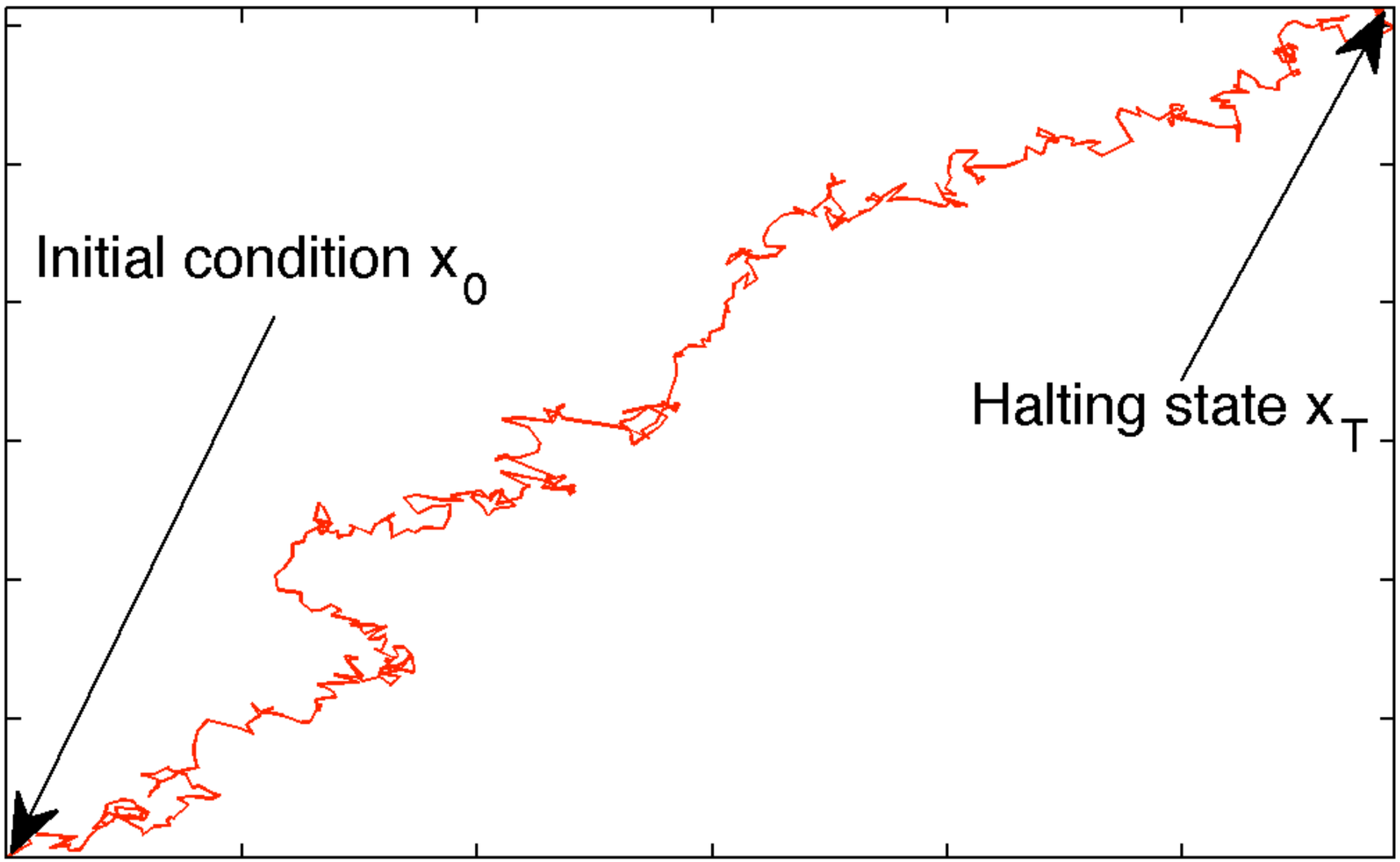
Parallelization by Simulated Tunneling

Amos Waterland,[1] Jonathan Appavoo,[2] Elaine Angelino,[1] and Margo Seltzer[1]

