CAS CS 210 : COMPUTER SYSTEMS

Professor: Appavoo

- Objective: To provide the student with a fundamental understanding of how to computer systems work and are programmed.
- We will discuss the following in the first lecture:
  - What is a computer system?
  - Course Info, Piazza, Syllabus
  - Discussion and Quizzes
  - General advice.
  - If there is time a little about who I am and what I do

http://www.youtube.com/watch?v=yEP-pLvSdXs

6:51

http://www.youtube.com/watch?v=YhsKCnDD3F8

10:06

http://www.youtube.com/watch?v=CoxQLJkLq1c

First 1:30

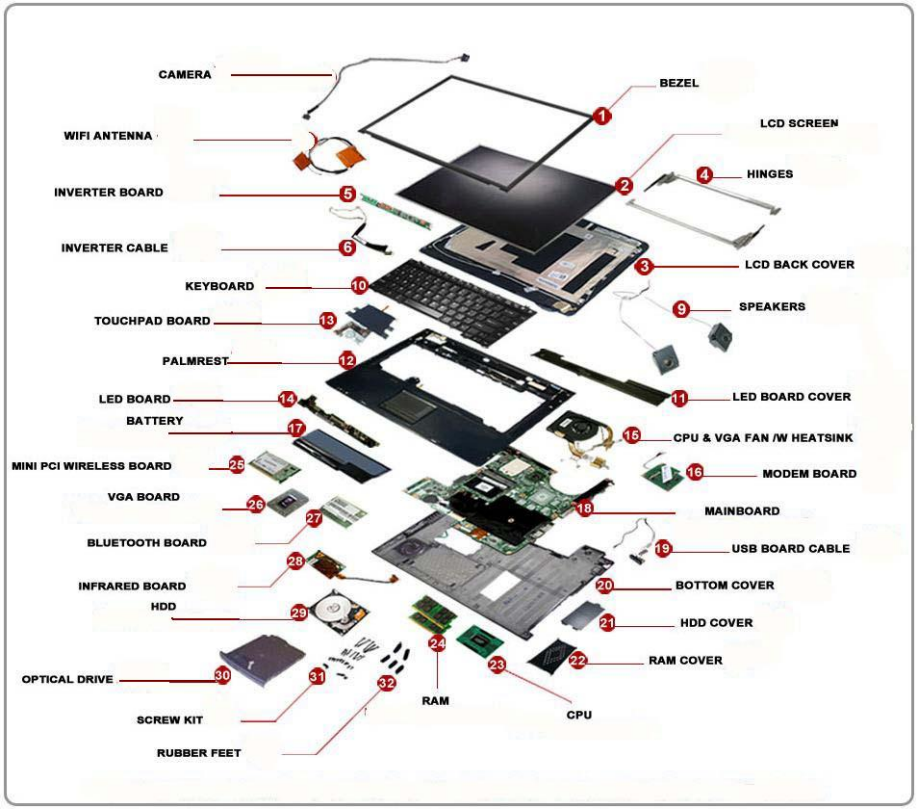
SO WHAT IS A COMPUTER?

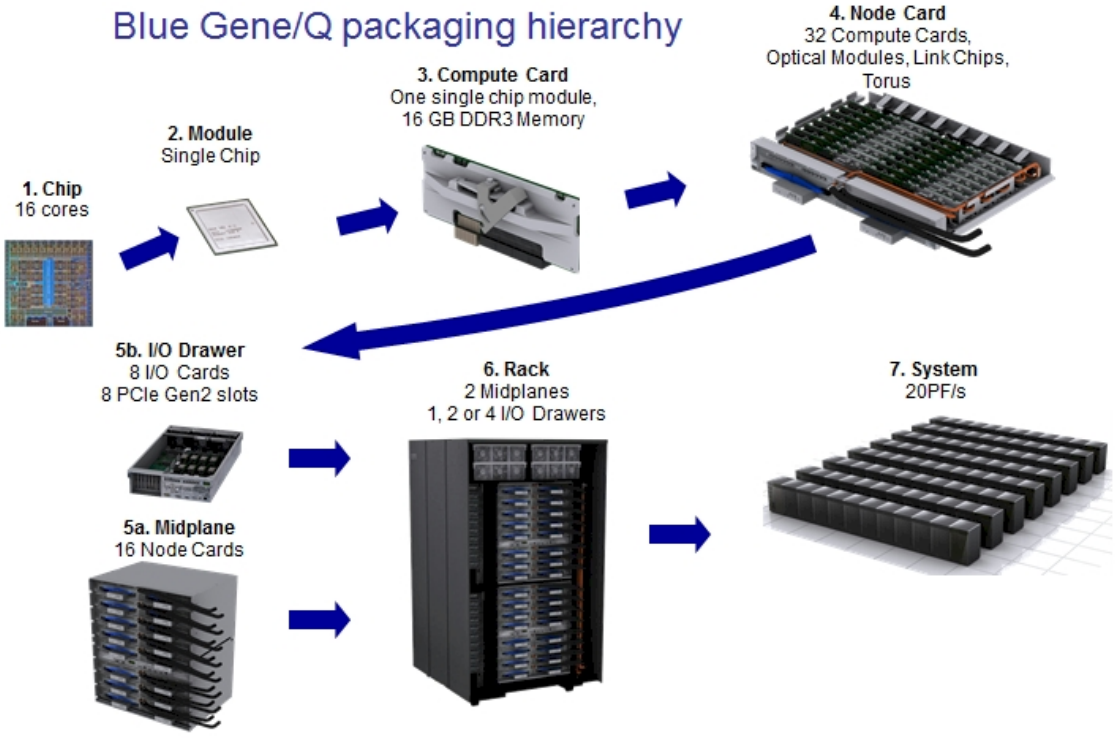

GREAT WALL OF COMPUTE
CHINA'S STILL #1.

WHAT IS A SYSTEM?

"SYSTEM: an assemblage or combination of things or parts forming a complex or unitary whole ..."


System Fan
Floppy
Heat Sink
Hard Drive
Power Supply
Optical Drive
Motherboard
Processors (CPU)
RAM Moduels


CAMERA
BEZEL
WIFI ANTENNA
LCD SCREEN
INVERTER BOARD
HINGES
INVERTER CABLE
LCD BACK COVER
KEYBOARD
SPEAKERS
TOUCHPAD BOARD
PALMREST
LED BOARD
LED BOARD COVER
BATTERY
CPU & VGA FAN /W HEATSINK
MINI PCI WIRELESS BOARD
MODEM BOARD
VGA BOARD
MAINBOARD
BLUETOOTH BOARD
USB BOARD CABLE
INFRARED BOARD
BOTTOM COVER
HDD
HDD COVER
RAM COVER
OPTICAL DRIVE
RAM
CPU
SCREW KIT
RUBBER FEET


Blue Gene/Q packaging hierarchy
1. Chip
16 cores
2. Module
Single Chip
3. Compute Card
One single chip module,
16 GB DDR3 Memory
4. Node Card
32 Compute Cards,
Optical Modules, Link Chips,
Torus
5b. I/O Drawer
8 I/O Cards
8 PCIe Gen2 slots
5a. Midplane
16 Node Cards
6. Rack
2 Midplanes
1, 2 or 4 I/O Drawers
7. System
20PF/s

WHAT ARE WE MISSING?

