

# *Perl, un linguaggio aperto*

## *Perikronos*

Silvano Catinella  
<catinella@yahoo.com>

Perl Workshop  
Pisa, 19 Luglio 2004



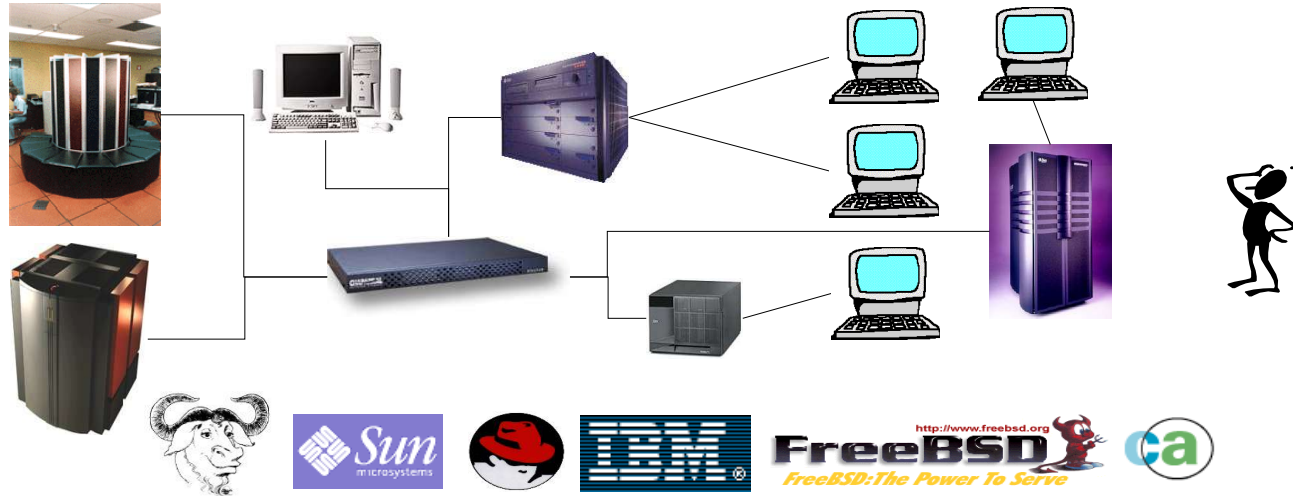
PeriKronos

# *Uno dei principali vantaggi dei software OS: lo sviluppo autonomo*

- Gli ambienti enterprise
- L'esigenza di un prodotto flessibile
- La funzione giusta al momento giusto



# Enterprise environments



- Risorse HS/SW eterogenee
- Il bisogno cronico di strumenti personalizzati
- Difficoltra di gestione: pacchetti monolitici contro strumenti modellabili.
- Gestione centralizzata

