

# The *Mercury* System: Embedding Computation into Disk Drives

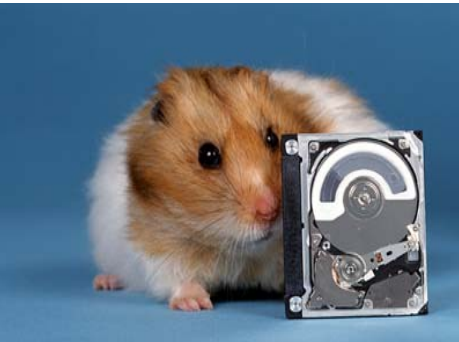
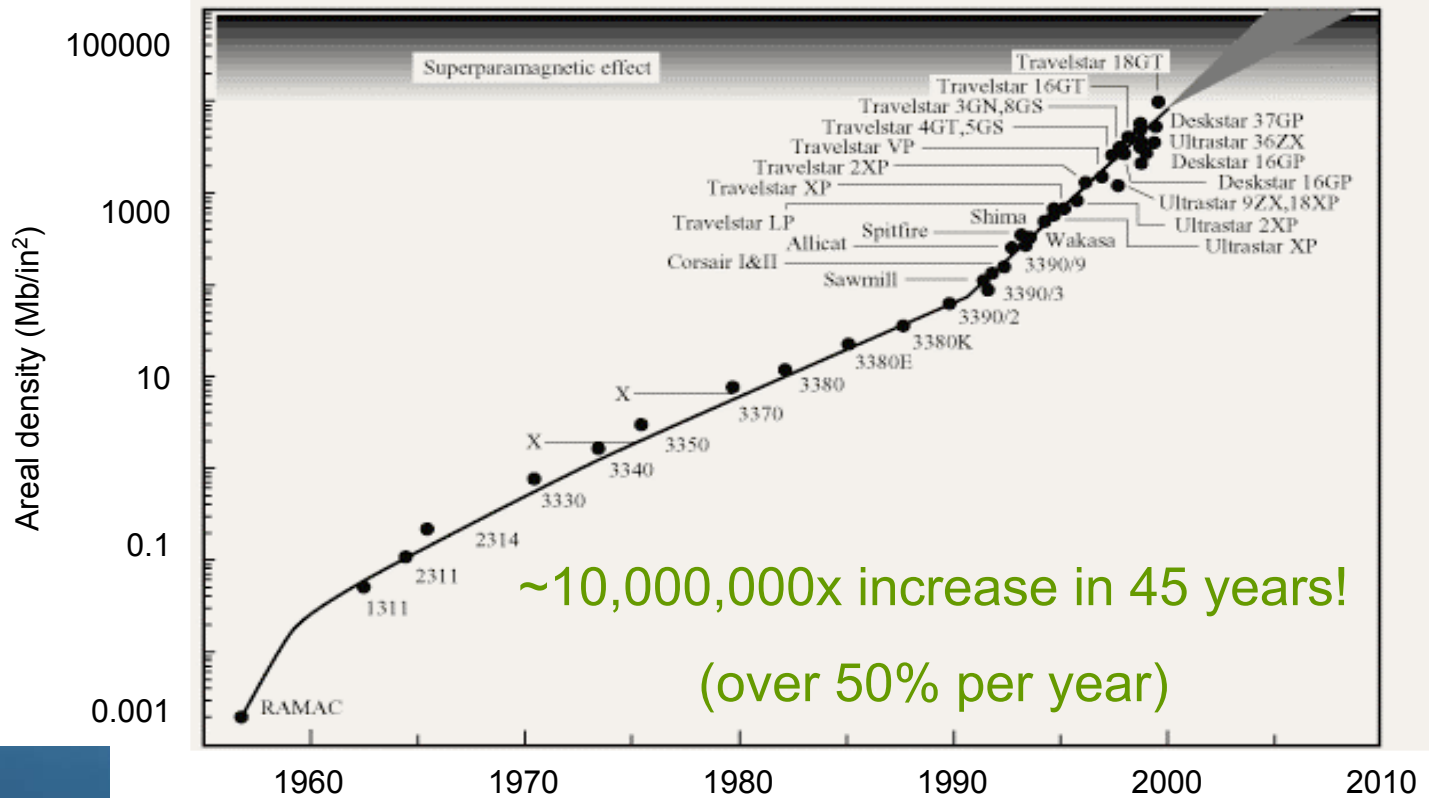
---