SOFTWARE!

WITHOUT IT…

SO WHAT IS SOFTWARE?

COMPUTER SYSTEM :

A rich and beautifully tapestry of **Interacting**
Hardware + Software components that are the substrate to our modern digital world

WELCOME TO CS210

- Course Objectives, Mechanics, and Syllabus
- Introduction to Computer Architecture (visible to programmer), and aspects of compilers and operating systems
- Not so much Computer Organization (not visible)

COURSE GOALS

- Understand components of computer systems
- Look under the hood
- Understand software / hardware interaction
- Understand design tradeoffs
- Use "real-world" examples to illustrate concepts

COURSE OBJECTIVES

- Understand assembly language
  - Intel architecture (IA32): x86, Pentium, …
  - Able to understand other architectures quickly
- Understand how C programs execute
- Understand how data is represented and operators applied to it

OBJECTIVES CONTINUED

- Learn functional units and interfaces
  - Processor, memory, Input/Output
  - Software
  - Interfaces
- Understand major sources of performance, e.g., how to exploit locality in code
- Prepare for future courses

WHY?

- Classical Reasons
  - You want to call yourself a "computer scientist"
  - You want to build software people use (need correctness and performance)
  - You need to make a purchasing decision or offer "expert" advice
  - It's Required ;-)
- Other Reasons
  - To be Masters of the Digital universe you must demystify the magic
  - Computer Systems are just F'ing cool and a marvel and testament to human ingenuity

PIAZZA

- Piazza will be the main forum for communication and online interaction -- Use it to ask your questions

  - https://piazza.com/bu/fall2014/cs210/home

    - Who has taken 131?
    - Who has taken 112?

SYLLABUS

- https://piazza.com/bu/fall2014/cs210/home

DISCUSSIONS & QUIZZES

- Discussion's are a fundamental aspect of this course — small setting in which you can get the details sorted out
- There will be some graded quizzes in discussions
- Union of Lectures and Discussions cover the material for the course.
- This we also have a Undergraduate Assistant (UA) who will hold office hours and additional info sessions

GENERAL ADVICE

- Start assignments early and don't underestimate the amount of work.
- This is your chance to explore your curiosity ... ask questions and "tinker".
- Becoming and expert at anything takes a lot of practice.
- We love this stuff... make use of use we are happy to help (we are available 10hr/week out side of lectures + discussions)
- Don't over thinking things machines are dumb.
- The basics are critical ... squash miss-understandings and problems early!

WHAT DO SYSTEMS COMPUTERS REALLY RUN ON?

WHAT DO SYSTEMS COMPUTERS REALLY RUN ON?

COFFEE!

- Please drop by and have some with me Monday's 1:00 - 2:00 at Pavement.

NEXT FEW LECTURES

- Thursday: Overview of Computer Systems — **READ CSAPP:Ch. 1**
- Monday Discussion and Next Tuesday's Class — "C" and Systems Programming — **READ K&R Ch. 1 and Sections 4.1-4.5,5.1-5.6,6.1-6.4 — First Problem set will be assigned**
- After that we will move on to a proceed through the Basics of computational hardware and Data Representation — **CSAPP: Ch 2. — keep up with the readings**

MAKE SURE YOU CAN GET TO THE PIAZZA COURSE SITE

READ AND SIGN SYLLABUS.
(http://www.cs.bu.edu/~jappavoo/Resources/210/syllabus.pdf)

GO GET YOUR COMPUTER ACCOUNT SETUP!
EMA 302
(http://www.cs.bu.edu/labs/)

DO READINGS

WHO AM I AND WHY AM I HERE?

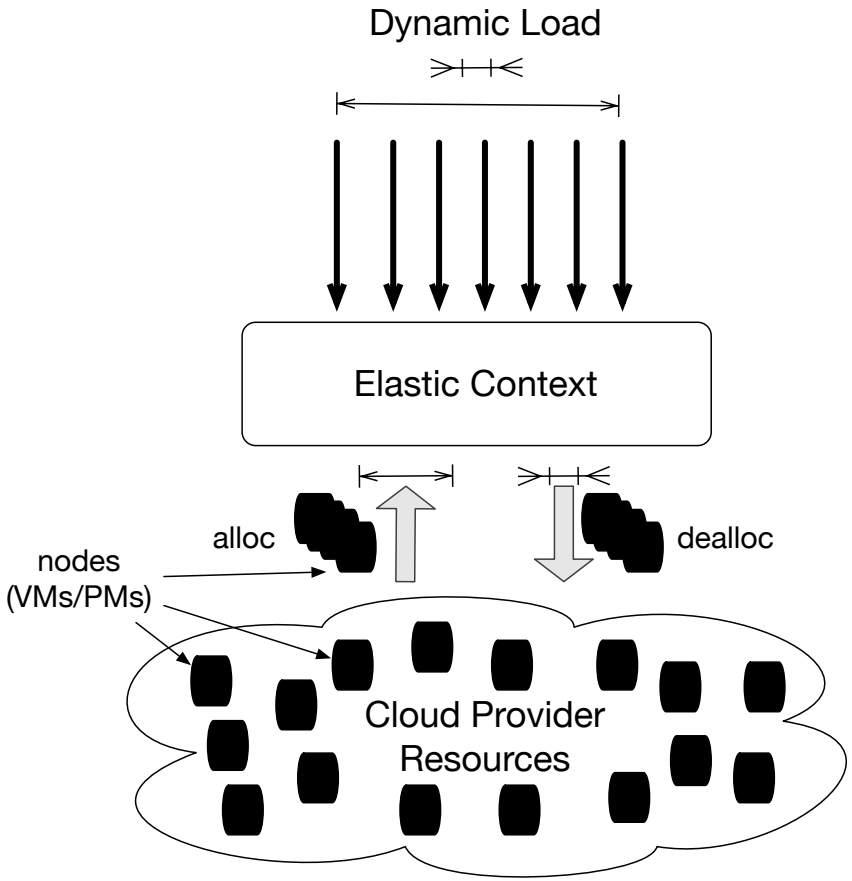
MY LIFE IN COMPUTING

- Built and programmed my first 3 computers starting when I was 10 with my dad.
- My first UNIX account — Love at first sight and scaring the !@#$%* out of me
- 16 months as a mainframe programmer (old but not that old)
- IBM AIX support — Vision Lab and Playbot
- NUMachine & Tornado — Research Large scale Multi-Processor and OS
- K42 — IBM follow on to the Tornado OS (my PhD)
- Libra — A new kind of OS for data-center applications — JAVA, Mapreduce
- Kittyhawk and the Global Cloud Computer

