 原来的 SETI 项目曾经使用望远镜旁专用的超级计算机来进行大量的数据分析。1995年，David Gedye 提议射电 SETI 使用由全球联网的大量计算机所组成的虚拟超级计算机来进行计算，并创建了 SETI@home 项目来实验这个想法。SETI@home 项目于1999年5月开始运行。

§1.1 分布式系统

NASA SETI寻找外星人计划



§1.1 分布式系统

■分布式系统面临的困难

- ❖**异质性：** 软硬件环境
- ❖**异步性：** 事件发生的绝对、甚至相对时间不可能总是精确地知道
- ❖**局部性：** 每个计算实体只有全局情况的一个局部视图
- ❖**故障：** 各计算实体会独立地出故障，影响其他计算实体的工作。

§1.2 分布式计算的理论

- **目标：** 针对分布式系统完成类似于顺序式计算中对算法的研究
  - **具体：** 对各种分布式情况发生的问题进行抽象，精确地陈述这些问题，设计和分析有效算法解决这些问题，证明这些算法的最优性。
- **计算模型：**
  - **通信：** 计算实体间msg传递还是共享变量？
  - 哪些计时信息和行为是可用的？
  - 容许哪些错误
- **复杂性度量标准**
  - 时间，空间
  - 通信成本：msg的个数，共享变量的大小及个数
  - 故障和非故障的数目

§1.2 分布式计算的理论

■否定结果、下界和不可能的结果

常常要证明在一个特定的分布式系统中，某个特定问题的不可解性。

就像NP-完全问题一样，表示我们不应该总花精力去求解这些问题。

当然，可以改变规则，在一种较弱的情况下去求解问题。

■我们侧重研究：

- ❖可计算性：问题是否可解？
- ❖计算复杂性：求解问题的代价是什么？

§1.3 理论和实际之关系

主要的分布式系统的种类，分布式计算理论中常用的形式模型之间的关系

- 种类
  - ❖支持多任务的OS：互斥，死锁检测和防止等技术在分布式系统中同样存在。
  - ❖MIMD机器：紧耦合系统，它由分离的硬件运行共同的软件构成。
  - ❖更松散的分布式系统：由网络（局域、广域等）连接起来的自治主机构成

    特点是由分离的硬件运行分离的软件。实体间通过诸如TCP/IP栈、CORBA或某些其它组件或中间件等接口互相作用。

§1.3 理论和实际之关系

- **模型**

模型太多。这里主要考虑三种，基于通信介质和同步程度考虑。

① **异步共享存储模型**：用于紧耦合机器，通常情况下各处理机的时钟信号不是来源于同一信号源

② **异步msg传递模型**：用于松散耦合机器及广域网

③ **同步msg传递模型**：这是一个理想的msg传递系统。该系统中，某些计时信息（如msg延迟上界）是已知的，系统的执行划分为轮执行，是异步系统的一种特例。

该模型便于设计算法，然后将其翻译成更实际的模型。

- Dijkstra E W. Co-operating Sequential Process. In programming Language. F. Genyus(ed.). [S.I.]: Academic Press, 1968, 43-112;
- Owicki S, Gries D. Verifying Properties of Parallel Programs: An Axiomatic Approach. Communication ACM 19, 5(1976), 279-285;

§1.3 理论和实际之关系

- 错误的种类
  - 初始死进程

    指在局部算法中没有执行过一步。
  - Crash failure崩溃错误(损毁模型)