Roger Chamberlain, Ron Cytron,  
Mark Franklin, Ron Indeck

Center for Security Technologies  
Washington University in St. Louis

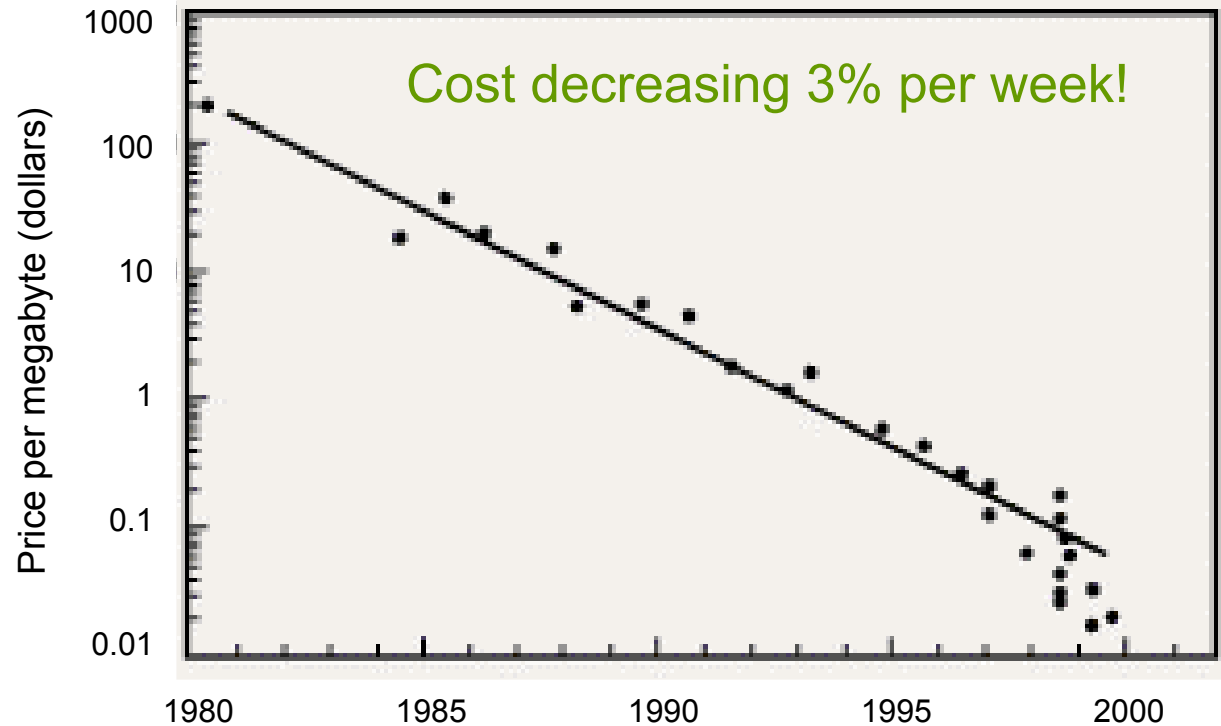


# Enabling Technology: Disk Drives



Magnetic disk storage areal density vs. year of IBM product introduction (From D. A. Thompson)

# Cost per Megabyte



Price history of hard disk product vs. year of product introduction (From D. A. Thompson)

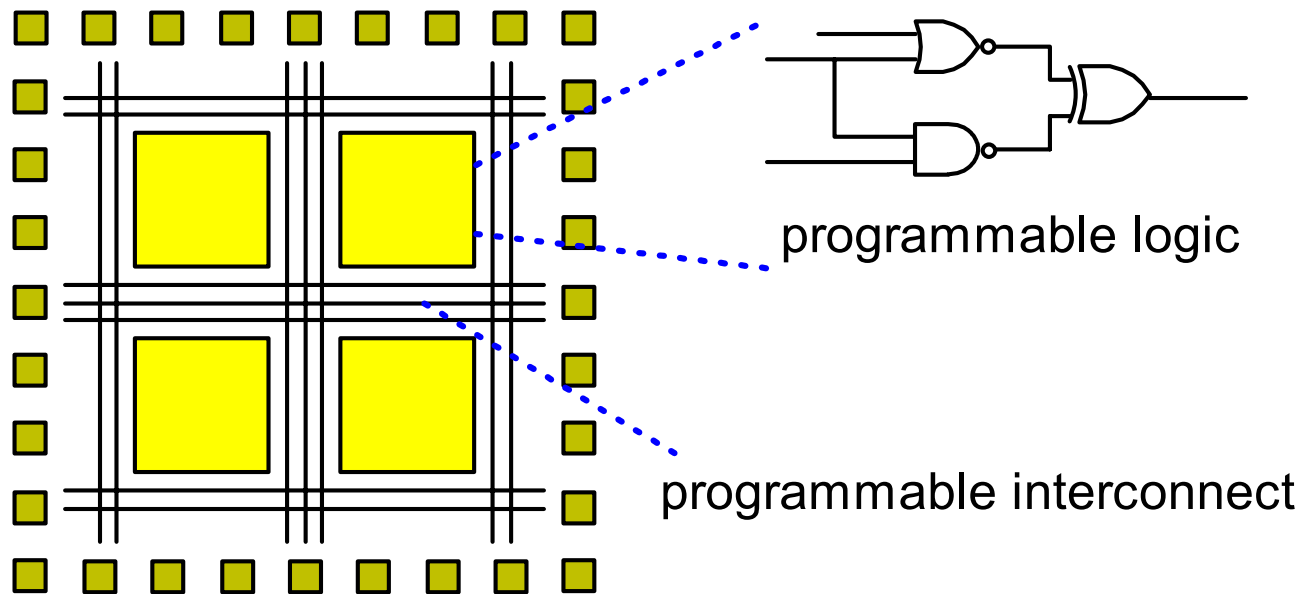
# Massive Data

---

- Storage industry shipped 4,000,000,000,000,000,000 Bytes last year
- MasterCard recently installed a 200 TByte data warehouse in St. Louis
- US intelligence services collect data equaling the printed collection of the US Library of Congress every day!



# Enabling Technology: Reconfigurable Hardware



- Field Programmable Gate Arrays (FPGAs) provide custom logic function capability
- Operate at hardware speeds
- Can be altered (reconfigured) in the field to meet specific application needs



# What are we doing?

---

Within the Center, we are combining the capabilities of these two enabling technologies to build extremely fast data search engines.

We do this by moving the search closer to the data, and performing it in hardware rather than software.



# Important Application: Intelligence Data

---



- Lots of data
  - Public (e.g., web pages)
  - Clandestine (e.g., via national technical means)
- Growing constantly
- Many perturbations of individual words
  - Tzar, Tsar, Czar, ...
- Query and field types aren't known *a priori*

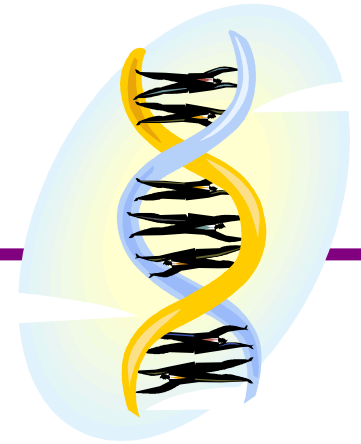
# Finding a needle in a haystack

---

- Text can contain errors
- Often seek an *approximate* match, e.g.  
    needle
- No match? Try 2-transpositions  
    enedle, needle, nedele, neelde, needel
- No match? Try 1-deletions  
    eedle, nedle, nedle, neele, neede, needl
- No match? Try insertions, larger edits, ...