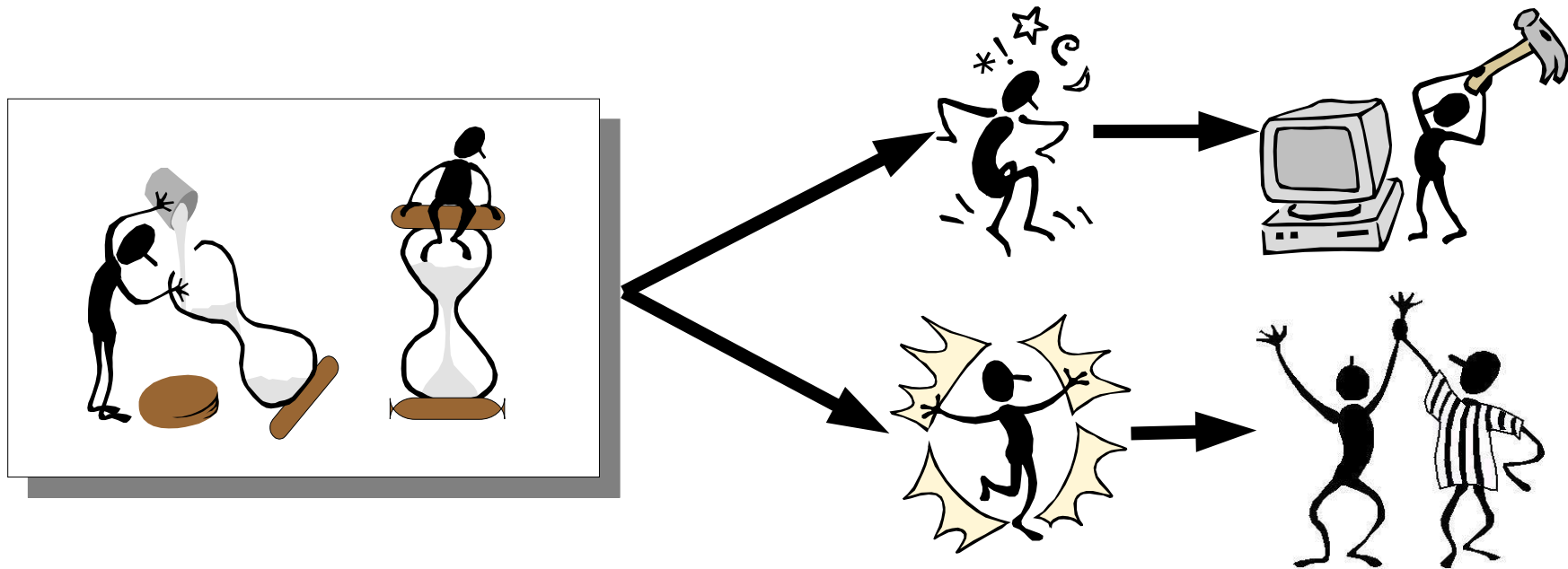


# L'esigenza di un ambiente flessibile

- Competizione
- Innovazione
- Cambiamento



# La soluzione giusta al momento giusto



Rispettare le deadline può essere vitale.

Chi vi può garantire la soluzione giusta al momento giusto?

Come avere una reale visibilità sullo stato dei vs. progetti?

Come avere un maggior controllo?

La soluzione italiana (outsourcing) è la migliore?

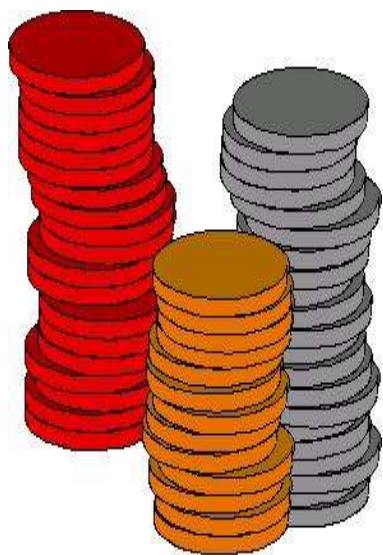


# *Il modello di collaborazione OS*

- Il software come strumento per il business
- La caduta dei limiti geografici: il CVS



# Il Software come strumento per il business



## Avere o essere

Produrre profitto in virtù di un prodotto.... o delle proprie capacità?

## Consulenza ed educational il business Open Source

La libertà di scelta e la schiavitù della paura  
“Nessuno è mai stato licenziato per aver scelto xxx”



# La caduta dei limiti geografici: il CVS



Concurrent Version System

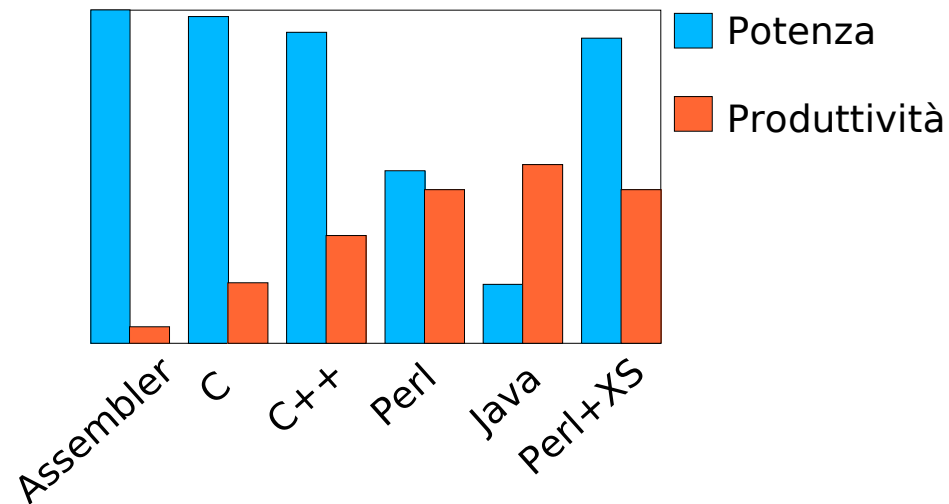


# *La filosofia del Perl: la Libertà*

- Possibilità di estendere le funzionalità del Perl (XS)
- Contratto di fiducia con l'utilizzatore: l'incapsulamento
- Libertà vs. Affidabilità: comportamento responsabile



# La possibilità di estendere le funzionalità del Perl (XS)



Due caratteristiche a confronto: potenza e produttività

potenza ---> potere ---> cosa posso fare?  
produttività ----> quanto tempo mi richiede?

Interfacciare Perl e C



# La possibilità di estendere le funzionalità del Perl (XS)

Changes

Example.pm

**Example.xs**

Makefile.PL

MANIFEST

README

test.pl

modulo perl

**procedure in C/XS**

programma perl per creare il Makefile

contiene la lista dei files del progetto

contiene dei test

-----C-----

```
#include <stdio.h>
```

```
int tre (int x) {
```

```
    x *= 3;
```

```
    return (x);
```

```
}
```

-----XS-----

```
int tre (x)
```

```
    int x
```

```
    CODE:
```

```
        RETVAL=3*x;
```

```
    OUTPUT:
```

```
        RETVAL
```



# Libertà vs. Affidabilità comportamento responsabile

- Libertà di espressione.