WHAT I AM CURRENTLY WORKING ON

- Elastic Building Block Runtime — EbbRT
- Programmable Smart Machines — PSM

The Cloud: A platform for a new class of Interactive High-Performance Web Apps



Elastic Backend View of Cloud Computing

EbbRT a unique runtime for Enabling Elastic Application Acceleration

EbbRT

- Allows developers to easily stitch in scalable elastic computation into their apps.
- App function can have full control of the hardware -- shedding full OS's when they are not necessary (nodes naturally become light-weight custom accelerators).
- Provides a Distributed Object Model that enables composition, customization and reuse of optimized distributed data-structures and algorithms.
- Built with Multi-core and Multi-node optimization in mind.
- First class low-level support for Event Driven Programming.

EbbRT well suited to directly hosting managed environments on the bare-metal: Java, LUA, Python and construction of new distributed custom ones
eg. Distributed In-Memory Hash Tables

The Loop

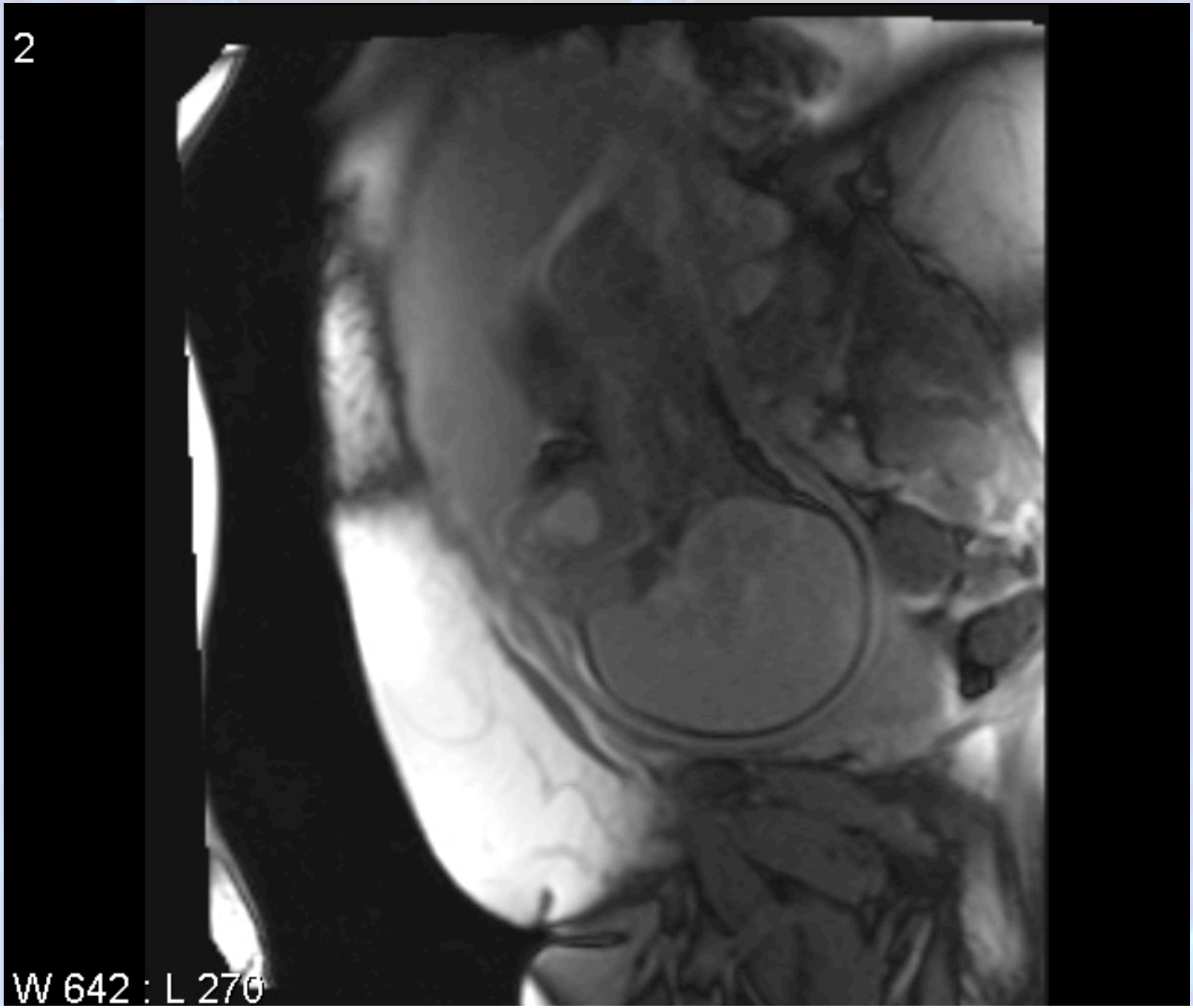
Patient

Feedback

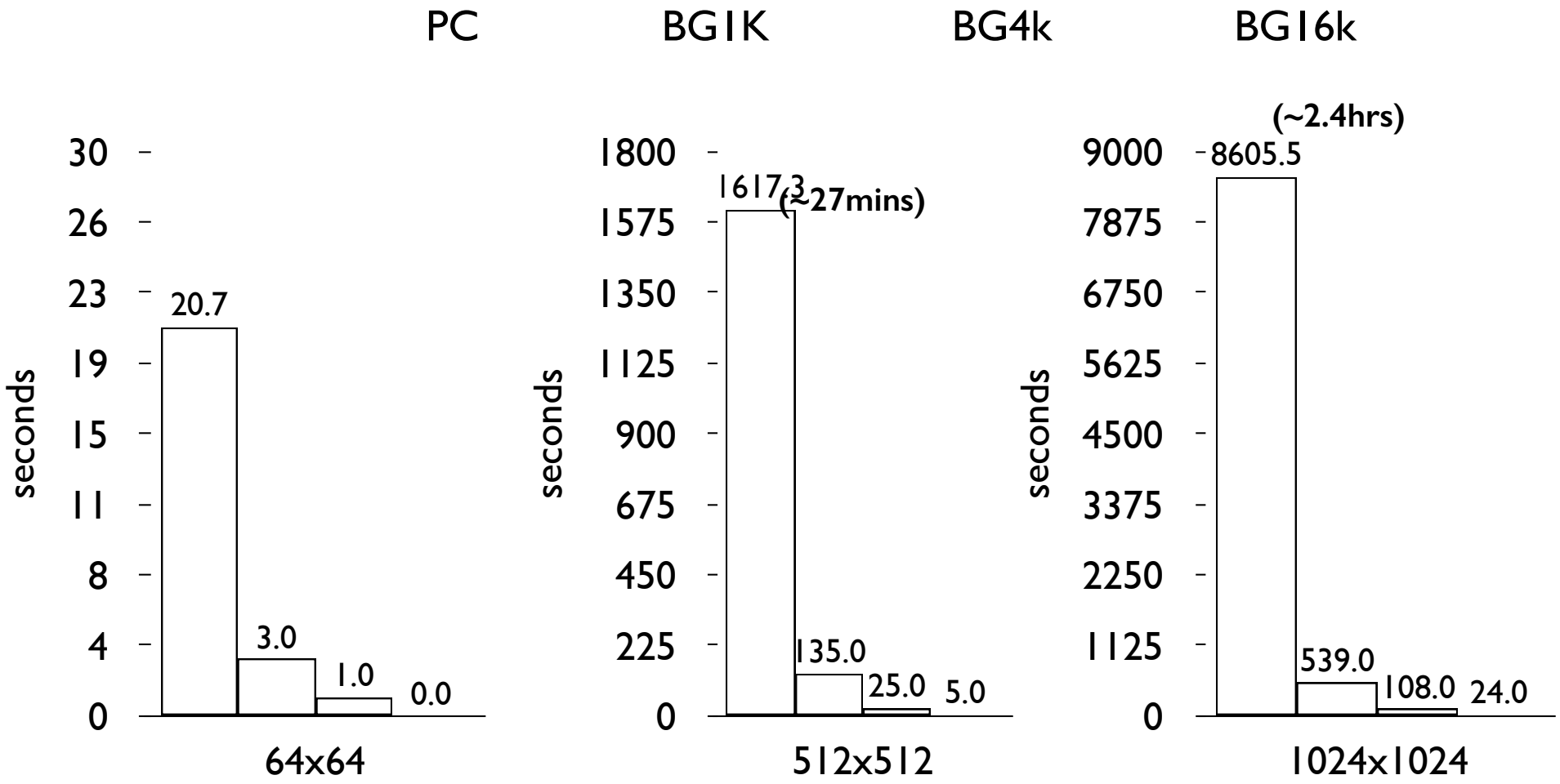
Image

Analysis & Diagnosis

Motion is a BIG Problem!



Results


PC
BG1K