# Genome Application

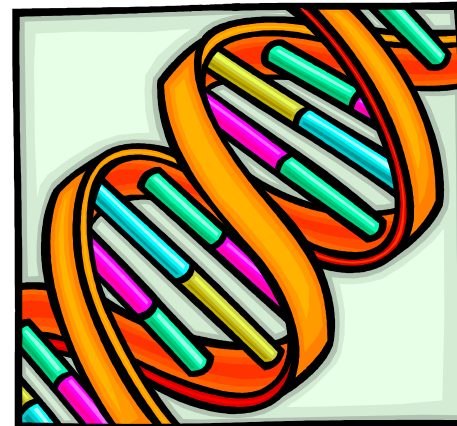


- Genome maps being expanded daily
  - 80,000 genes, 3 billion base pairs (A,C,G,T)
- Look for matches
  - Identify function
  - Disease: understand, diagnose, detect, therapy
  - Biofuels, warfare, toxic waste
  - Understand evolution
  - Forensics, organ donors, authentication
  - More effective crops, disease resistance

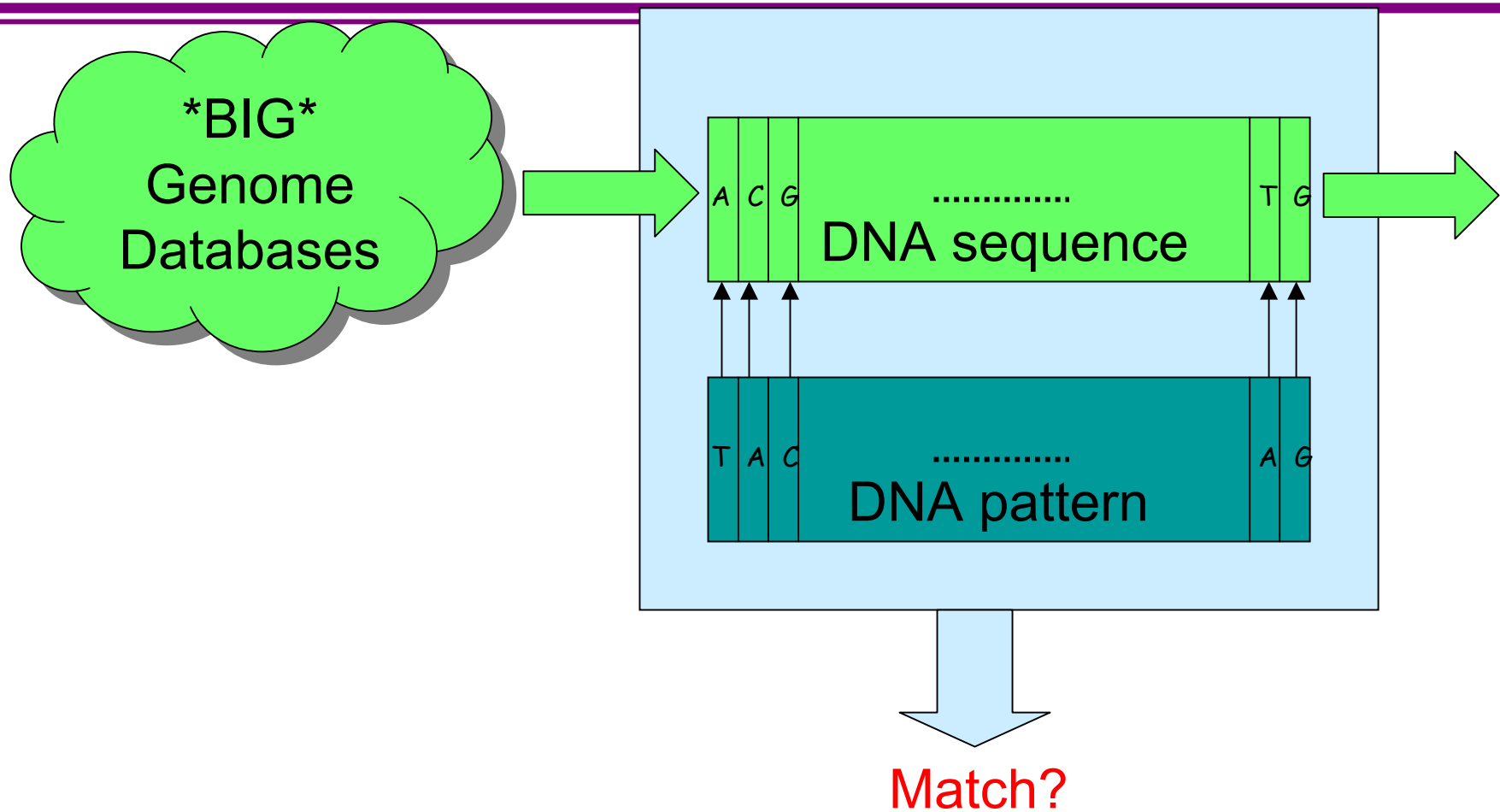


# DNA String Matching

- Looking for *CACGTTAGT...TAGC*
- Interested in matches and near matches
- Search human genome, other gene oceans
  - Need to search entire data sets



# Bio Computation Problem



# Image Database Applications

---

- Challenging database
- Unstructured
- Massive data sets
- Don't know what we need to look for in each picture



# Object Recognition

---

- Face recognition
- Match template with image
- Template database must be searched
- Strict time constraints for matching and overall search



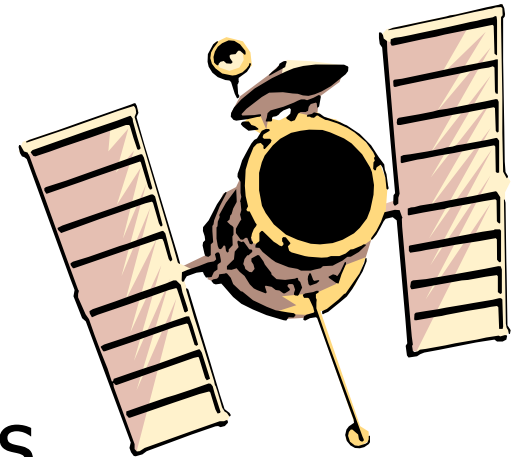
# Washington University Campus

---



# Satellite Data

- Low orbit fly-over every 90 minutes
- Look for *differences* in images
  - Large objects
  - Troops
  - Changes to landscape
- Flag, transmit these differences immediately





---

How do we find what we're looking for most effectively?!