Il Perl permette di sviluppare la stessa funzione in molti modi logicamente e sintatticamente diversi.

```
<command> or die "<error message>";  
die "<error message>" unless <command>;  
if (not <command>) { die "<error message>"}
```

```
if ($a =~ /<pattern>/) {<command>}  
elsif ($a =~ /<pattern>/) {<command>}
```

```
....  
$_=$a;  
SWITCH: {  
    if (/<pattern>/) {<command>; last}  
    if (/<pattern>/) {<command>; last}  
}
```

- Libertà di scrivere pessimo codice

La mancanza di restrizioni permette con facilità di scrivere codice non strutturato, caotico e illeggibile.

Autoregolamentazione.





Perikronos

<http://perikronos.sourceforge.net>

---

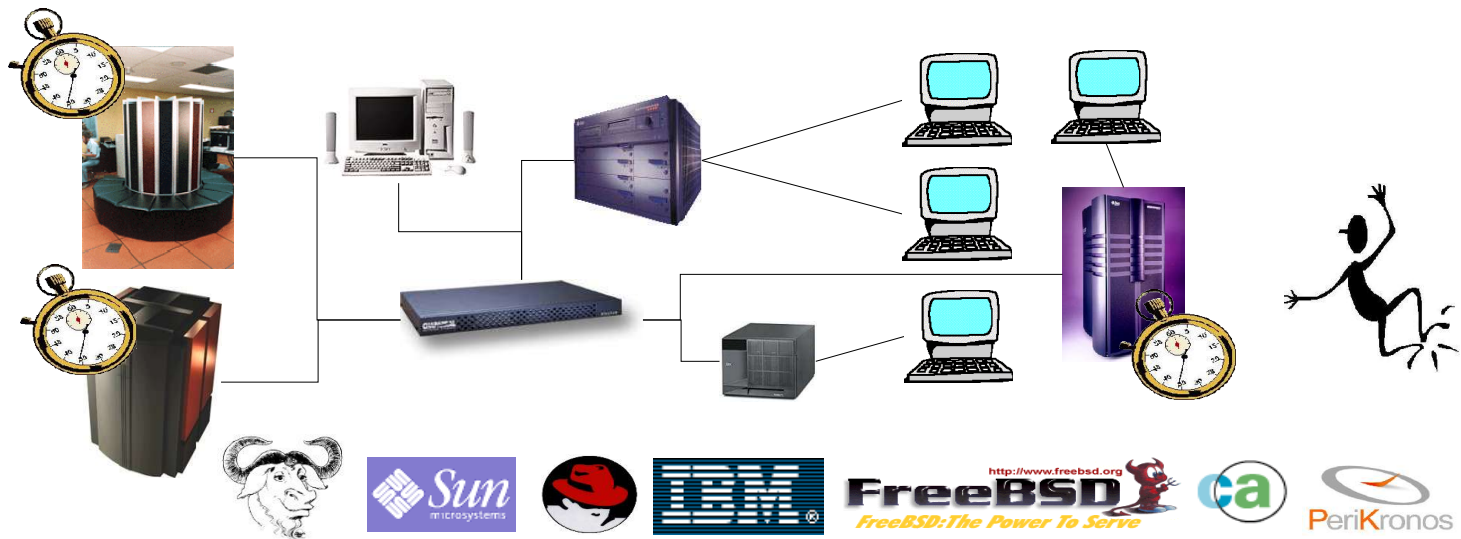
---

# *Un esempio pratico: il progetto Perikronos*

- Cos'è PK?
- Perché è nato?
- Perché il Perl?
- Il Perl in ambiti non convenzionali



# Cos'è Perikronos?



Enterprise Multyplatform Scheduler



PeriKronos

# Perchè è nato questo progetto?

Perikronos è nato dalla critica ai prodotti proprietari che si occupano di schedulazione in ambito enterprise.

La sfida di Perikronos:

sviluppare un software di dimensioni contenute (1.4 Mbyte) avente le stesse funzioni dei suoi mastodontici omologhi commerciali (asvariate centinaia di megabyte)



# Perchè è stato scelto il Perl per questo progetto?

Perikronos è un framework

Integrazione con il sistema  
controllo dei job

Linguaggio multiplatform  
ambienti eterogenei

Potenza dove serve!  
Possibilità di scrivere codice a basso-livello



# Il Perl in ambiti non convenzionali

- **l'idea sbagliata:**  
“...il Perl è adatto solo a sviluppare piccoli script di sistema o CGI”
- **Ambiti non convenzionali:**  
software che richiedono la potenza di C per alcune funzioni periferiche, e una elevata produttività (OOP) per poterle gestire
- **I limiti di Perl:**
  - complessi progetti ove sia richiesta la possibilità di isolarne le componenti
  - applicazioni con richieste estreme di performance complessive