BG4k
BG16k
seconds
30
26
23
19
15
11
8
4
0
20.7
3.0
1.0
0.0
64x64
seconds
1800
1575
1350
1125
900
675
450
225
0
1617.3
(~27mins)
135.0
25.0
5.0
512x512
seconds
9000
7875
6750
5625
4500
3375
2250
1125
0
(~2.4hrs)
8605.5
539.0
108.0
24.0
1024x1024

In Pursuit of a New Kind of Computer

Programable Smart Machine Lab (PSML)
Jonathan Appavoo, Boston University

"And now for something completely different..."

PSML
Programmable
Smart Machines
Lab

Properties

Simply and yet Richly Programmed

Automatically improves with its size

Automatically improves/adapts with experience

Amenable to implementation with low power devices

A programmable child that is pedantically obedient without the attitude :-)

Hmmm Now What?

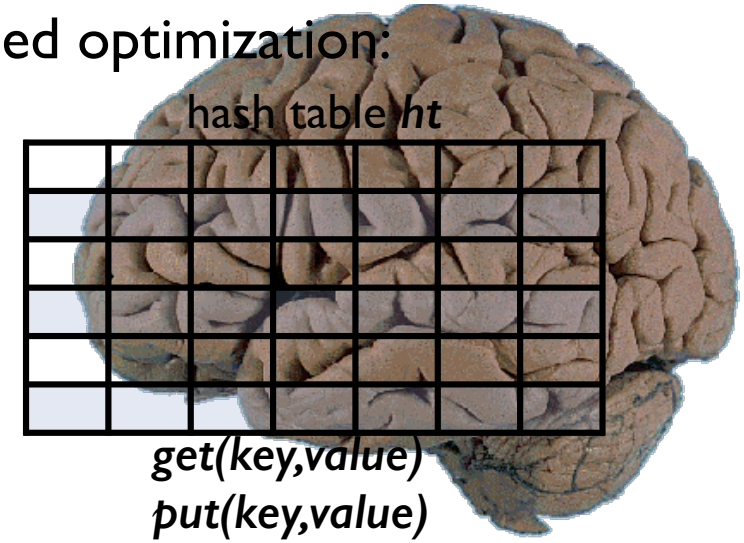
~~Simply and yet Richly Programmed~~

Automatically improves with its size

Automatically improves/adapts with experience

Amenable to implementation with low power devices

Using knowledge of runtime behavior introduce a cache based optimization:




```
key = hf(s);
if (get(key,&value)==hit) {
  fast(value);
} else {
  slow(&value);
  put(key,value);
}
```

Yikes...

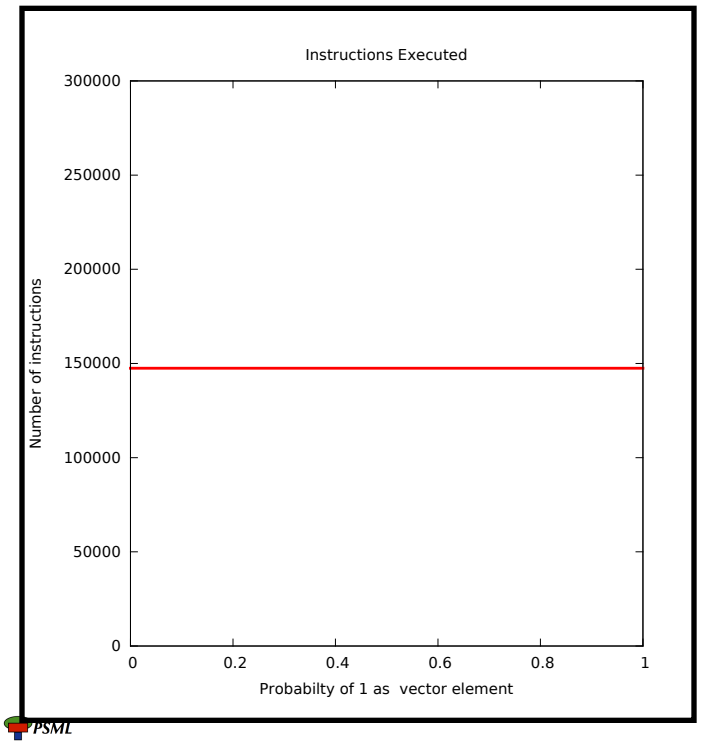
I did warn you ;-)