# Conventional Structured Database

<i>D<sub>id</sub></i>	<i>Document</i>
1	Agent James Bond
2	Agent mobile computer
3	James Madison movie
4	James Bond movie

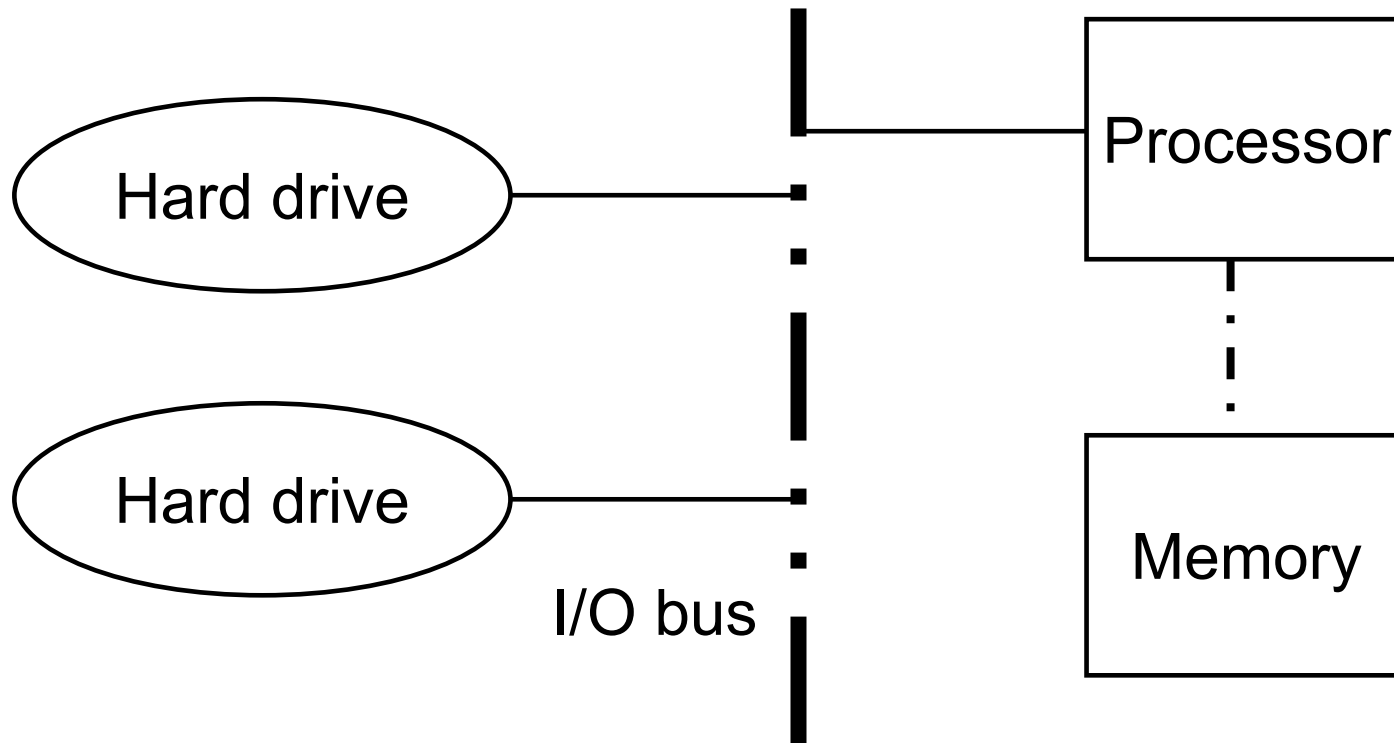
<i>Word</i>	<i>Inverted list - pointers</i>
agent	<1,2>
Bond	<1,4>
computer	<2>
James	<1,3,4>
Madison	<3>
mobile	<2>
movie	<3,4>

