

改变世界发展 历程的 40 年



Intel Developer
FORUM

创造美丽新世界



数字时代的变革

钱安达

英特尔研究
副总裁



变革仍在继续.....

提高生活质量 · 推动创新 · 扩大机会



学习环境的变革

19 世纪：
工业化



20 世纪：
信息时代



21 世纪：
互动时代



第二代学生电脑

完善的笔记本电脑平台：

- 改进的使用模式
- 全面的无线支持
- 集成的网络摄像头
- 结实耐用、可定制





提高生活质量



推动创新

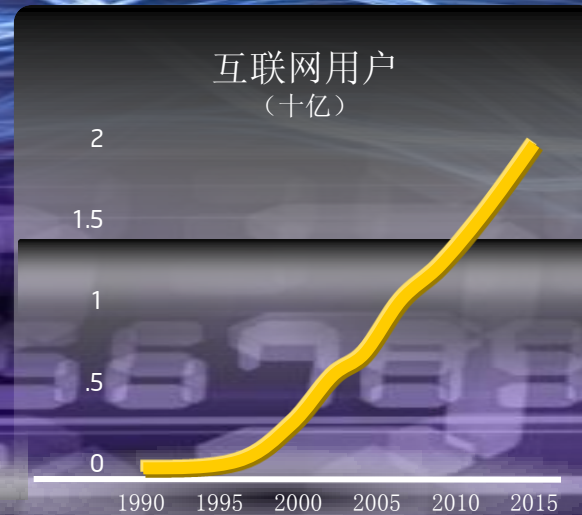
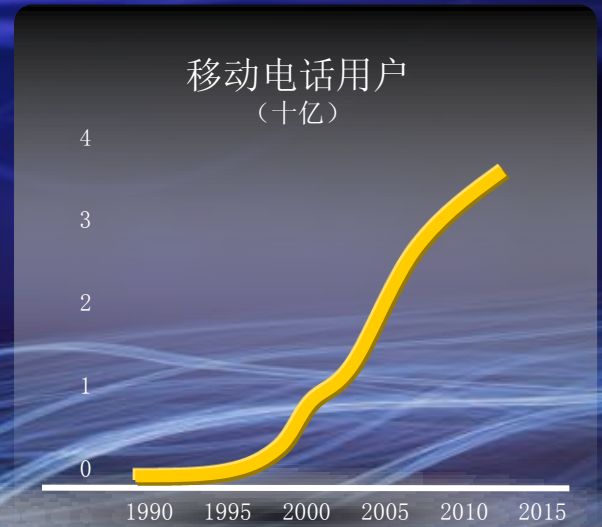
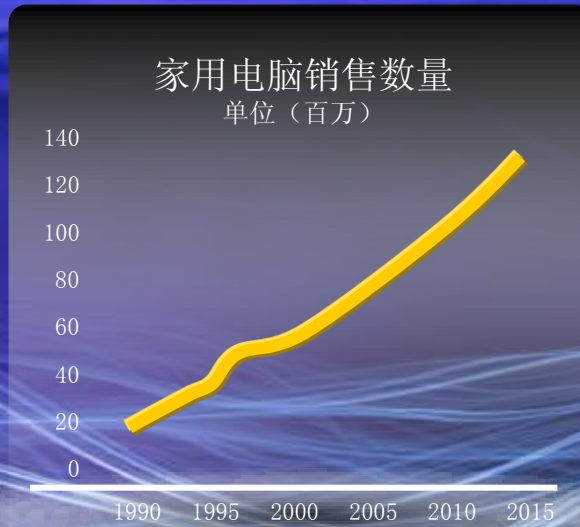
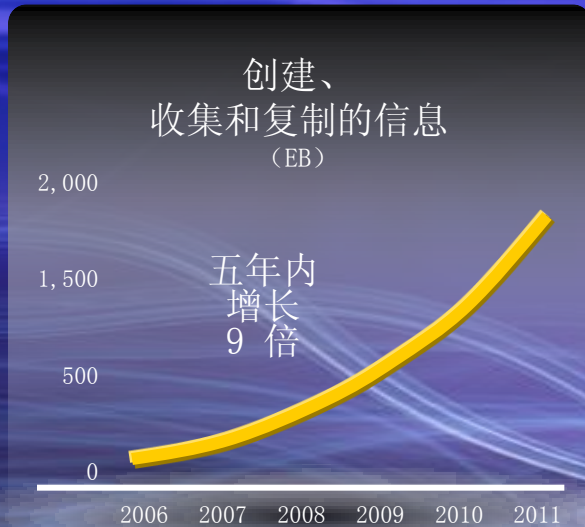