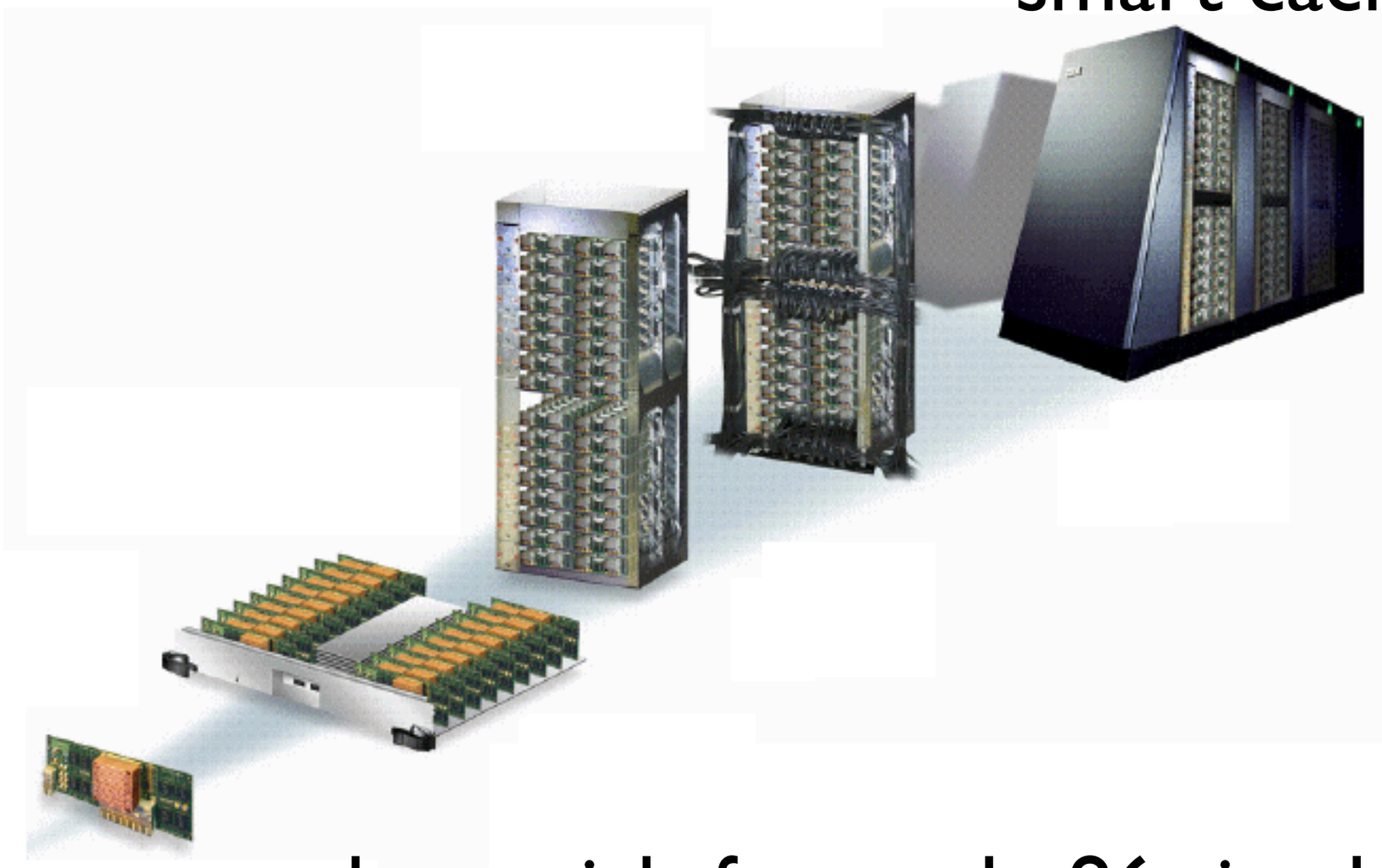
[1] Harvard University, 33 Oxford Street, Cambridge, MA 02138, USA

[2] Boston University, 111 Cummington Street, Boston MA 02215, USA

June 7, 2012



Vast majority of transistors used as a massively parallel "smart cache"