# Challenges in Searching These Massive Databases

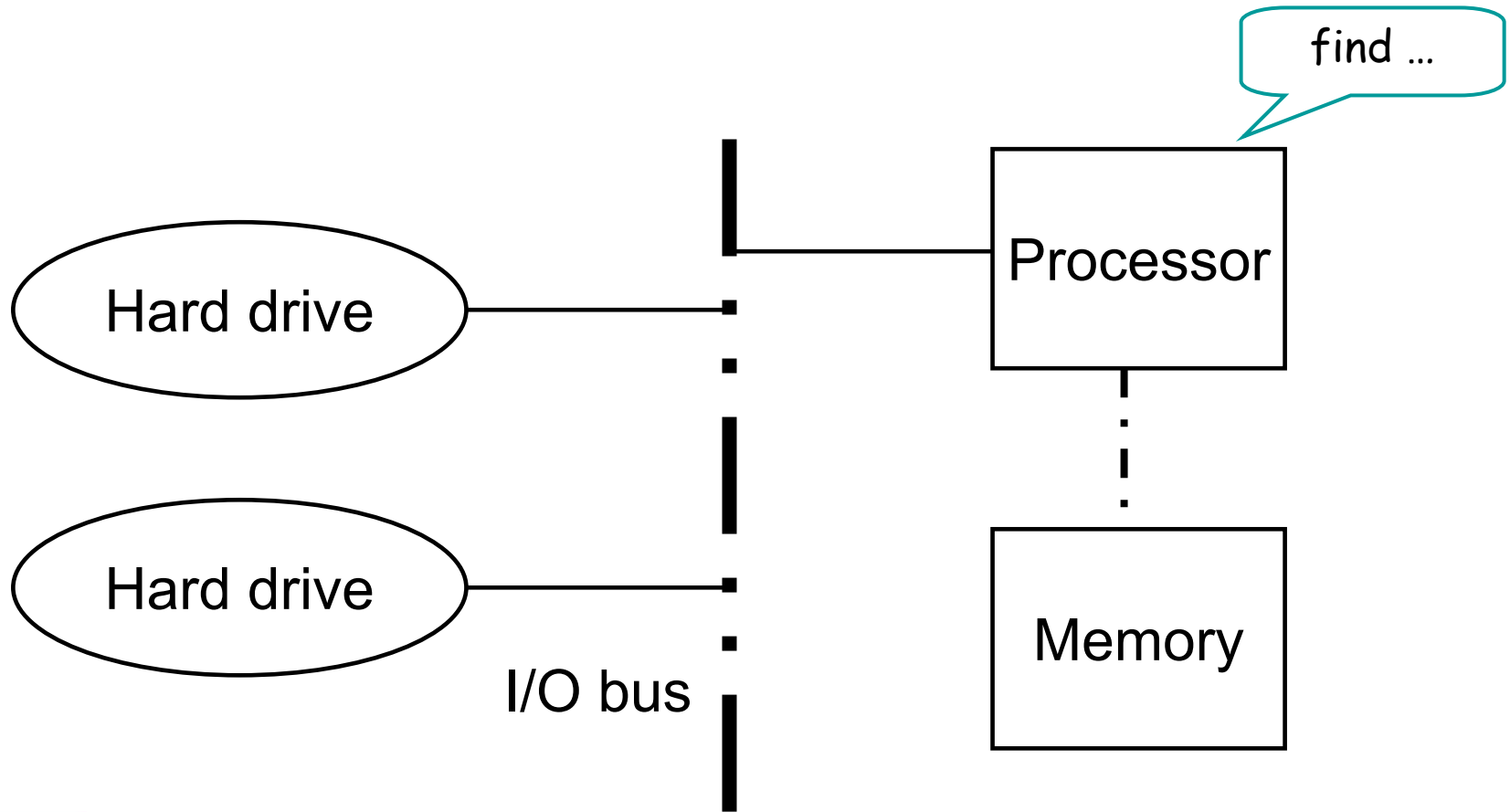
---

- If we know what we will be looking for
  - Need to build index beforehand
  - Maintain index as it changes
- If we ***don't*** know what we want *a priori*
  - Need to search the whole database!