扩大机会

共享
内容
连通



数字技术在全球的发展



倪光南教授

中国工程院院士

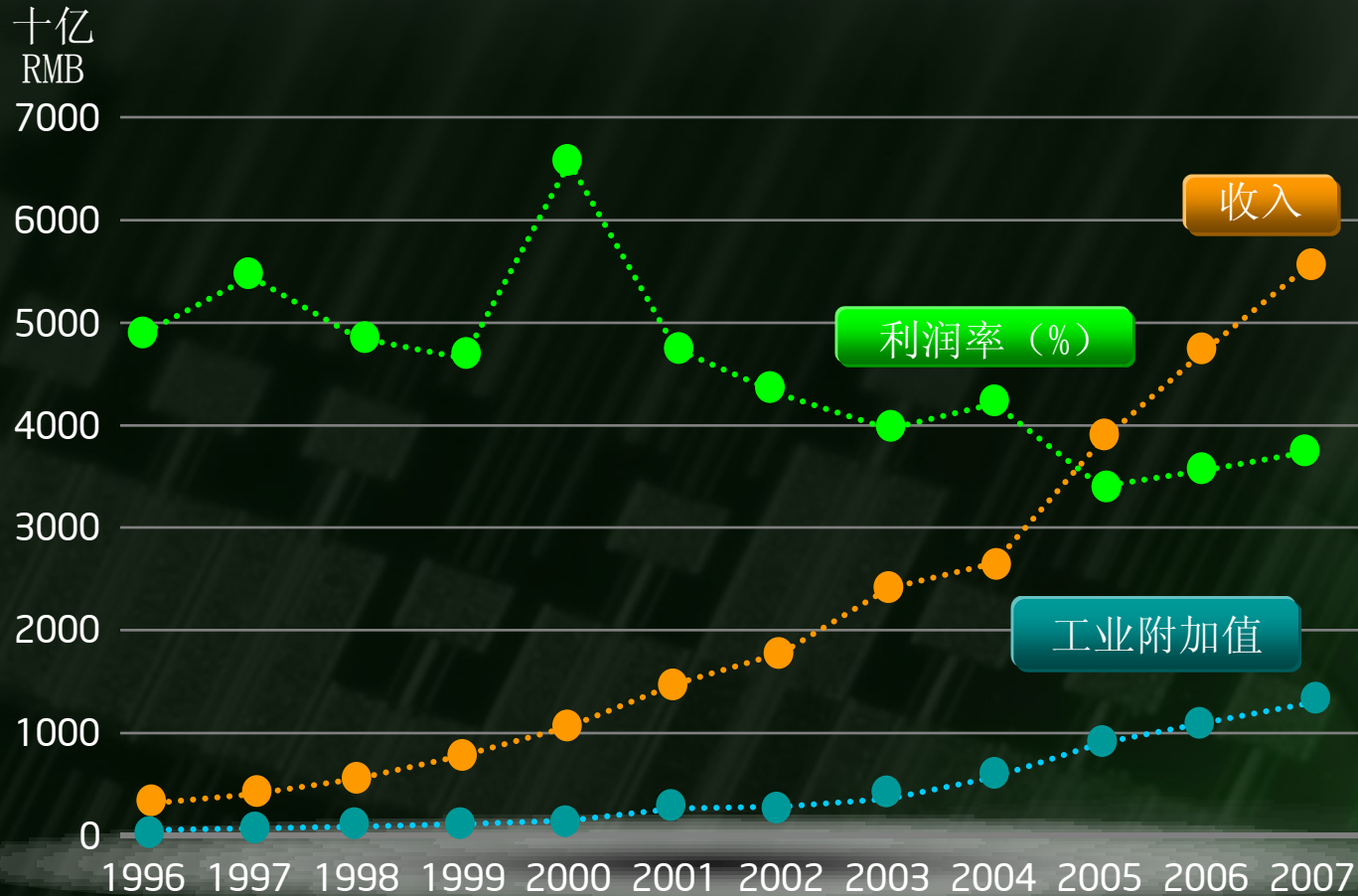
中国科学院计算技术研究所研究员



中国工程院
CHINESE
ACADEMY of
ENGINEERING

www.cae.cn

中国 IT 行业的发展



中国工程院
CHINESE
ACADEMY of
ENGINEERING

www.cae.cn

挑战

- 研发投入太少
- 处于低端制造环节，因而利润微薄
- 核心技术依赖进口



中国工程院
CHINESE
ACADEMY of
ENGINEERING

www.cae.cn

中国企业的对策

- 充分利用巨大的国内市场掌握核心技术并开发自己的 IP
- 推进开放标准
- 推广开放源代码软件



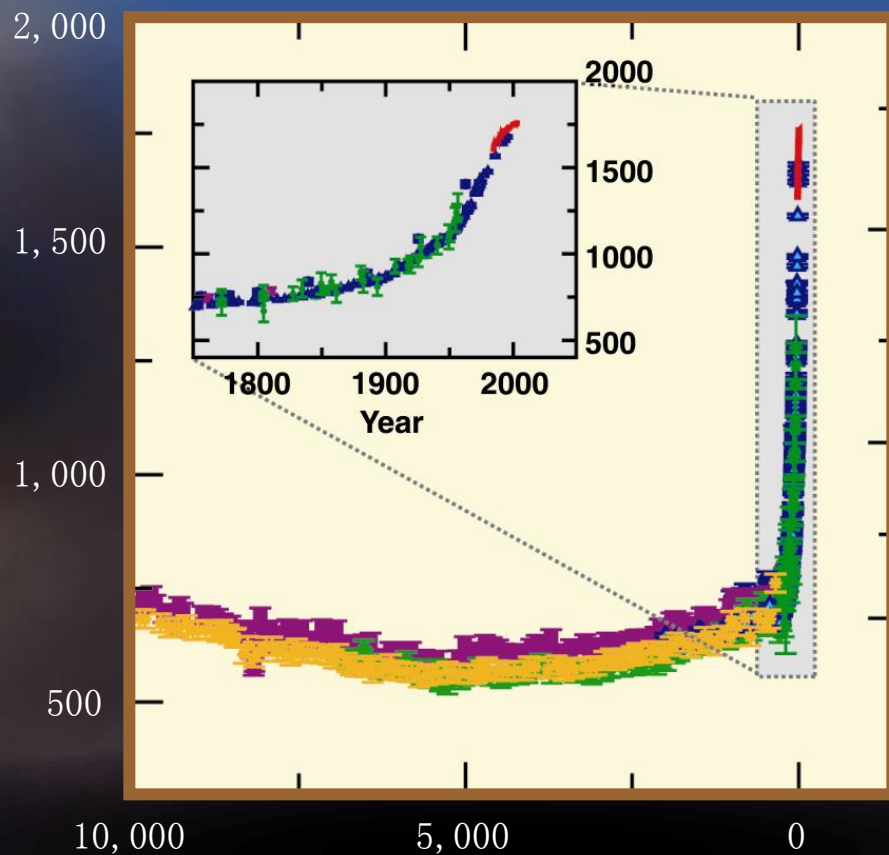
改善环境



温室效应

冰芯中温室气体含量的变化及现代数据

甲烷 (ppb)



辐射强度
(W/m²)

时间 (2005年前)