    指处理机没有任何警告而在某点上停止操作。
  - Byzantine failure拜占庭错误

    一个出错可引起任意的动作, 即执行了与局部算法不一致的任意步。拜占庭错误的进程发送的消息可能包含任意内容。

Ch.2 消息传递系统中的基本算法

本章介绍无故障的msg传递系统，考虑两个主要的计时模型：同步及异步。
定义主要的复杂性度量、描述伪代码约定，最后介绍几个简单算法

§2.1 消息传递系统的形式化模型

§2.1.1 系统

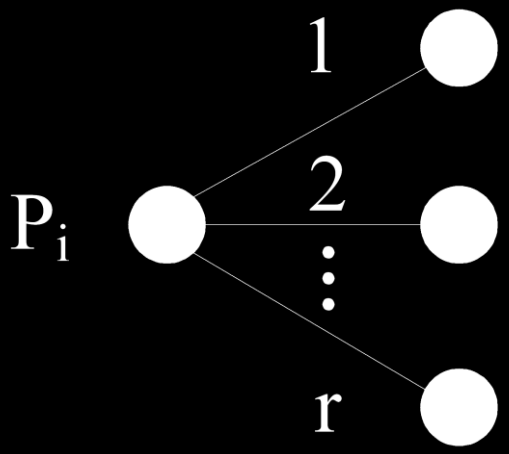
1.基本概念

- 拓扑：无向图 结点——处理机
  边 ——双向信道

§2.1.1 系统

- 算法：由系统中每个处理器上的局部程序构成
  - 局部程序
    - 执行局部计算——本地机器
    - 发送和接收msg——邻居
  - 形式地：一个系统或一个算法是由n个处理器p_0，p_1,…p_{n-1}构成，每个处理器p_i可以模型化为一个具有状态集Q_i的状态机（可能是无限的）

§2.1.1 系统



- 状态（进程的局部状态）

  由p_i的变量，p_i的msgs构成。

  p_i的每个状态由2r个msg集构成：

  - $outbuf_i[l](1 \le l \le r)$：$p_i$经第$l$条关联的信道发送给邻居，但尚未传到邻居的msg。
  - $inbuf_i[l](1 \le l \le r)$：在$p_i$的第$l$条信道上已传递到$p_i$，但尚未经$p_i$内部计算步骤处理的msg。

  模拟在信道上传输的msgs

§2.1.1 系统

- **初始状态：**
  - Q_i包含一个特殊的初始状态子集：每个$inbuf_i[l]$必须为空，但$outbuf_i[l]$未必为空。
- **转换函数(transition)：**

  处理器p_i的转换函数(实际上是一个局部程序)
  - **输入：** p_i可访问的状态
  - **输出：** 对每个信道l，至多产生一个msg输出
  - 转换函数使输入缓冲区$(1 \le k \le r)$清空。

§2.1.1 系统

- **配置：**配置是分布式系统在某点上整个算法的全局状态

  向量=(q_0, q_1,…q_{n-1}), q_i是p_i的一个状态

  一个配置里的outbuf变量的状态表示在通信信道上传输的信息，由del事件模拟传输

  一个初始的配置是向量=(q_0, q_1,…q_{n-1}), 其中每个q_i是p_i的初始状态，即每个处理器处于初始状态

§2.1.1 系统

- **事件：** 系统里所发生的事情均被模型化为事件，对于msg传递系统，有两种：

  comp(i)——计算事件。代表处理器p_i的一个计算步骤。其中，p_i的转换函数被用于当前可访问状态

  del(i,j,m)——传递事件，表示msg m从p_i传送到p_j

- **执行：** 系统在时间上的行为被模型化为一个执行。

  它是一个由配置和事件交错的序列。该序列须满足各种条件，主要分为两类：

§2.1.1 系统

①Safety条件：(安全性)

表示某个性质在每次执行中每个可到达的配置里都必须成立

在序列的每个有限前缀里必须成立的条件

例如：“在leader选举中，除了p_{max}外，没有哪个结点宣称自己是leader”

非形式地：安全性条件陈述了“尚未发生坏的情况” “坏事从不发生”

§2.1.1 系统

②liveness条件：(活跃性)

表示某个性质在每次执行中的某些可达配置里必须成立。

必须成立一定次数的条件(可能是无数次)

例如：条件：“p_1最终须终止”，要求p_1的终止至少发生一次；“leader选举，p_{max}最终宣布自己是leader”

非形式地，一个活跃条件陈述：“最终某个好的情况发生”

对特定系统，满足所有要求的安全性条件的序列称为一个执行;

若一个执行也满足所有要求的活跃性条件，则称为容许(合法的)(admissible)执行