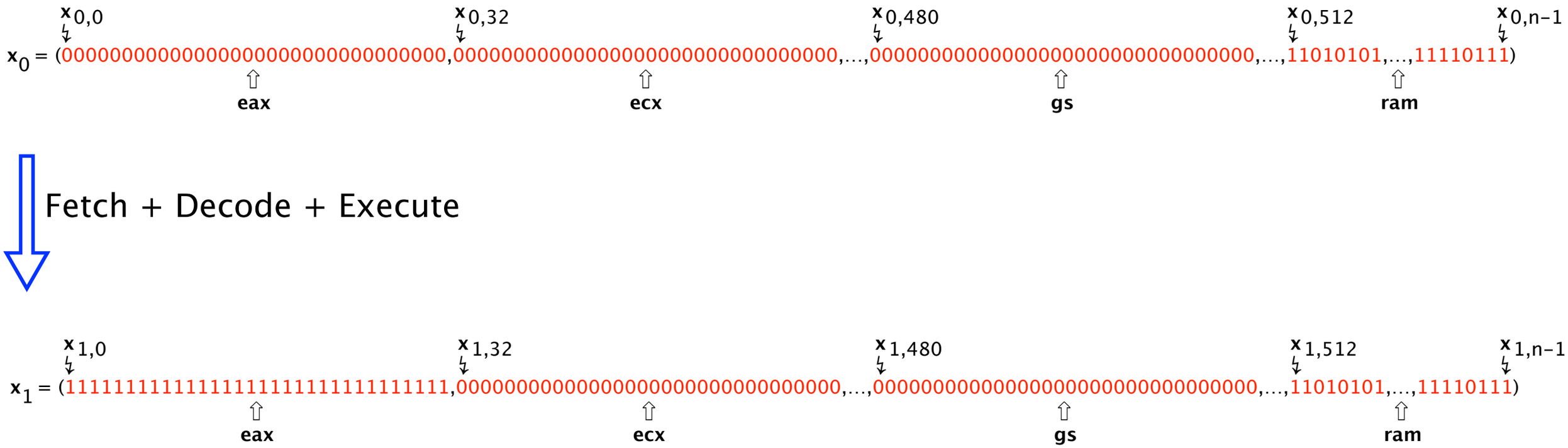
Pick one master node: straightforward x86 simulator

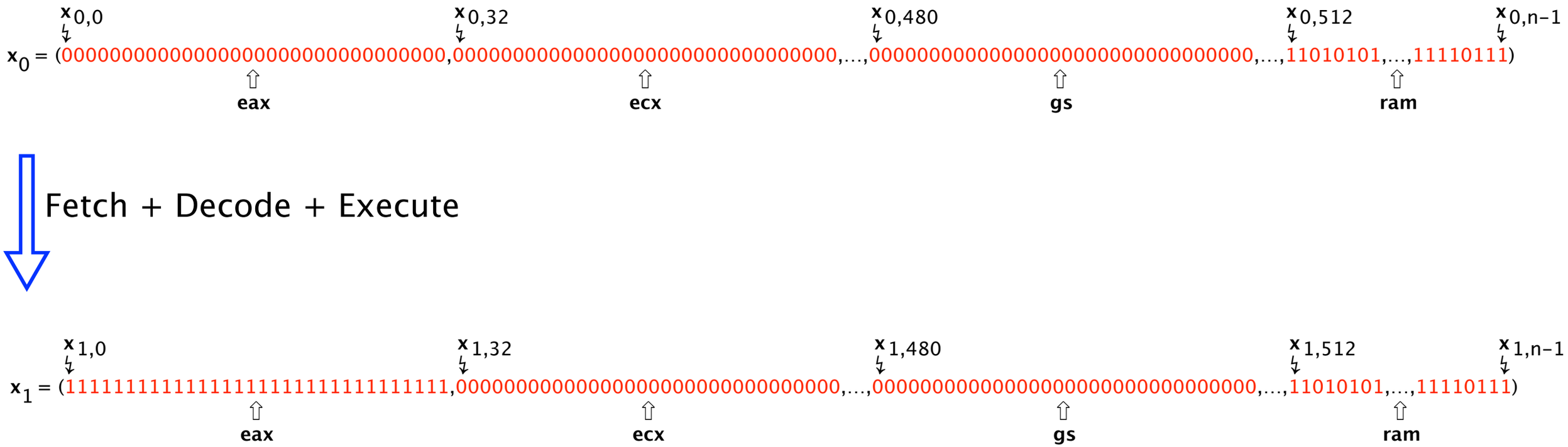
Machine model: master node queries cache, "tunnels" to new state if match