甲烷低成本光学检测

分子光谱学和气体感测领域的典型应用



首款级联拉曼硅激光器

全球对一次能源的需求

百万吨标准油

20,000

16,000

12,000

8,000

4,000

0

1971

2002

2010

2020

2030

石油

煤

天然气

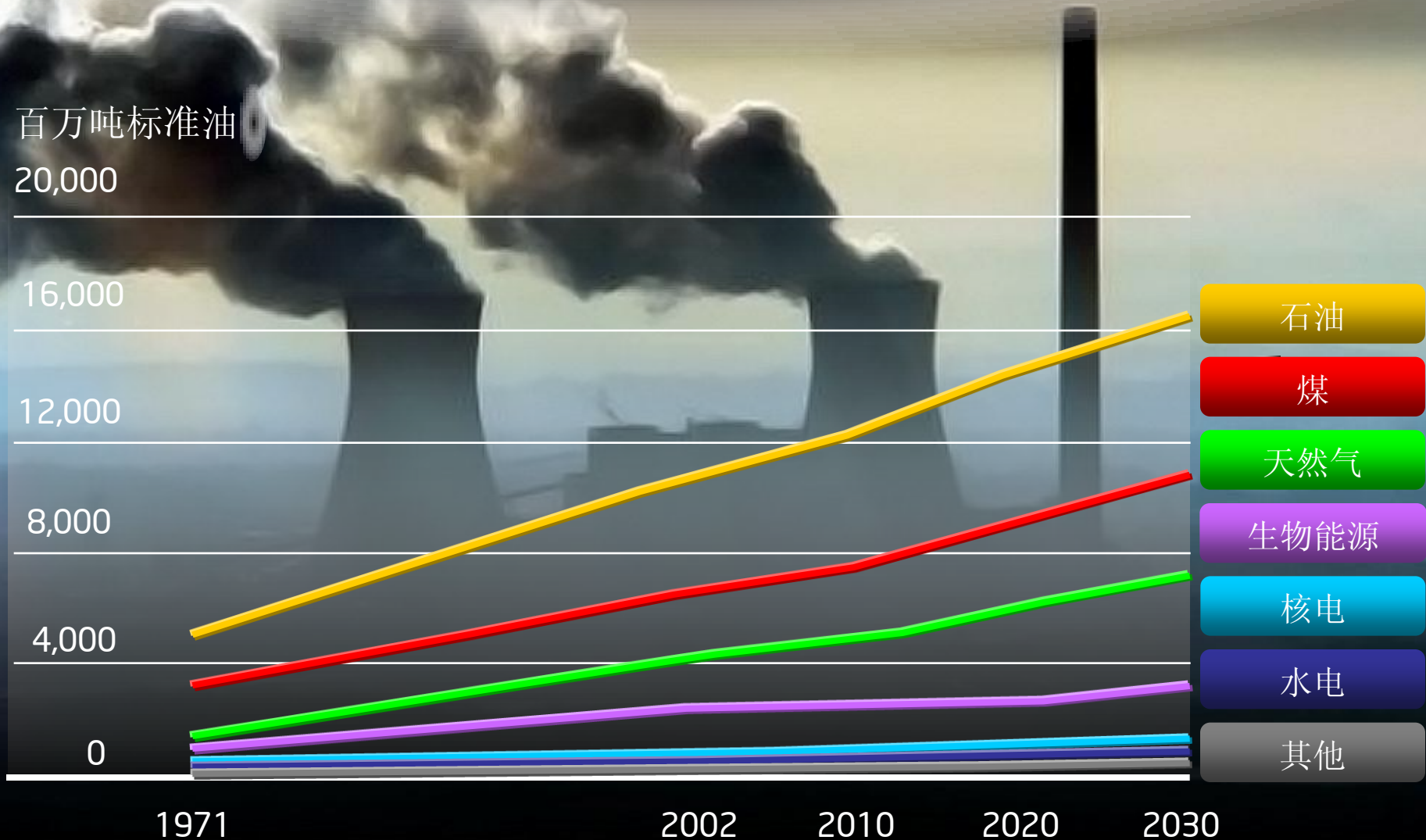
生物能源

核电

水电

其他

资料来源： IEA 2004

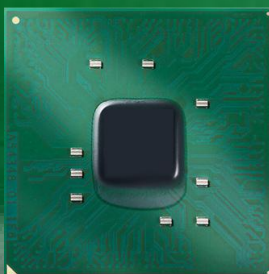


提高能效的整体性方法

平台电源管理



英特尔™
酷睿™ 2 双核

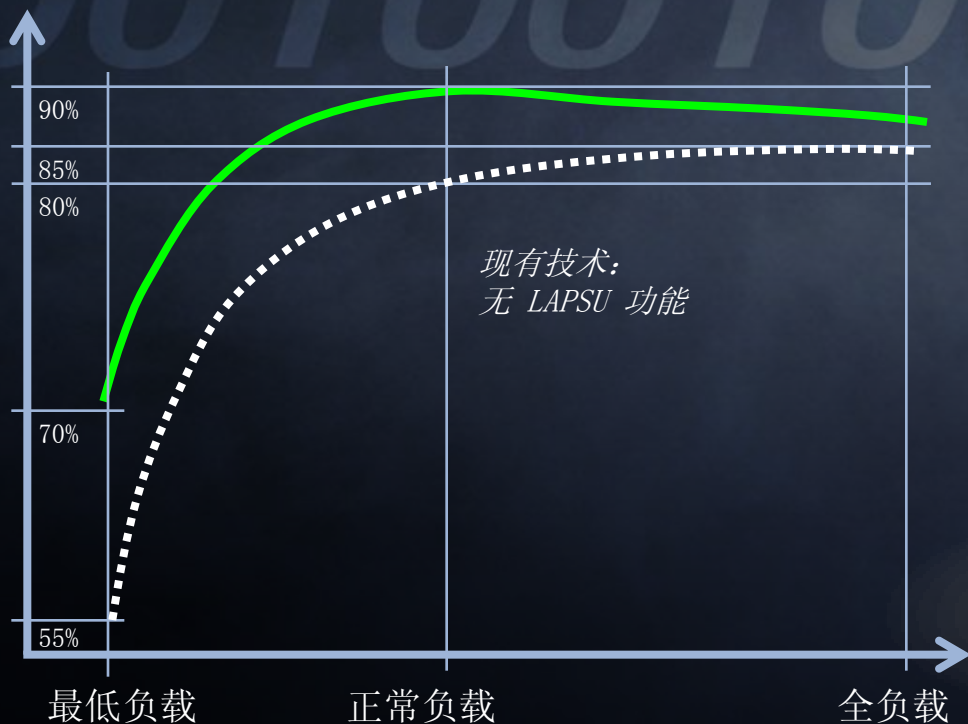


平台
电源管理

- 核心逻辑
- 操作系统和虚拟机监视器
- 可管理性
- 互连及外设
- 远程监测
- 供电和散热

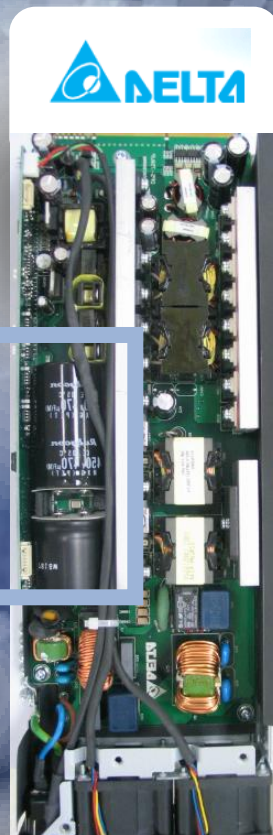
面向服务器的自适应电源的优势

具有 LAPSU 功能的高效 PSU:
低负载下电源效率提高15%，正常负载下电源效率提高10%



最低负载状态下
电源效率改进
达

70%+



其它优势包括:

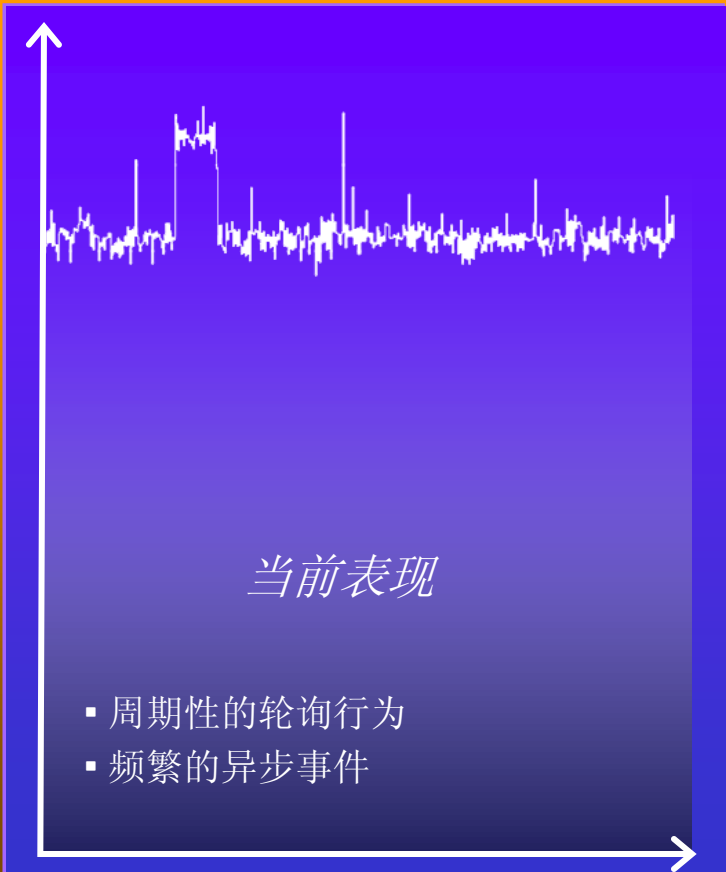
降低约40%的稳压电容(Bulk Capacitor) 或延长平台的维持时间

降低闲置功耗

常规闲置功耗：S0状态

闲置
平台
功耗

100%



时间

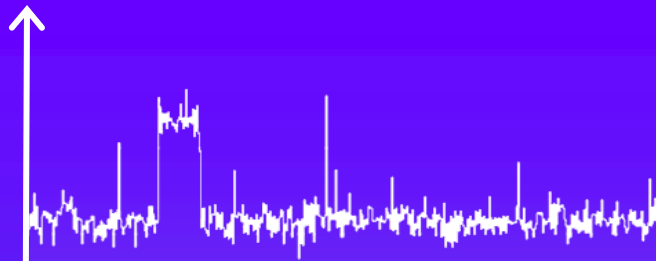
降低闲置功耗

常规闲置功耗：S0状态

改进后的闲置功耗：S0ix状态

闲置
平台
功耗

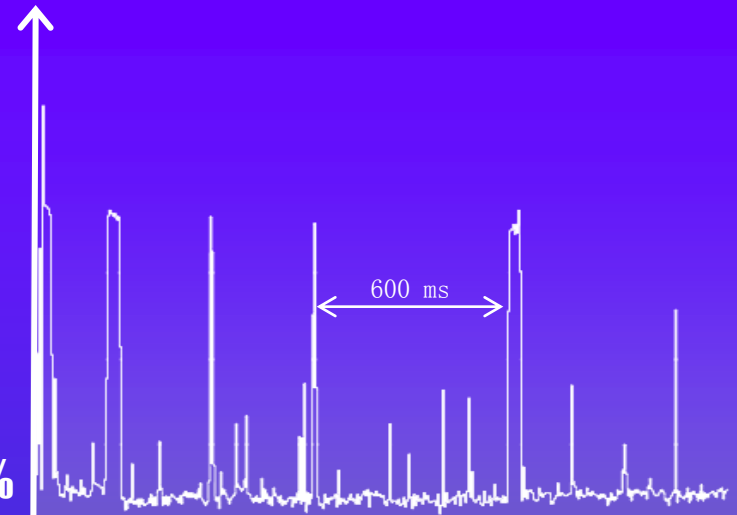
100%



当前表现

- 周期性的轮询行为
- 频繁的异步事件

50%



平台电源管理

- 无空循环 (Tickless) 的事件驱动型操作系统
- 协调一致的行为

时间

通过整个平台范围内的改动 最大限度地提高能效

闲置
系统
功耗
(瓦)



资料来源：轻薄型笔记本电脑闲置功耗降低幅度（注：功耗降低幅度为累计值）

通过整个平台范围内的改动 最大限度地提高能效

闲置
系统
功耗
(瓦)