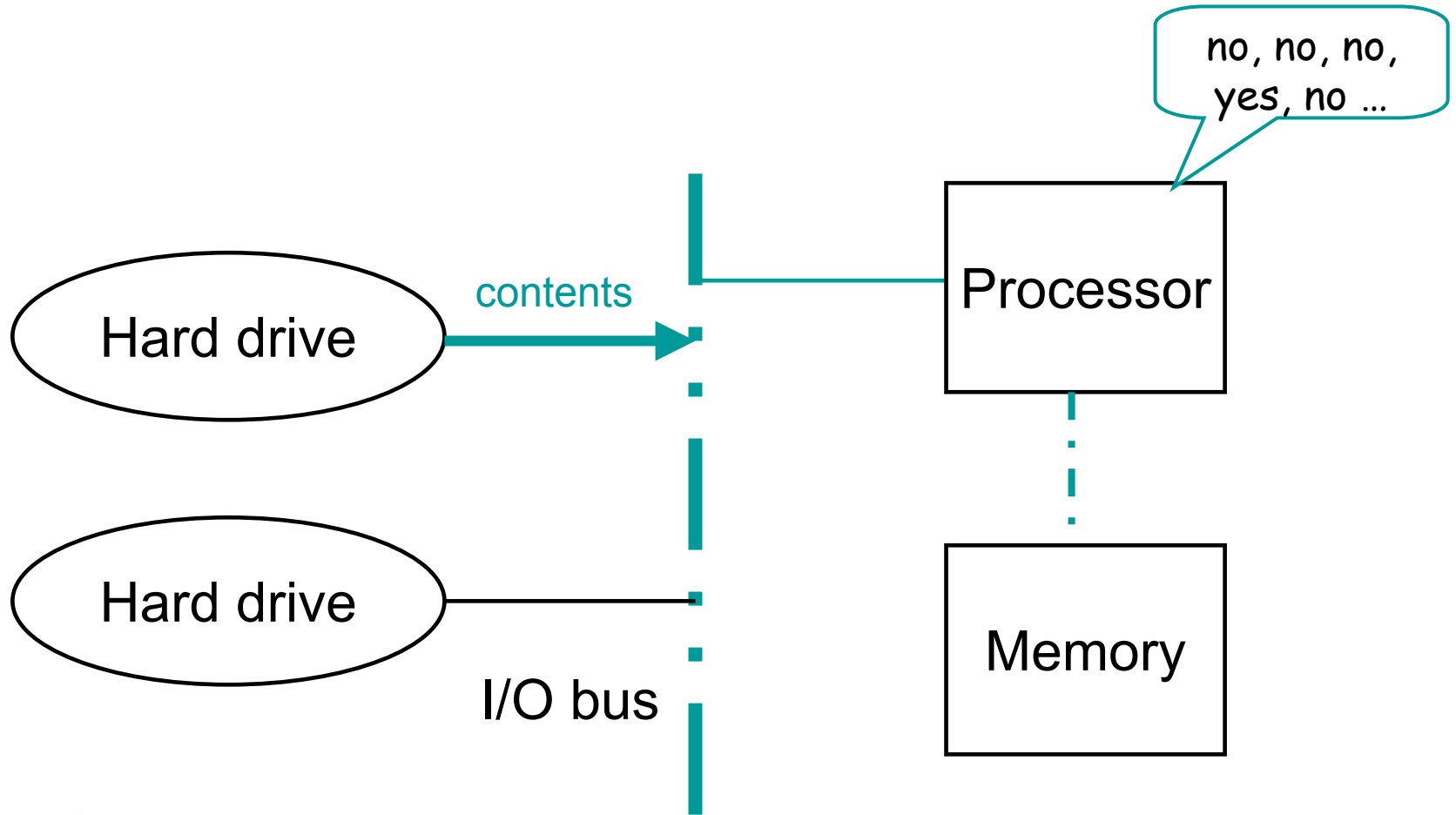
# Conventional Search



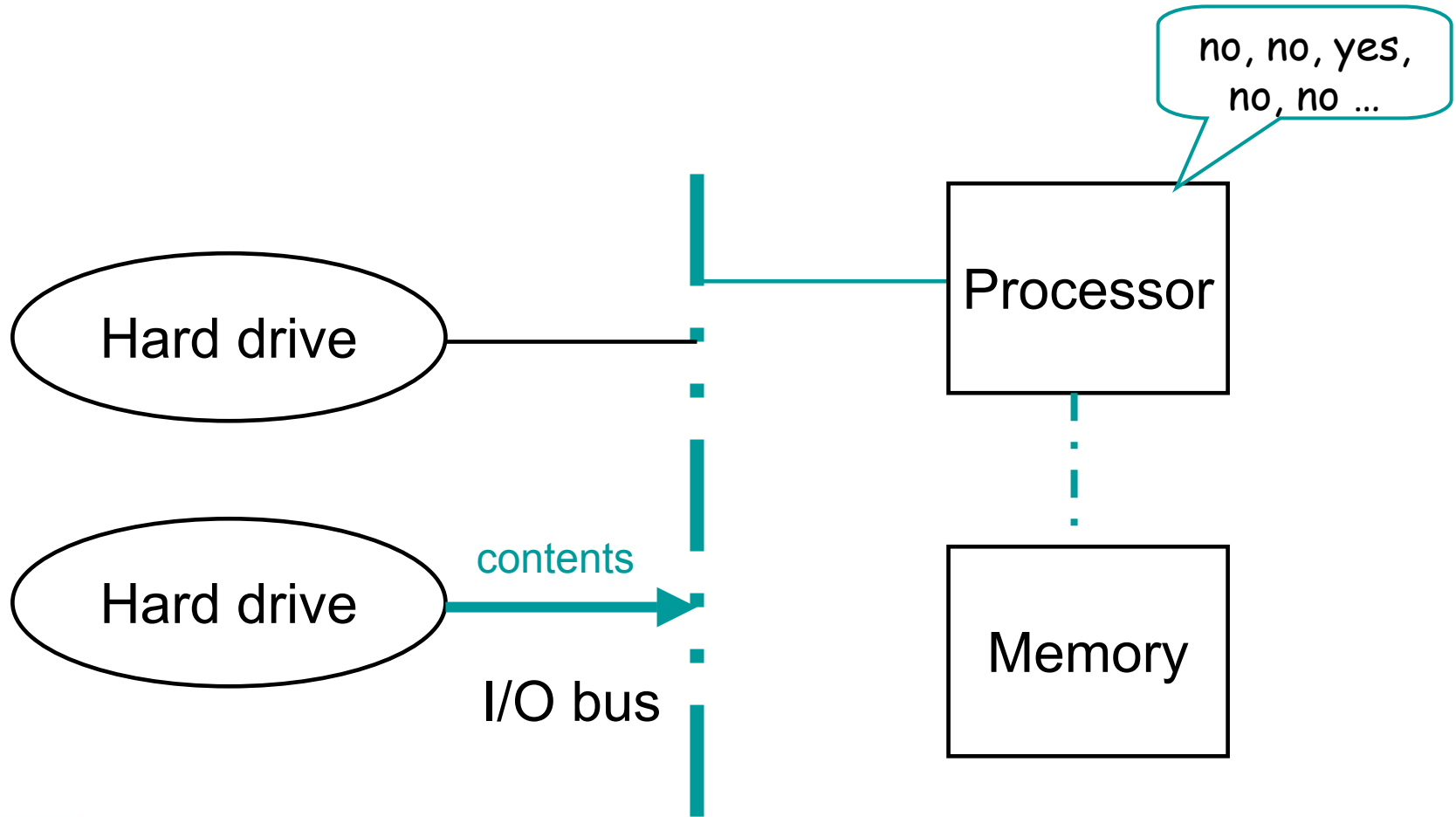
# Conventional Search



# Conventional