Deterministic programs: input allowed, but up front

$$
\begin{aligned}
\mathit{state} &= \mathit{registers} \times \mathit{instructions} \times \mathit{heap} \times \mathit{stack} \\
&= \mathit{registers} \times \mathit{memory} \\
&= (00000100010101111101010 \ldots 11111111010110101010101)
\end{aligned}
$$


x0,0
x0,32
x0,480
x0,512
x0,n−1
x0 = (00000000000000000000000000000000,00000000000000000000000000000000,...,00000000000000000000000000000000,...,11010101,...,11110111)
eax
ecx
gs
ram


x0,0
x0,32
x0,480
x0,512
x0,n−1
x0 = (00000000000000000000000000000000,00000000000000000000000000000000,...,00000000000000000000000000000000,...,11010101,...,11110111)
eax
ecx
gs
ram
Fetch + Decode + Execute
x1,0
x1,32
x1,480
x1,512
x1,n−1
x1 = (11111111111111111111111111111111,00000000000000000000000000000000,...,00000000000000000000000000000000,...,11010101,...,11110111)
eax
ecx
gs
ram



A dynamical system is a state space $\mathcal{X}$ and an evolution rule f.

$$x \in \mathcal{X}$$

$$f : \mathcal{X} \to \mathcal{X}$$

Initial condition x_0

Halting state x_T

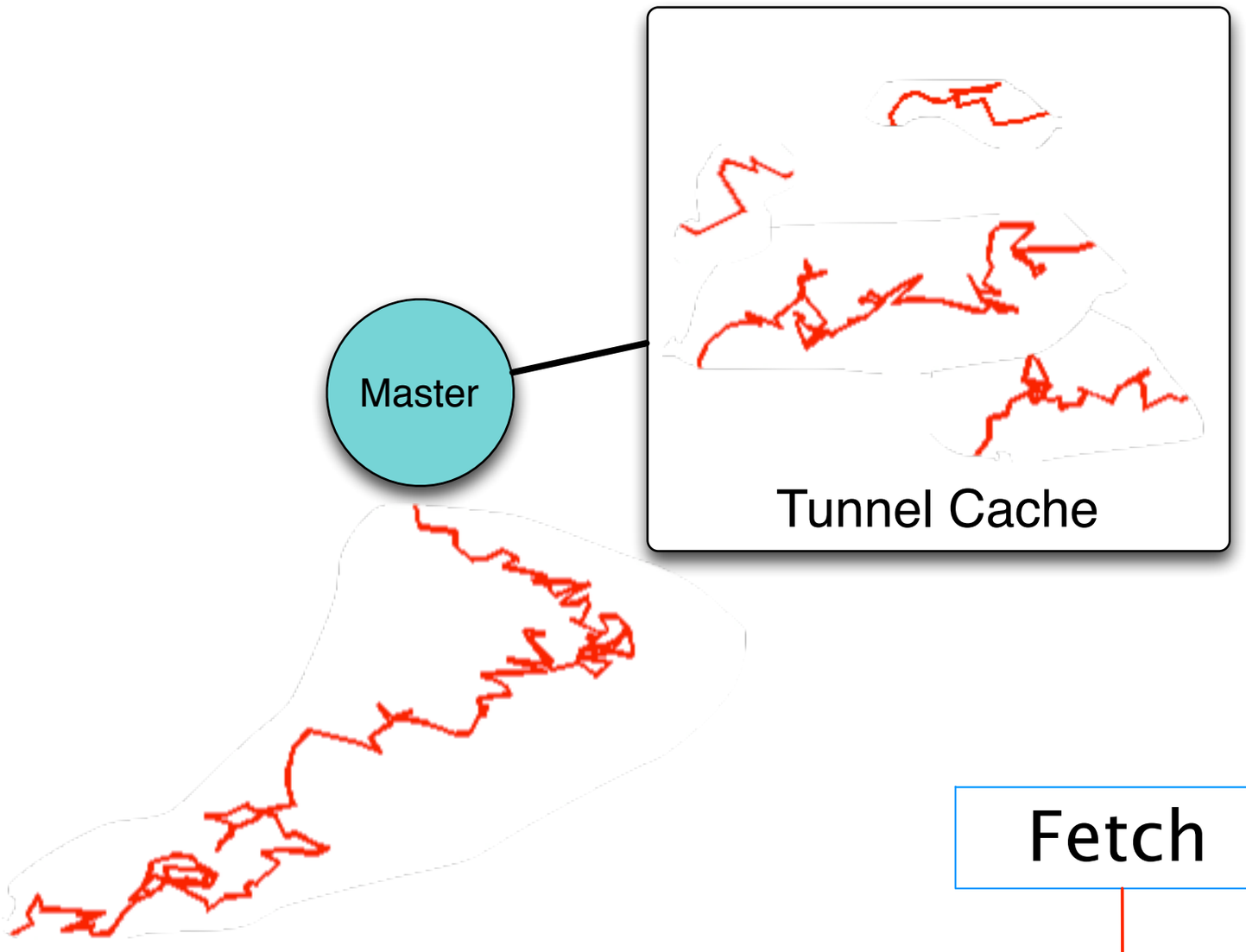
Fetch

Decode

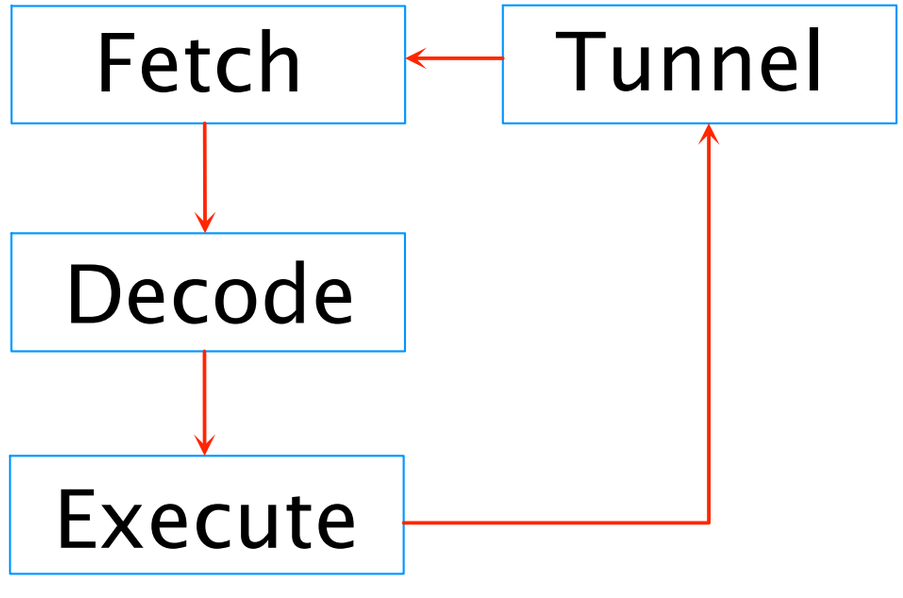
Execute

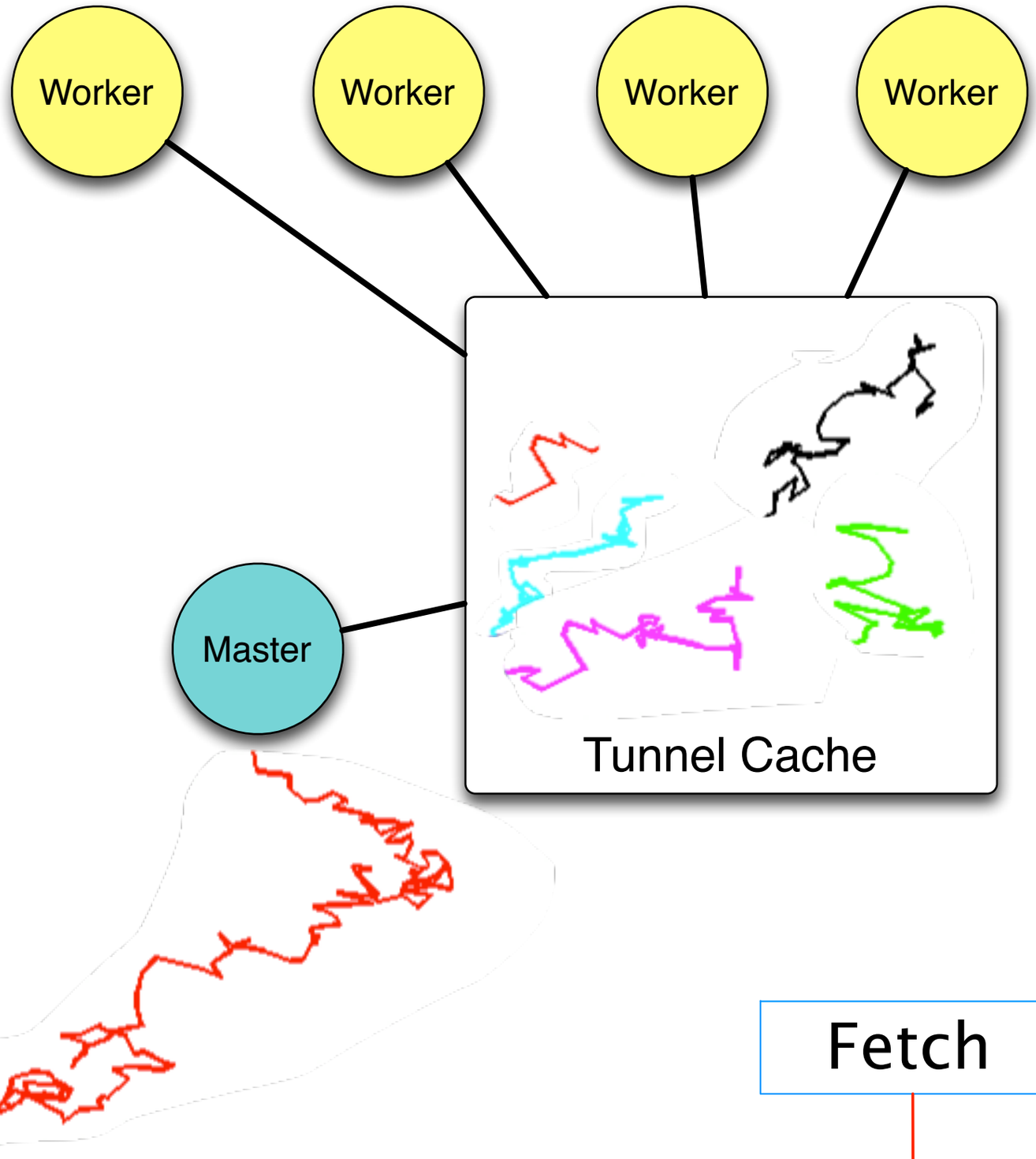
$f : \mathcal{X} \to \mathcal{X}$

If initial point of tunnel matches master's current point (up to symmetries) -- tunnel immediately to final point