3 x 7

Filter

inner 0.1 Execution

```
4001ca  4831ed              xor rbp, rbp
4001cd  5f                  pop rdi
4001ce  4889e6              mov rsi, rsp
4001d1  4883e4f0            and rsp, 0xfffffffffffffff0
4001d5  e8c6ffffff          call 0x4001a0
4001a0  55                  push rbp
4001a1  4889e5              mov rbp, rsp
4001a4  4883ec10            sub rsp, 0x10
4001a8  897dfc              mov [rbp-0x4], edi
4001ab  488975f0            mov [rbp-0x10], rsi
4001af  ba00200000          mov edx, 0x2000
4001b4  be00306000          mov esi, 0x603000
4001b9  bf00106000          mov edi, 0x601000
4001be  e881ffffff          call 0x400144
```

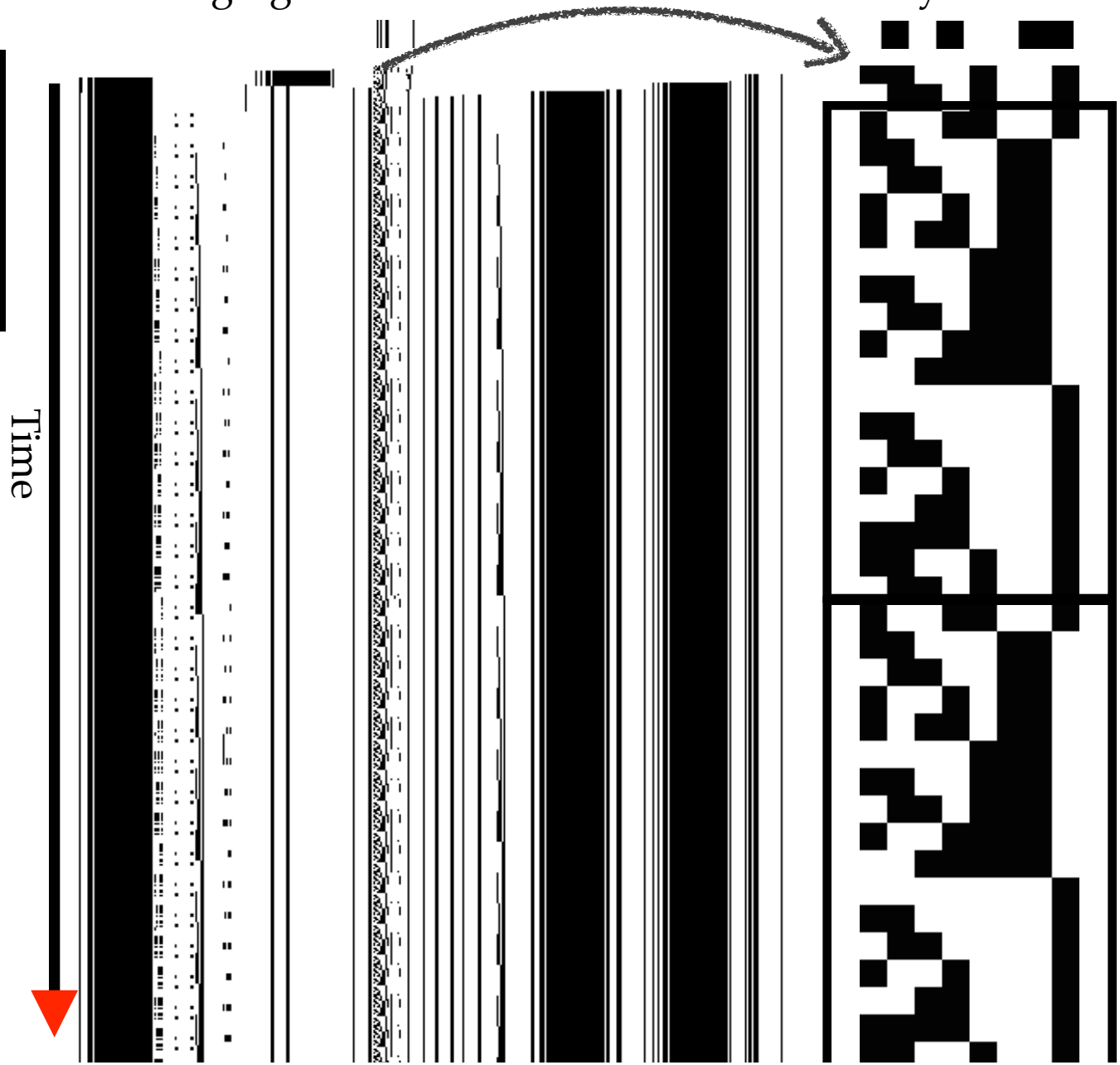


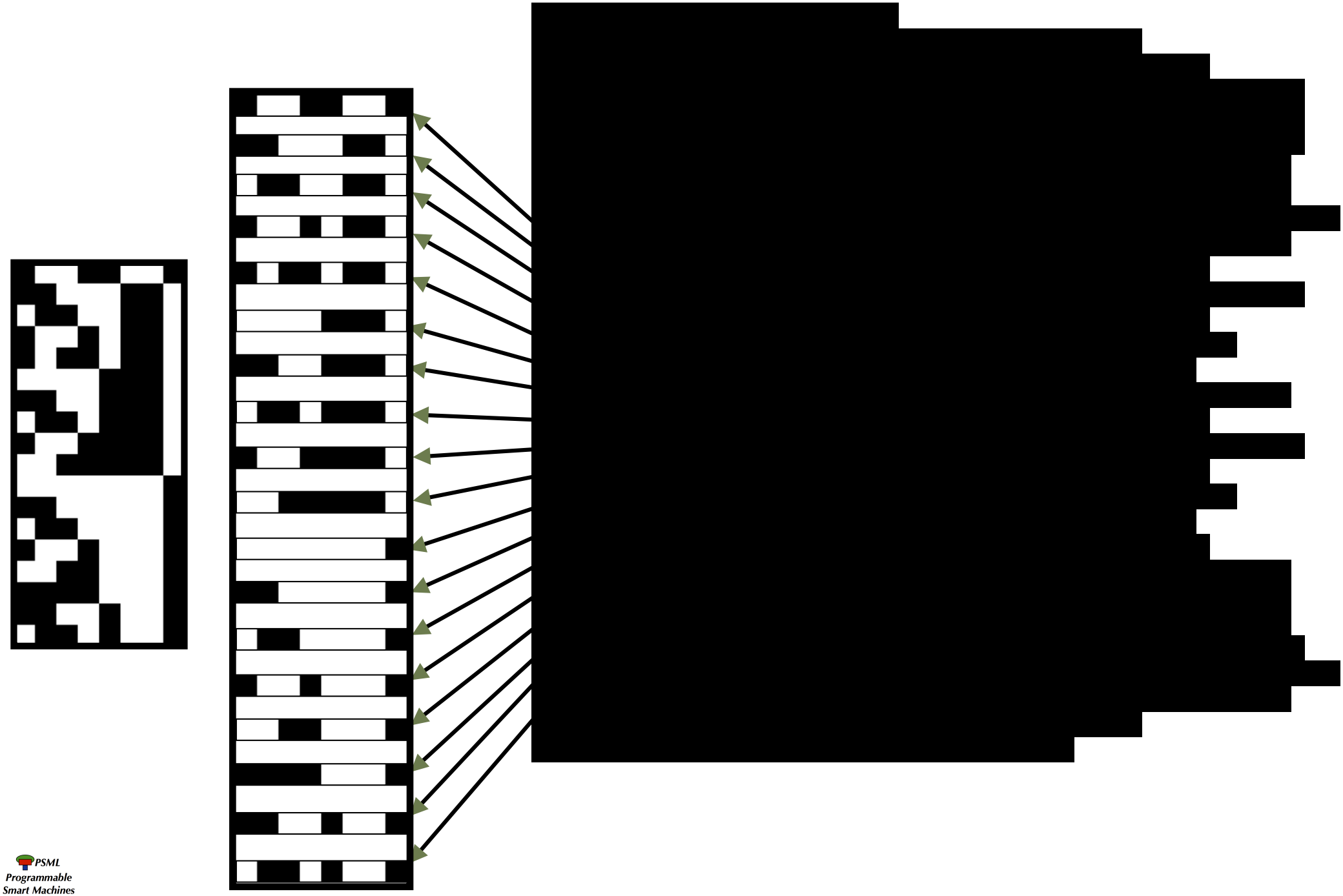
PSML