Search



# Conventional Search



# Conventional Approach

---

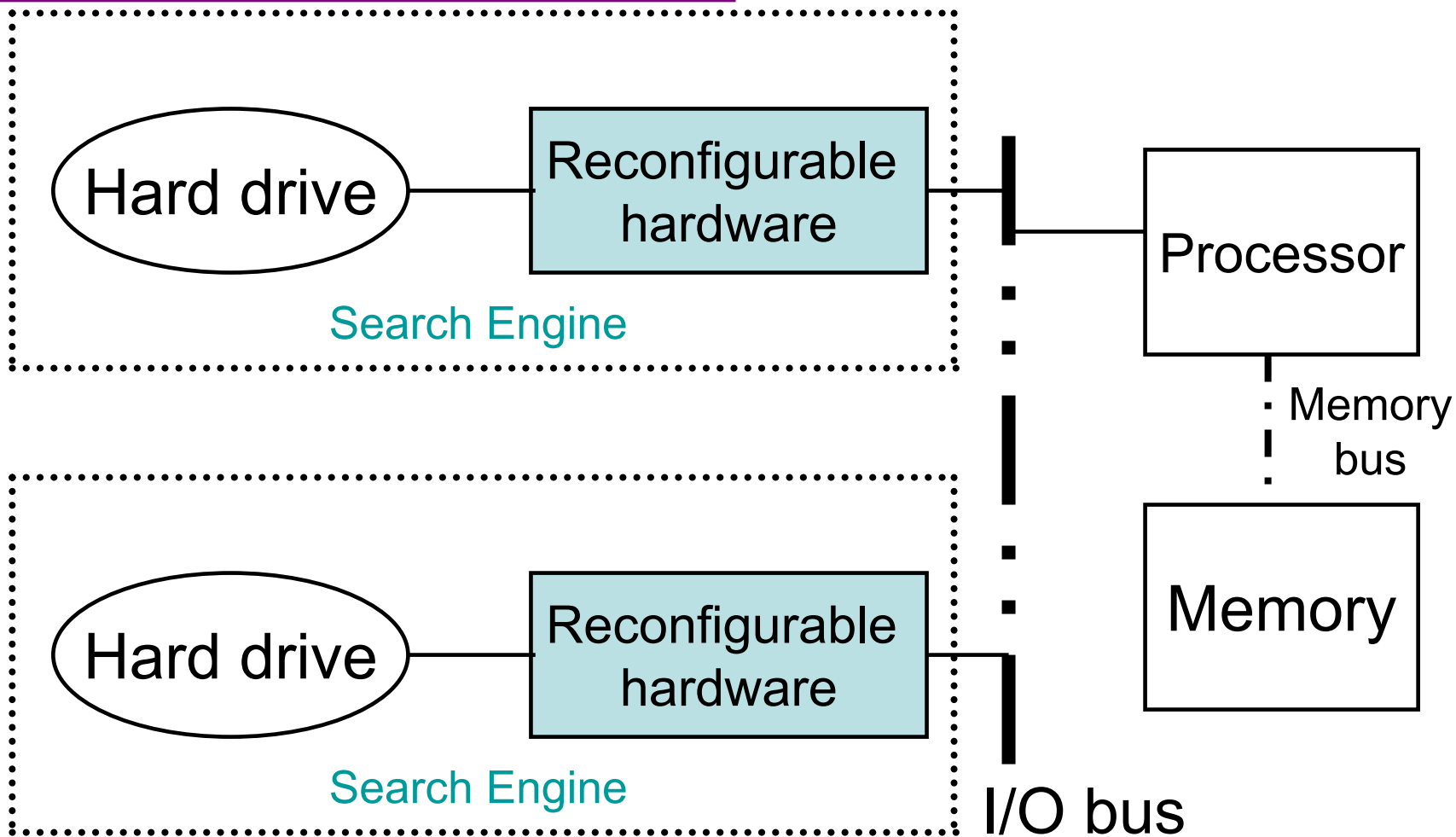


# WUSTL's Approach

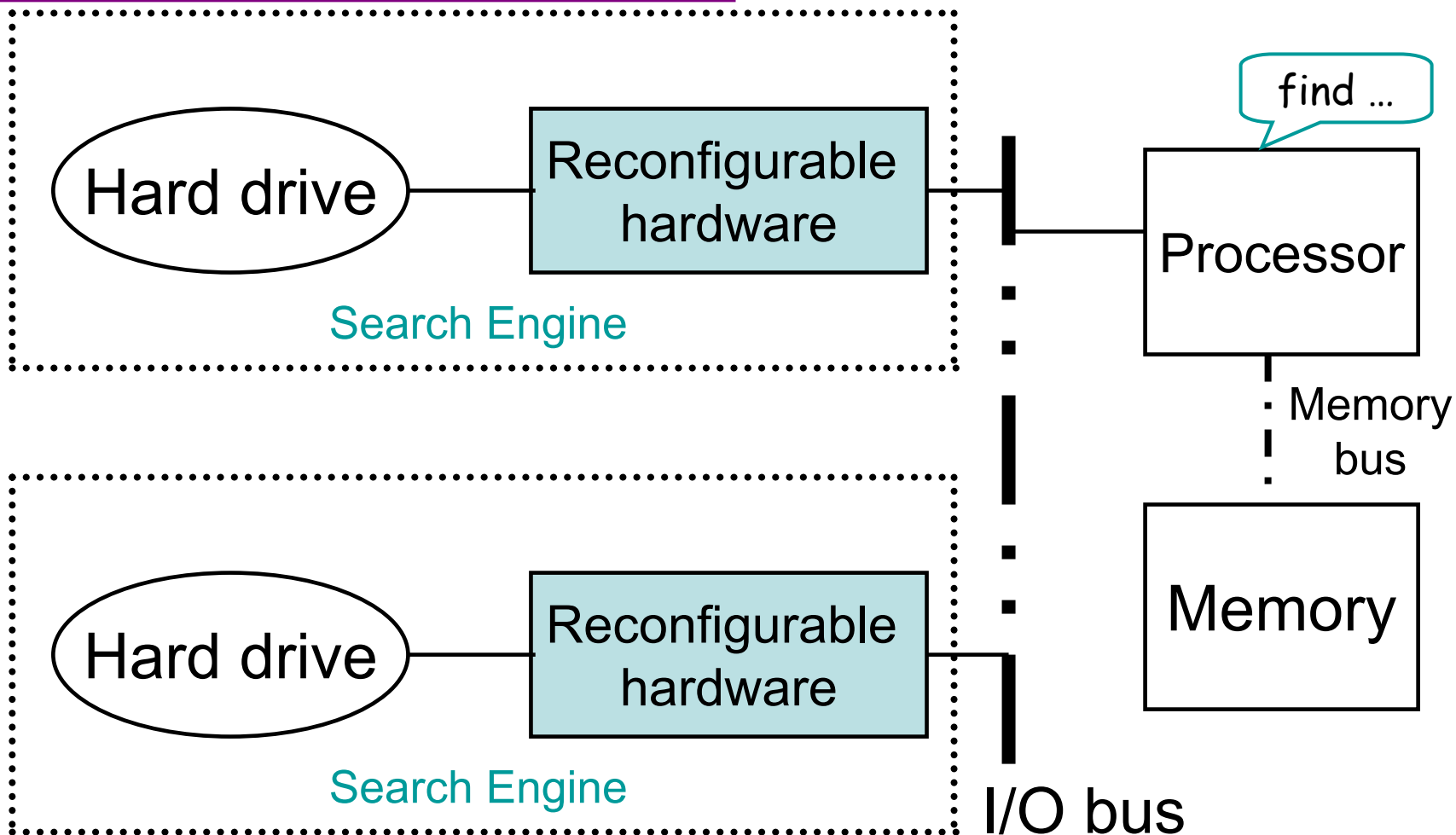