资料来源：轻薄型笔记本电脑闲置功耗降低幅度（注：功耗降低幅度为累计值）

摄影术的变革





Ren Ng 博士
Refocus Imaging
公司
总裁兼首席执行官

● ● ● **refocus** *imaging*

光场（Light Field）摄影

传统相机

仅能拍摄二维图片
百万像素
采用物理方法处理光线
陈旧的物理约束
性能提高缓慢

光场相机

拍摄整个光场
百万光线
采用软件处理光线
全新、无约束的功能
性能提高迅速

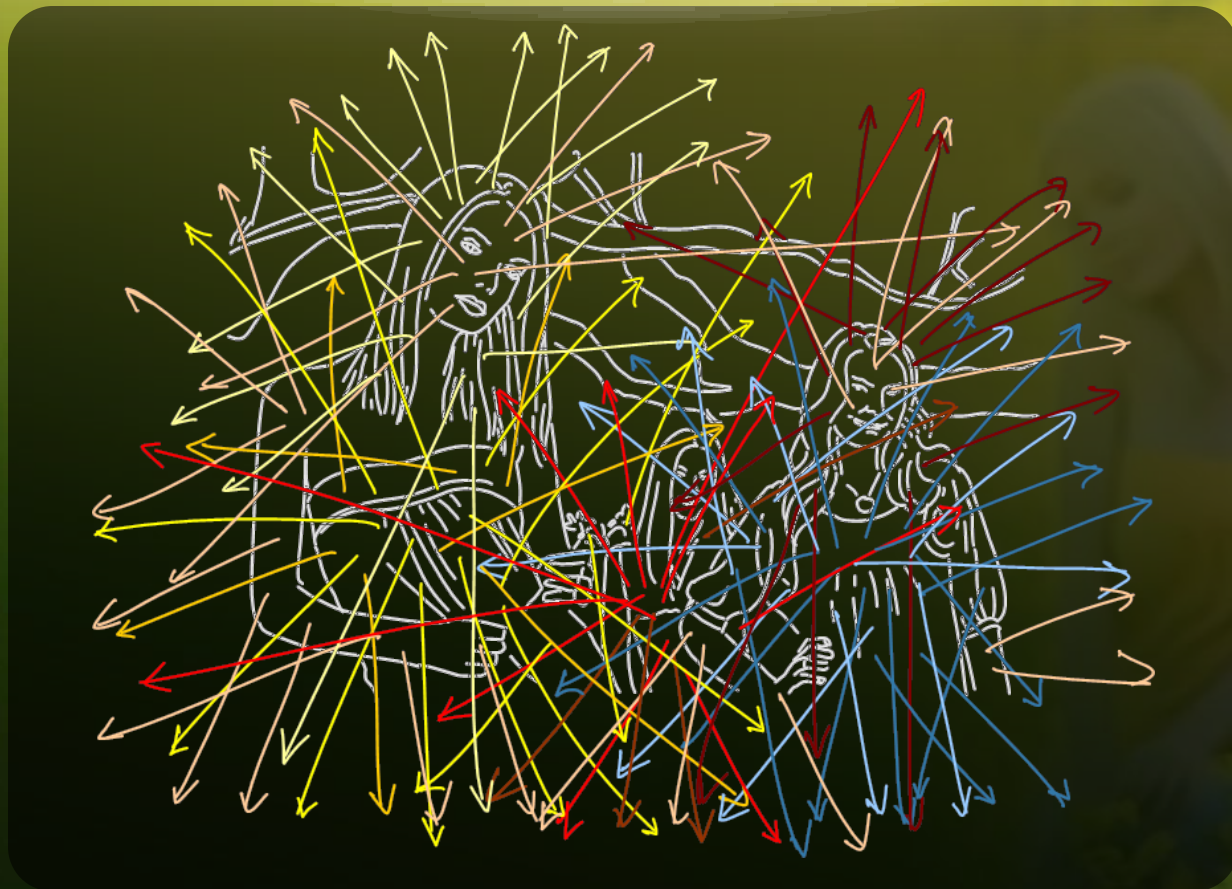
●●● **refocus**

●●● **refocus** imaging

光场



光场



光场： 所有光线
光场能够生成所有照片

光场应用

相机

软件

打印机

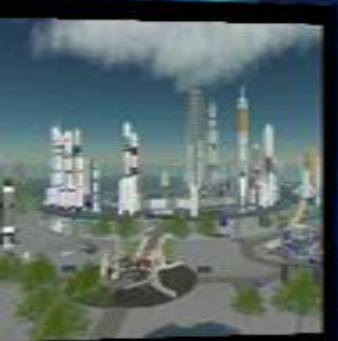
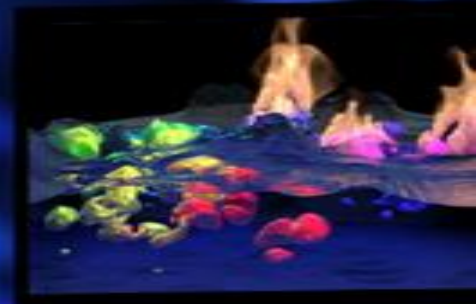
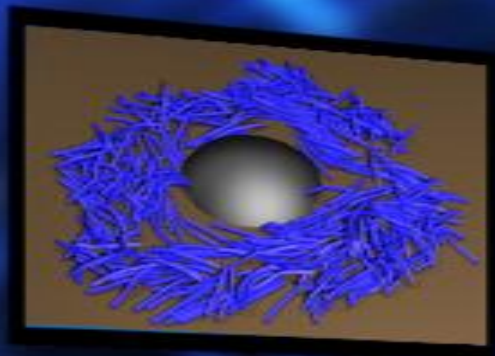
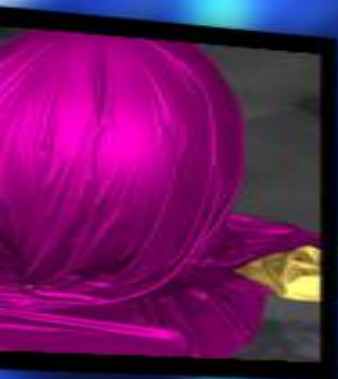


镜头

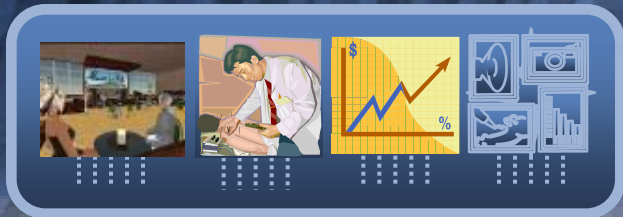
录像机

科研

视觉计算的变革



万亿级 (Tera-Scale) 计算： 现实与理想的差距



性能要求苛刻的应用

```
float s[N], x[N], r[N], v[N], t[N];  
float result[N];  
  
for(int i = 0; i < N; i++) {  
    float d1 = s[i] / ln(x[i]);  
    d1 += (r[i] + v[i] * v[i] * 0.5f) * t[i];  
    d1 /= sqrt(t[i]);  
    float d2 = d1 - sqrt(t[i]);  
  
    result[i] = x[i] * exp(r[i] * t[i]) *  
        (1.0f - CND(d2)) + (-s[i]) * (1.0f - CND(d1));  
}
```