Programmable Smart Machines Lab

265 Changing Bits out of 1345216 State Bits

Byte 128

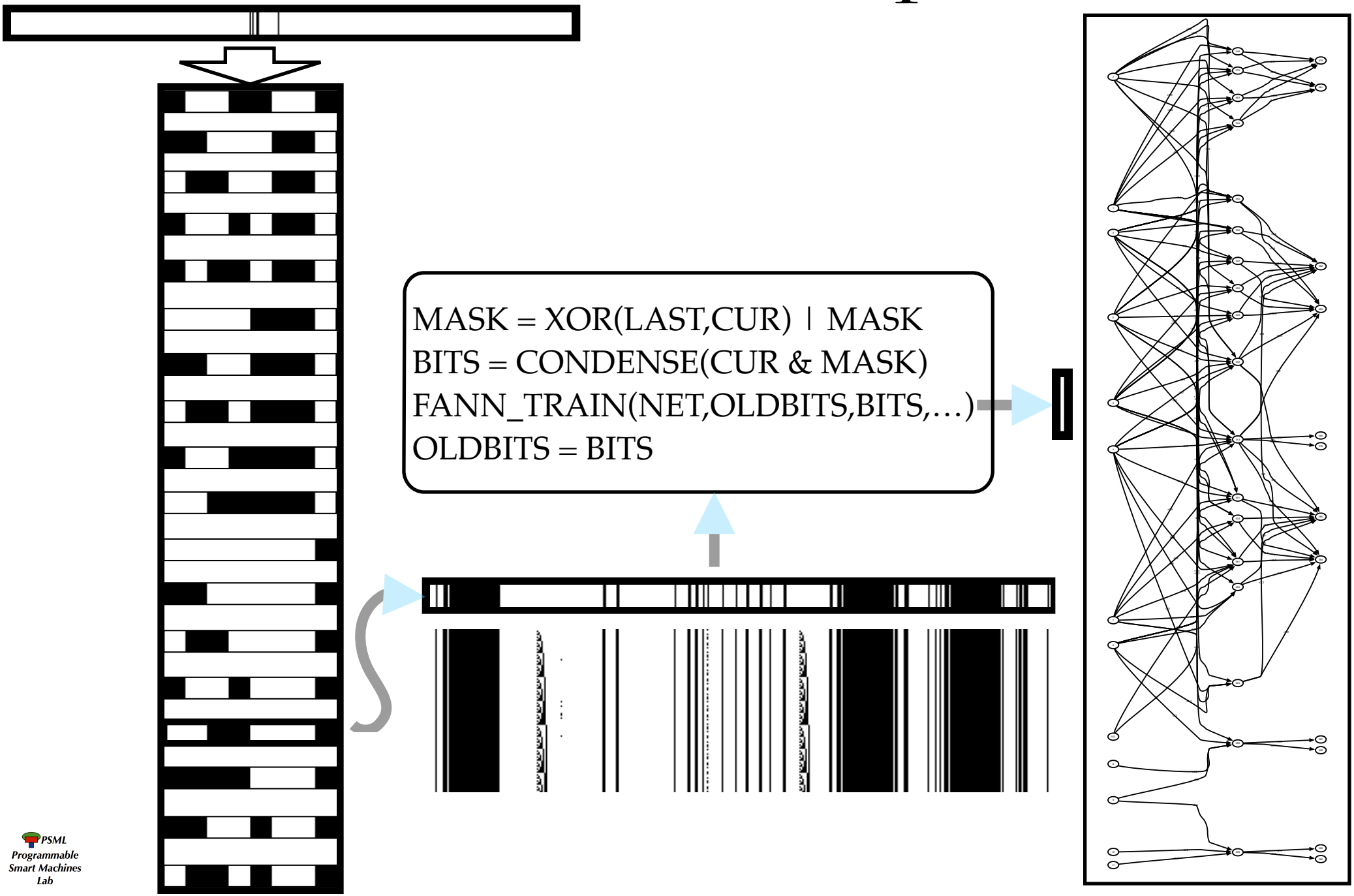
Time





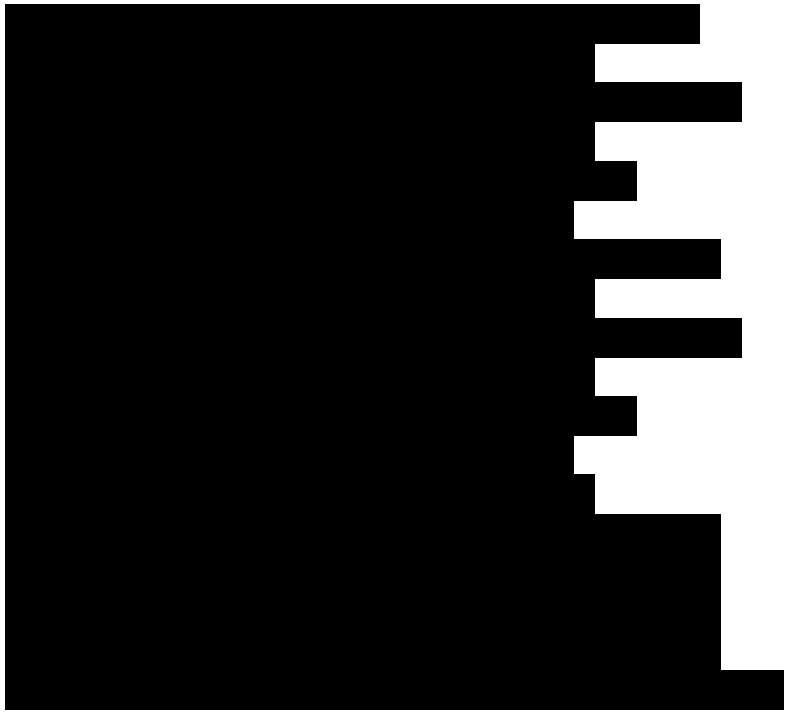
PSML
Programmable
Smart Machines
Lab

Train – Online Update

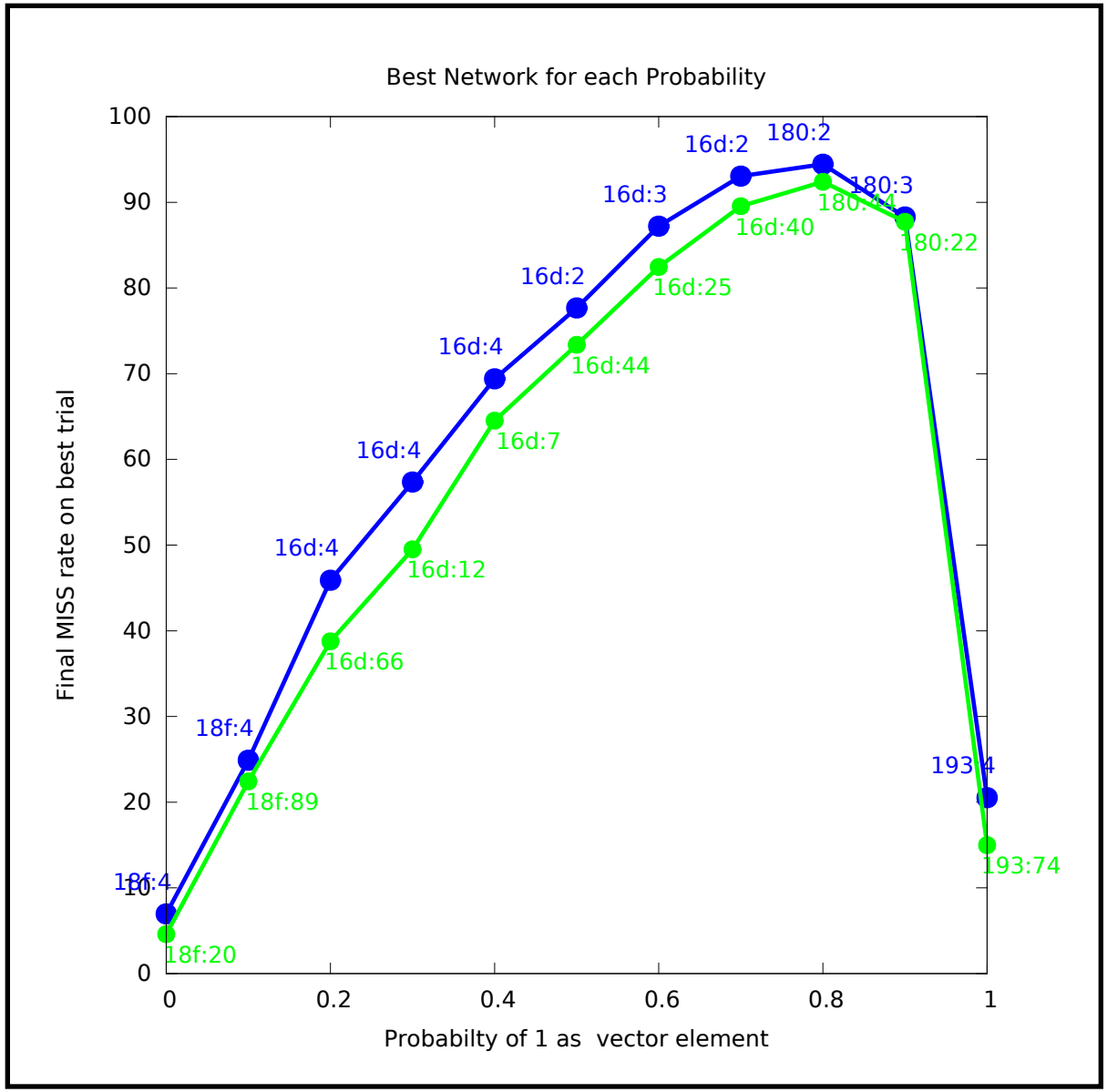


PSML
Programmable
Smart Machines
Lab

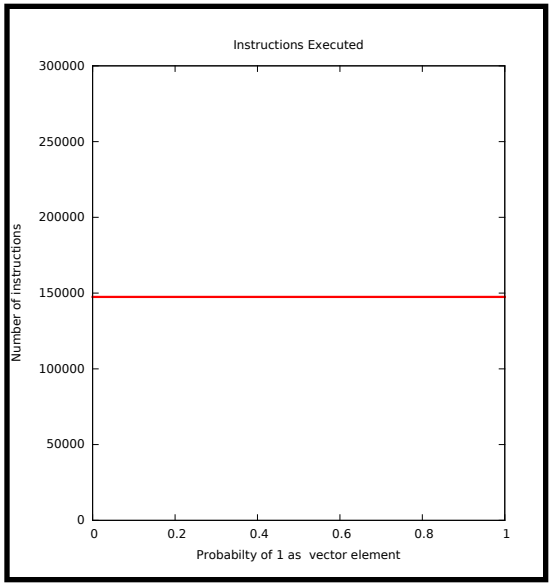
Predict