The crucial role of symmetries
E.g. translation on non-causal bits

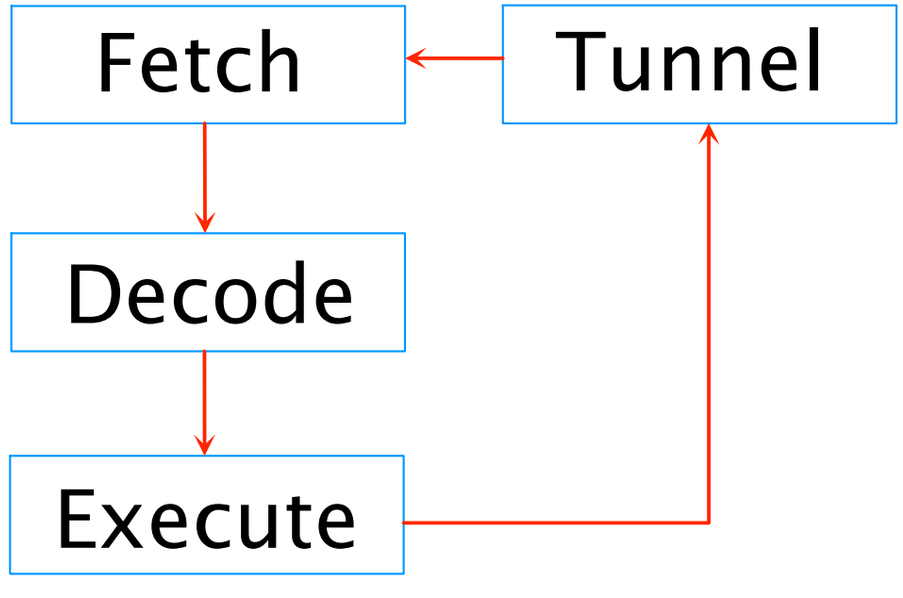


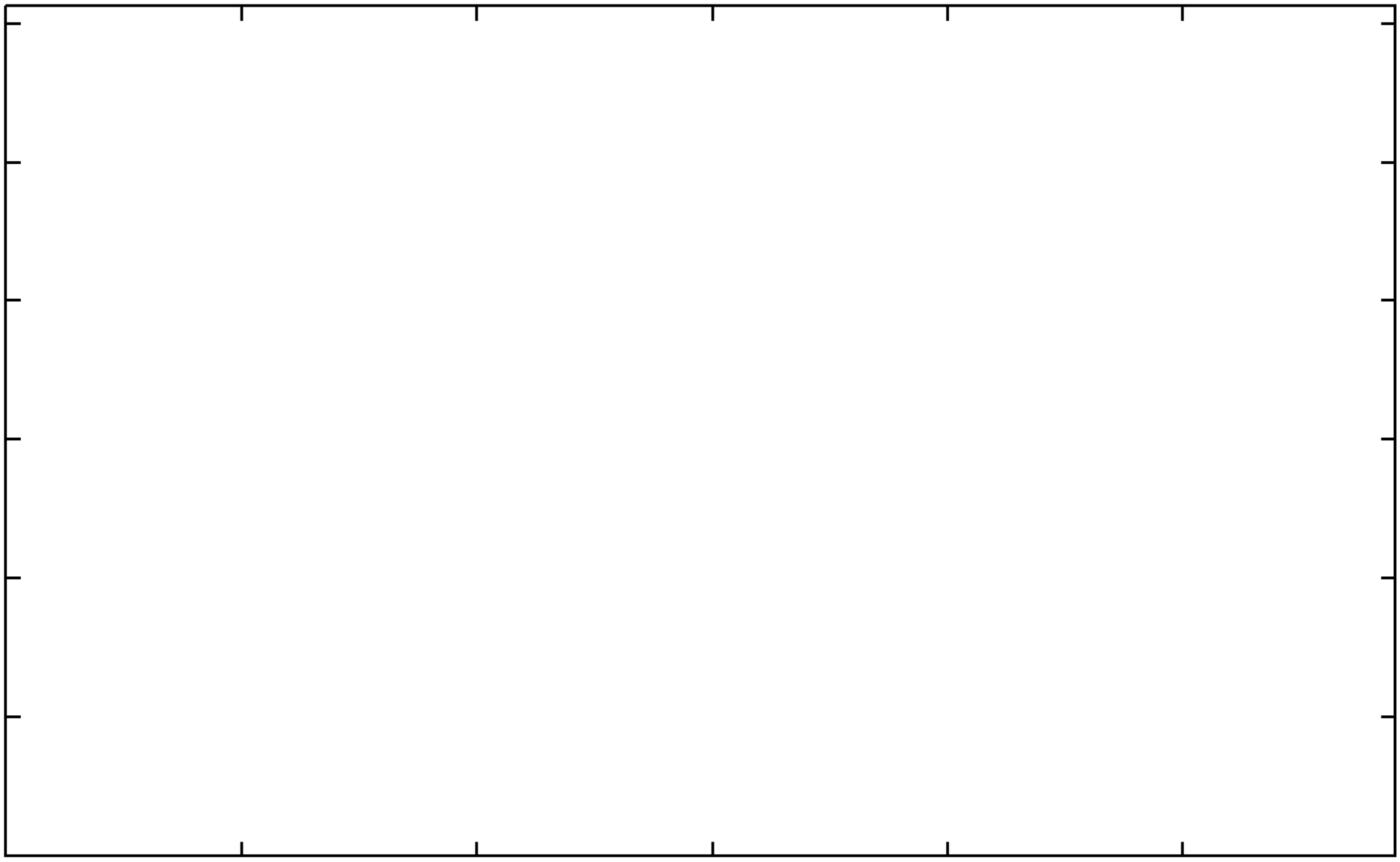


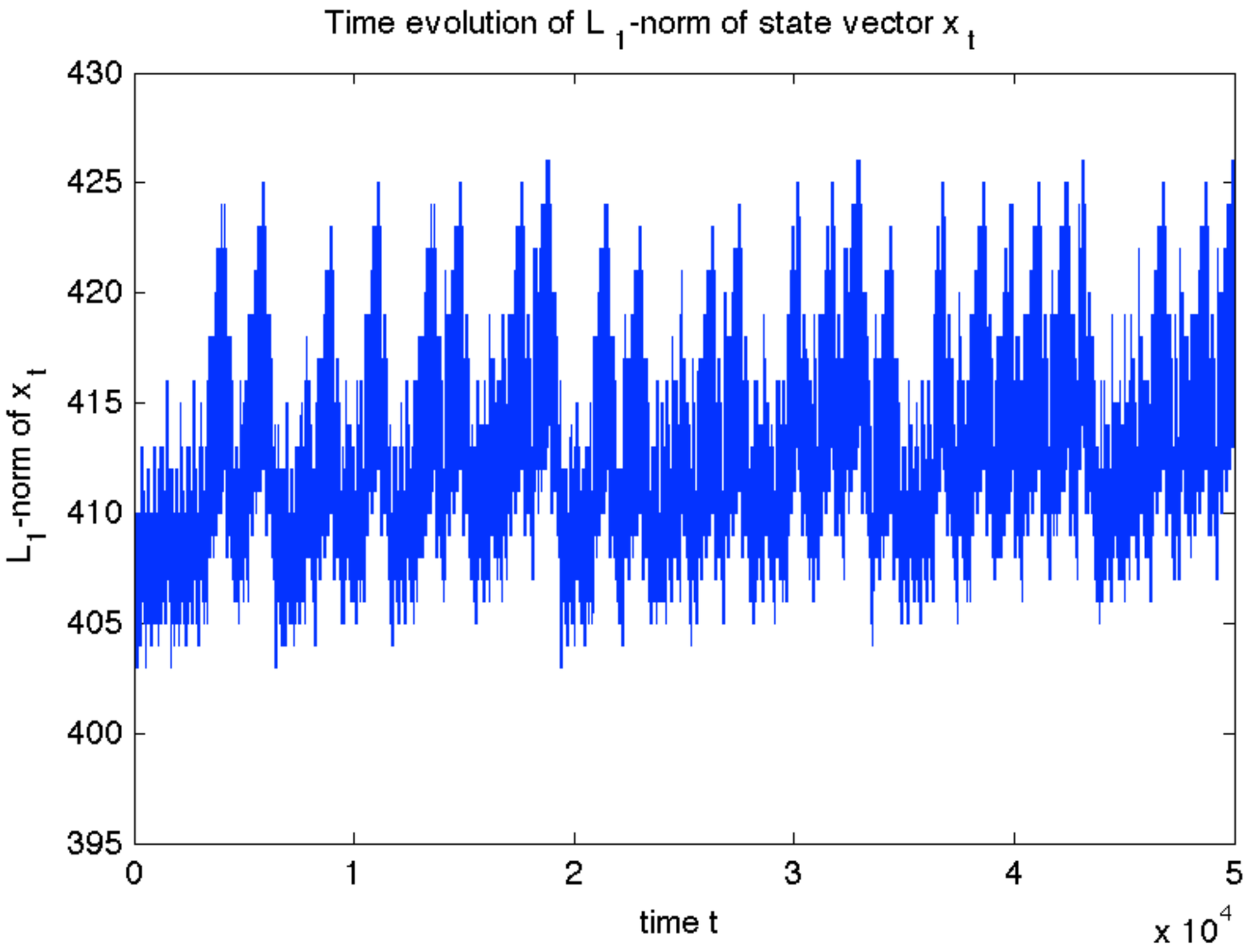
Symmetries again

E.g. only have to get 10% bits right

E.g. function memoization






Time evolution of L 1-norm of state vector x t
L1-norm of x t
430
425
420
415
410
405
400
395
0
1
2
3
4
5
time t
x 10^4


IBM
Blue Gene/P

Log-log plot of run time scaling

Our scaling
Perfect linear scaling
Bare metal

Run time in seconds

10^{6}
10^{5}
10^{4}
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}

10^{0} 10^{1} 10^{2} 10^{3} 10^{4} 10^{5}

Number of 850 MHz Blue Gene cores

Status

- Deterministic computation (for now)
  - All I/O up front
- Experimental infrastructure
  - Restricted *x*86 simulator
  - MPI on laptop / Blue Gene
  - Search programs
  - Simple Bayesian predictor