挑战：传统软件侧重于单线程

万亿级平台：
通过并行处理扩充性能

并行编程难度极高



方之熙 (Jesse Fang)

总经理

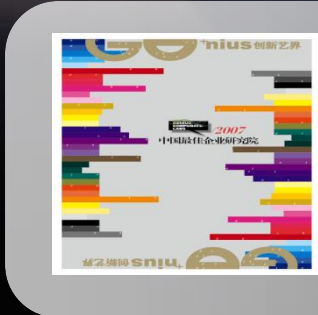
英特尔中国研究中心



联席主席

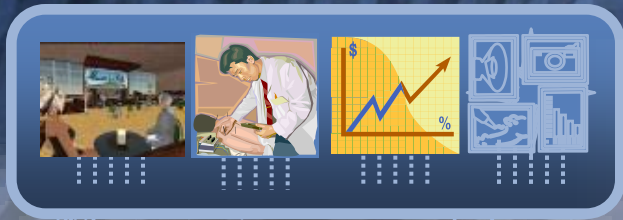
35届 ISCA 会议 (北京, 6月21-25日)

10-year
1998 - 2008
Intel China Research Center
英特尔中国研究中心10周年



“2007年中国最佳研发中心”
《环球企业家》杂志

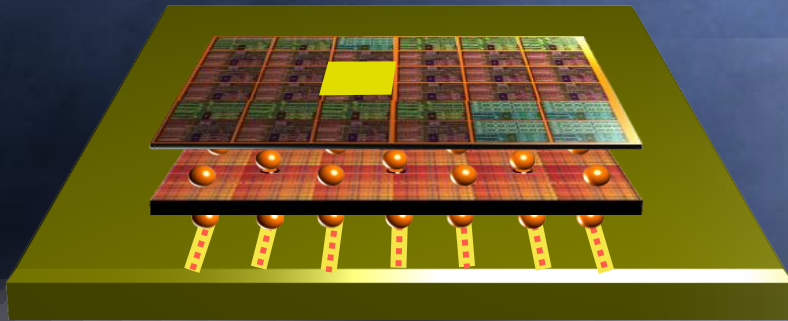
为什么并行编程难度如此高？



非常规模式与数据结构



当前扩充至多核 → 难
今后扩展至更多内核 → 更难

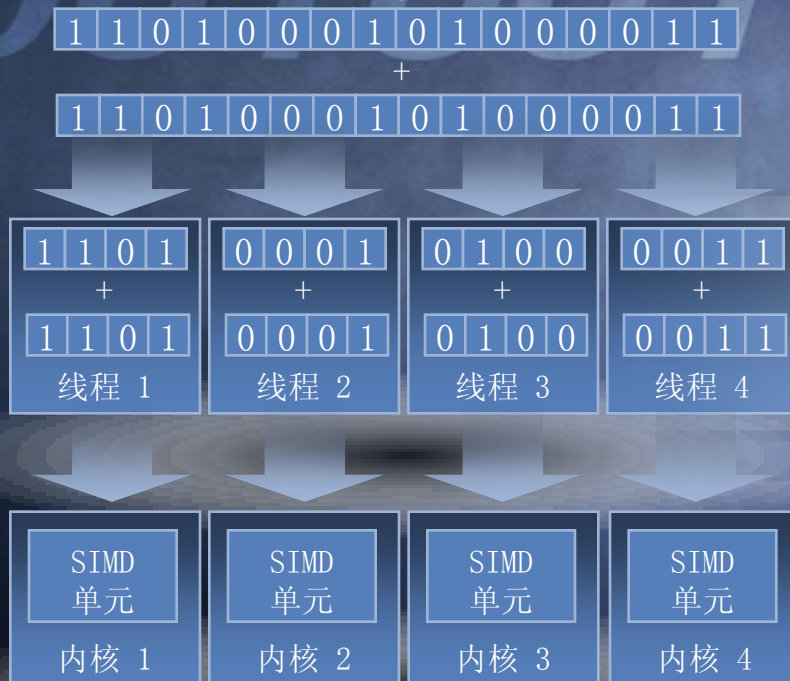


增加内核 (2 → 超过 64 个内核)
向量指令 (4 → 超过 8 位宽)
高性能互连

Ct: 吞吐率编程模型

```
TVEC<F32> a(src1), b(src2);  
TVEC<F32> c = a + b;  
c.copyOut(dest);
```