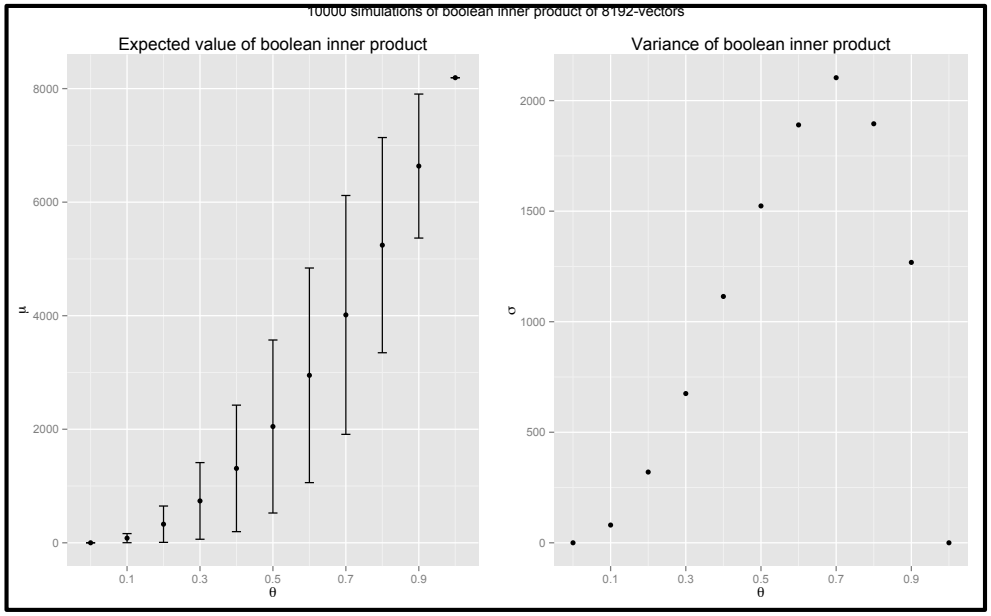


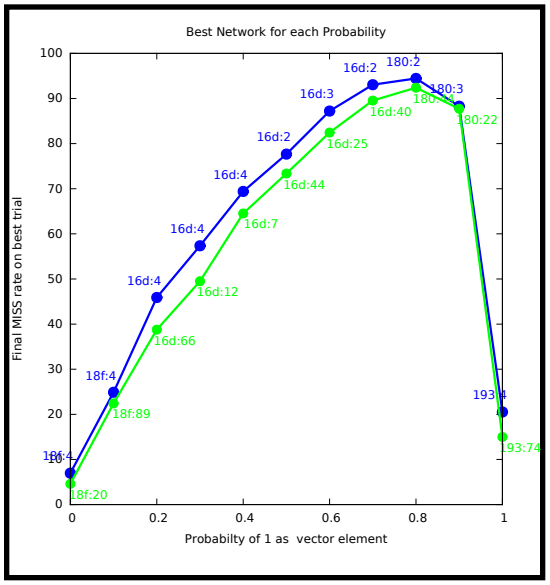
40018f	0.0,0.1
40016d	0.2-0.7
400180	0.8,0.9
400193	1.0



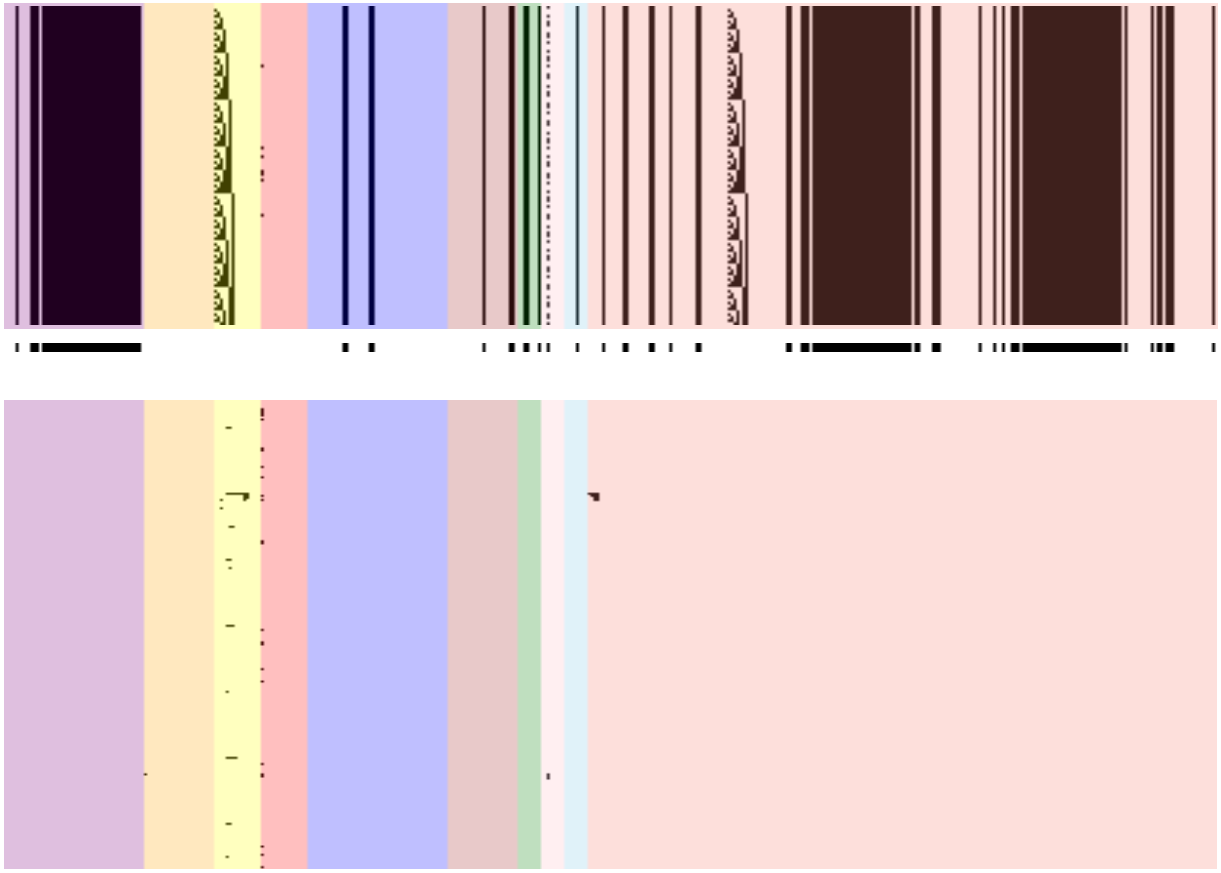
PSML
Programmable
Smart Machines
Lab







RBP	purple
RAX	orange
RCX	yellow
RDX	red
RSI	blue
RDI	brown
RIP	green
EFLAGS	pink
RSP	skyblue
[stack]	salmon



PSML
Programmable
Smart Machines
Lab