用户编写
内核独立 C++ 代码



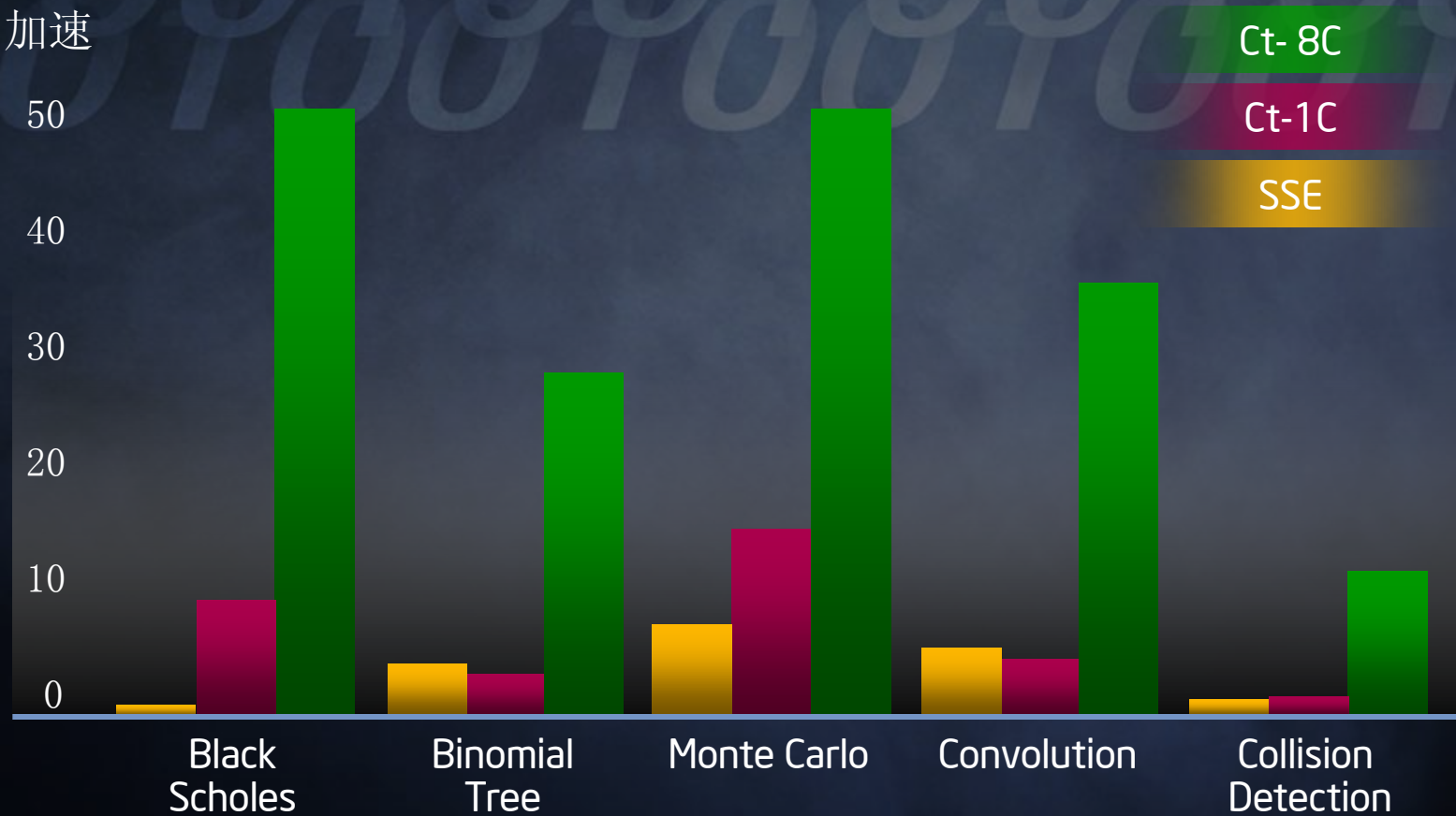
Ct 并行运行时:
自动扩展至增加内核

Ct JIT 编译器:
自动向量化、SSE、AVX、
Larrabee

程序员串行思维; Ct 采用并行处理

卓越的纵向扩充

加速





张霞博士
东软有限公司首席技术官
兼首席知识官

Neusoft[®]

演示代码示例： C vs. Ct vs. SSE



36 行
单线程

```
//C version of SVM
void ClrMain::calcPotValue(int k, int *x, int *y, int
*pdDecisionValue)
{
    int i = 0;
    int Zk = 0, qExp = 0;
    int iTemp = 0, x1 = 0, x2 = 0, xx = 0, tab1 = 0, tab2 =
0, tab3 = 0;

    for (i = 0; i < DIMENSION; i++, x++, y++) {
        iTemp = *x-*y;
        Zk += (iTemp * iTemp);
    }

    float lExp = exp(-1.0*GAMMA*Zk/65025);
    *pdDecisionValue = (int)((lExp * coef[k] * 8);
}

int ClrMain::SVM(int *x)
{
    int k = 0;
    int Decision_value = 0;
    int aDecisionValue[TOTAL_SV+SV_EXPEND];

    for (k = 0; k < TOTAL_SV+SV_EXPEND; k++) {
        calcPotValue(k, x, y + k * DIMENSION,
&aDecisionValue[k]);
    }

    for (k = 0; k < TOTAL_SV; k++) {
        Decision_value += aDecisionValue[k];
    }

    Decision_value -= (int)(RHO * 8 * ICOEF);
    Decision_value = Decision_value * 10000 / ICOEF / 8;

    return Decision_value;
}
```

C

29 行
向量化/多线程
可向前扩展

```
#!/ Ct version of SVM
int ClrMain::ctSVM(int *x)
{
    int    Decision_value;

    __TC__
    //! Copy in data
    TVEC<I32> tCoef(coef, ( TOTAL_SV+SV_EXPEND ));
    TVEC<I32> tx(x, DIMENSION);
    TVEC2D<I32> t2dY(y, ( TOTAL_SV + SV_EXPEND ), DIMENSION);

    //! compute Z
    TVEC2D<I32> t2dX = repeatRow(tx, ( TOTAL_SV + SV_EXPEND
));

    TVEC2D<I32> t2dZ = ( t2dX - t2dY );
    TVEC<F32> tz = addReduce( t2dZ * t2dZ );
    tz = exp( (TVEC<F32>)tz * ( (__F32)(-GAMMA) / 65025 ));

    //! compute decision value
    TVEC<I32> tDecVal = (TVEC<I32>)( tz * (TVEC<F32>tCoef *
(F32)8 );

    I32 decVal = addReduce( tDecVal)[0];
    decVal -= (I32)( RHO * 8 * ICOEF);
    decVal *= (I32)( 10000 / ICOF / 8 );

    Decision_value = decVal.toValue();
}__TC__

return Decision_value;
}
```

Ct

116 行
向量化/单线程

```
//SSE version of SVM
void ClrMain::sseValueSSE(int k, int *x, int *y, int
*pdDecisionValue)
{
    int i = 0;
    int *pX, *pY, *pZ;
    int *pZ;

    static __m128i float 2dY[4] = { 0, 0, 0, 0 };
    static __m128i float constValue[4] =
{ 1.0*GAMMA*G025, 1.0*GAMMA*G025,
1.0*GAMMA*G025, 1.0*GAMMA*G025 };

    __m128i tCoef, t2dY, t2dX, t2dZ, t2dZx, t2dZy, t2dZz;
    __m128i tx, tx1, tx2, tx3;
    __m128i tZ;
    __m128i tDecVal;
    __m128i tDecVal0;

    for (i = 0; i < DIMENSION; i++) {
        __m128i tCoef = _mm_cvtsi32_si128(i, a2D0);
        *pdDecisionValue += (int)((tCoef * 8);
    }

    for (k = 0; k < TOTAL_SV+SV_EXPEND; k++) {
        calcPotValue(k, x, y + k * DIMENSION,
&aDecisionValue[k]);
    }

    for (k = 0; k < TOTAL_SV; k++) {
        Decision_value += aDecisionValue[k];
    }

    Decision_value -= (int)(RHO * 8 * ICOEF);
    Decision_value = Decision_value * 10000 / ICOEF / 8;

    return Decision_value;
}

int ClrMain::SSE(int *x)
{
    int k = 0;
    int Decision_value = 0;
    float sum;

    __m128i tCoef, t2dY, t2dX, t2dZ, t2dZx, t2dZy, t2dZz;
    __m128i tx, tx1, tx2, tx3;
    __m128i tZ;
    __m128i tDecVal;
    __m128i tDecVal0;

    for (i = 0; i < DIMENSION; i++) {
        __m128i tCoef = _mm_cvtsi32_si128(i, a2D0);
        *pdDecisionValue += (int)((tCoef * 8);
    }

    for (k = 0; k < TOTAL_SV+SV_EXPEND; k++) {
        calcPotValue(k, x, y + k * DIMENSION,
&aDecisionValue[k]);
    }

    for (k = 0; k < TOTAL_SV; k++) {
        Decision_value += aDecisionValue[k];
    }

    Decision_value -= (int)(RHO * 8 * ICOEF);
    Decision_value = Decision_value * 10000 / ICOEF / 8;

    return Decision_value;
}

__m128i tCoef, t2dY, t2dX, t2dZ, t2dZx, t2dZy, t2dZz;
__m128i tx, tx1, tx2, tx3;
__m128i tZ;
__m128i tDecVal;
__m128i tDecVal0;

for (i = 0; i < DIMENSION; i++) {
    __m128i tCoef = _mm_cvtsi32_si128(i, a2D0);
    *pdDecisionValue += (int)((tCoef * 8);
}

for (k = 0; k < TOTAL_SV+SV_EXPEND; k++) {
    calcPotValue(k, x, y + k * DIMENSION,
&aDecisionValue[k]);
}

for (k = 0; k < TOTAL_SV; k++) {
    Decision_value += aDecisionValue[k];
}

Decision_value -= (int)(RHO * 8 * ICOEF);
Decision_value = Decision_value * 10000 / ICOEF / 8;

return Decision_value;
}
```

SSE

机器人的变革

无人驾驶汽车



2006年
“DARPA Grand Challenge”
比赛获胜车



2007年
“DARPA Urban Challenge”
比赛获胜车

池明旻 博士

复旦大学
计算机科学与工程系



復旦大學

日月光華 旦復旦兮

中文版



復旦大學

日月光華旦復旦今

中文版

有关 Fuwa 的研究



- 计算认知
- 自主心智发育学习
- 多模人机交互

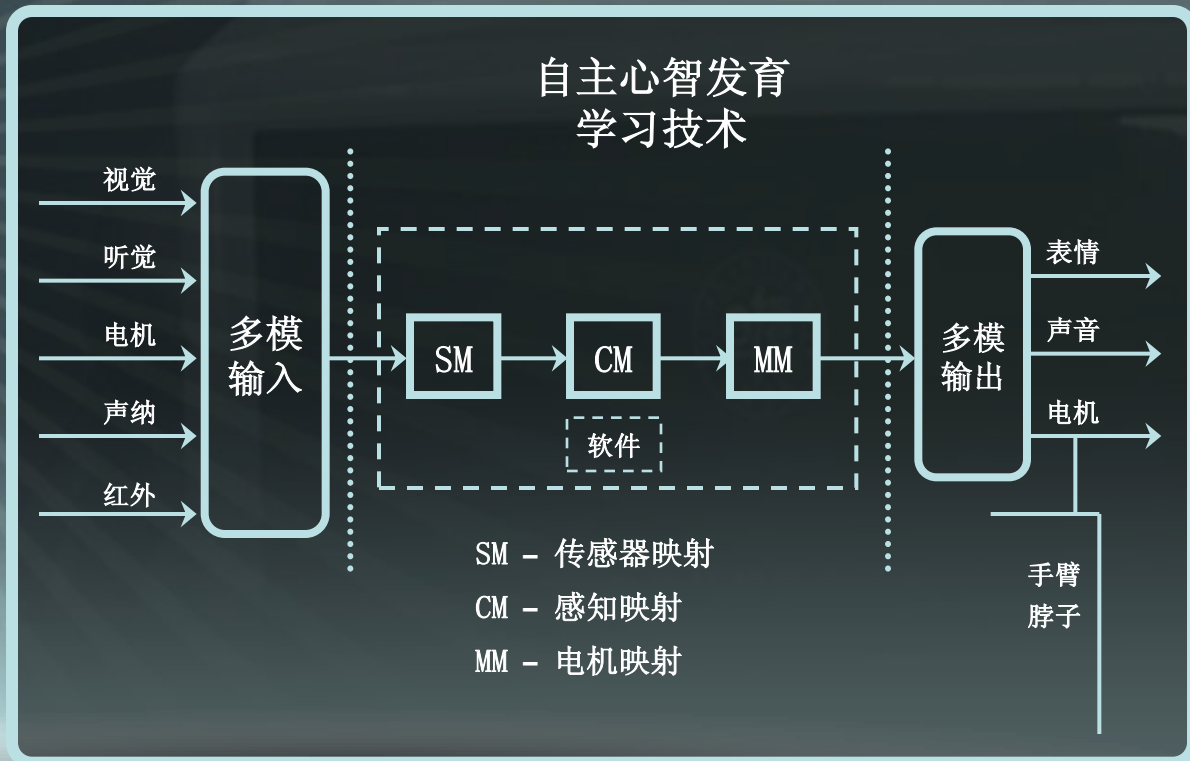


復旦大學

日月光華旦復旦今

中文版

系统架构





復旦大學

日月光華旦復旦今

中文版



未来远景

Fuwa 有许多潜在应用:

- 教育平台
- 娱乐控制台

英特尔 CPU 就是 Fuwa 的大脑

- 更强大的计算能力
- 更高级的智能

变革仍在继续.....

提高生活质量 · 推动创新 · 扩大机会



工欲善其事
必先利其器



改变世界发展 历程的 40 年



Intel Developer
FORUM

创造美丽新世界