---



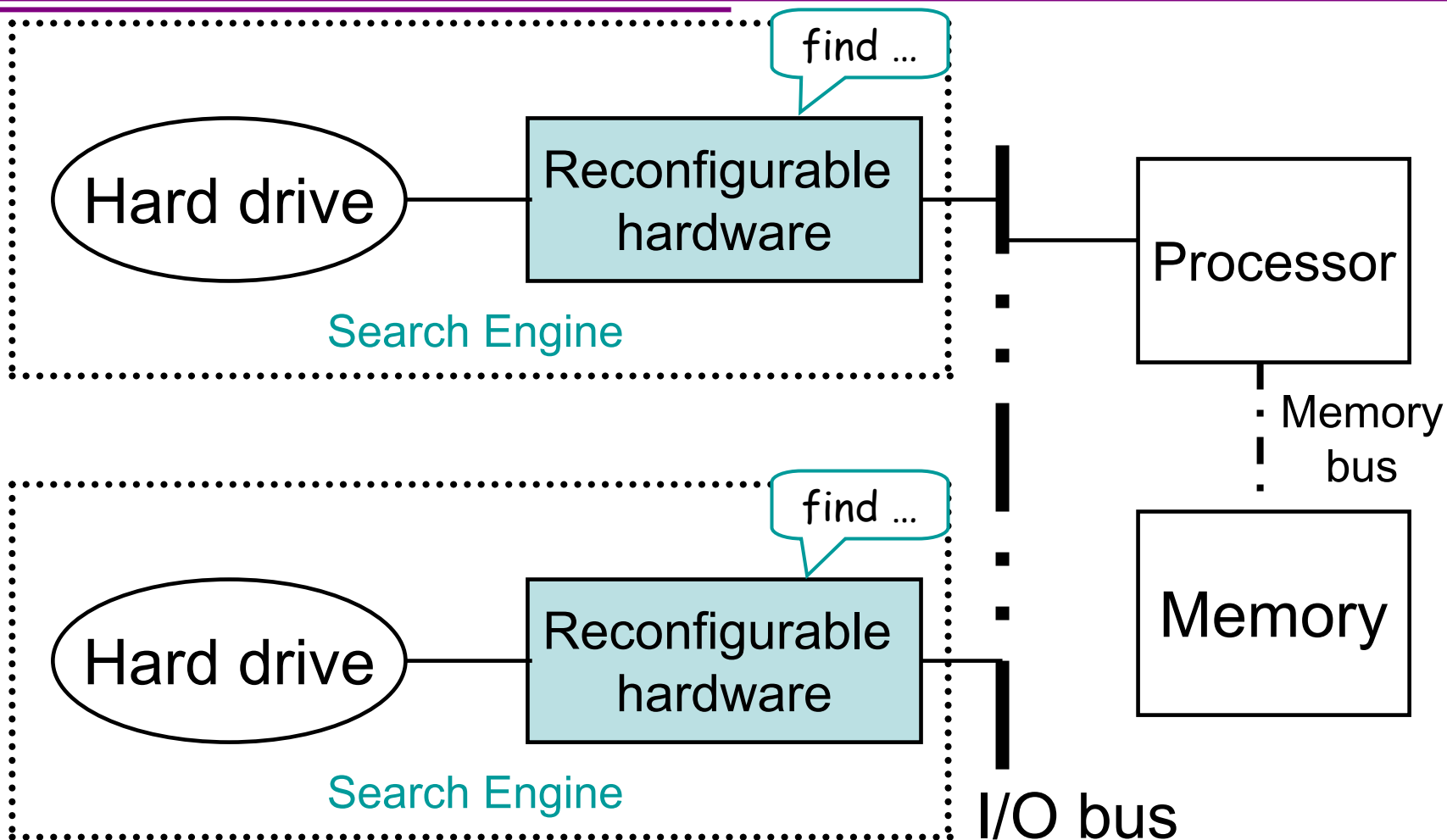
# Streaming Approach



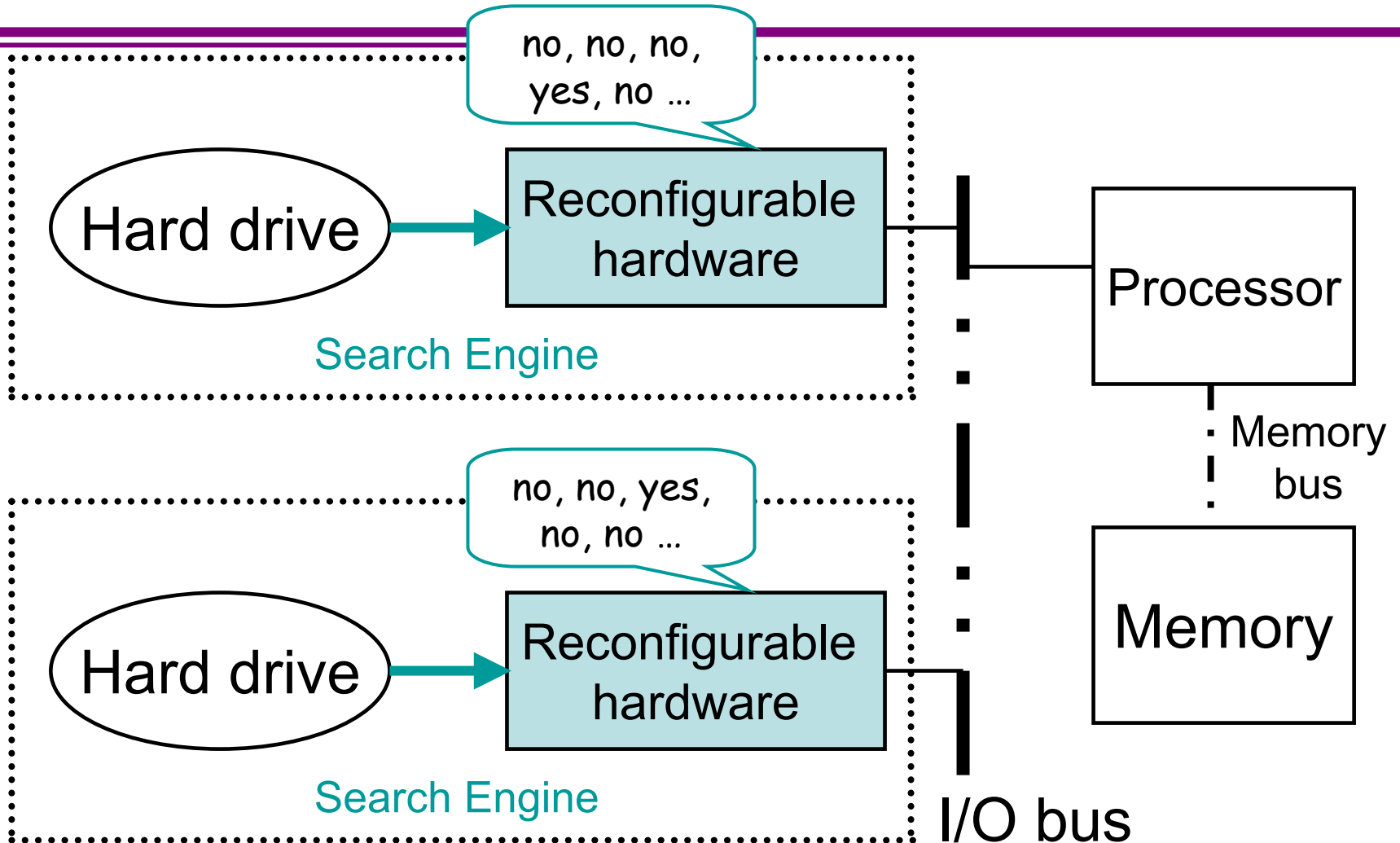
# Streaming Approach



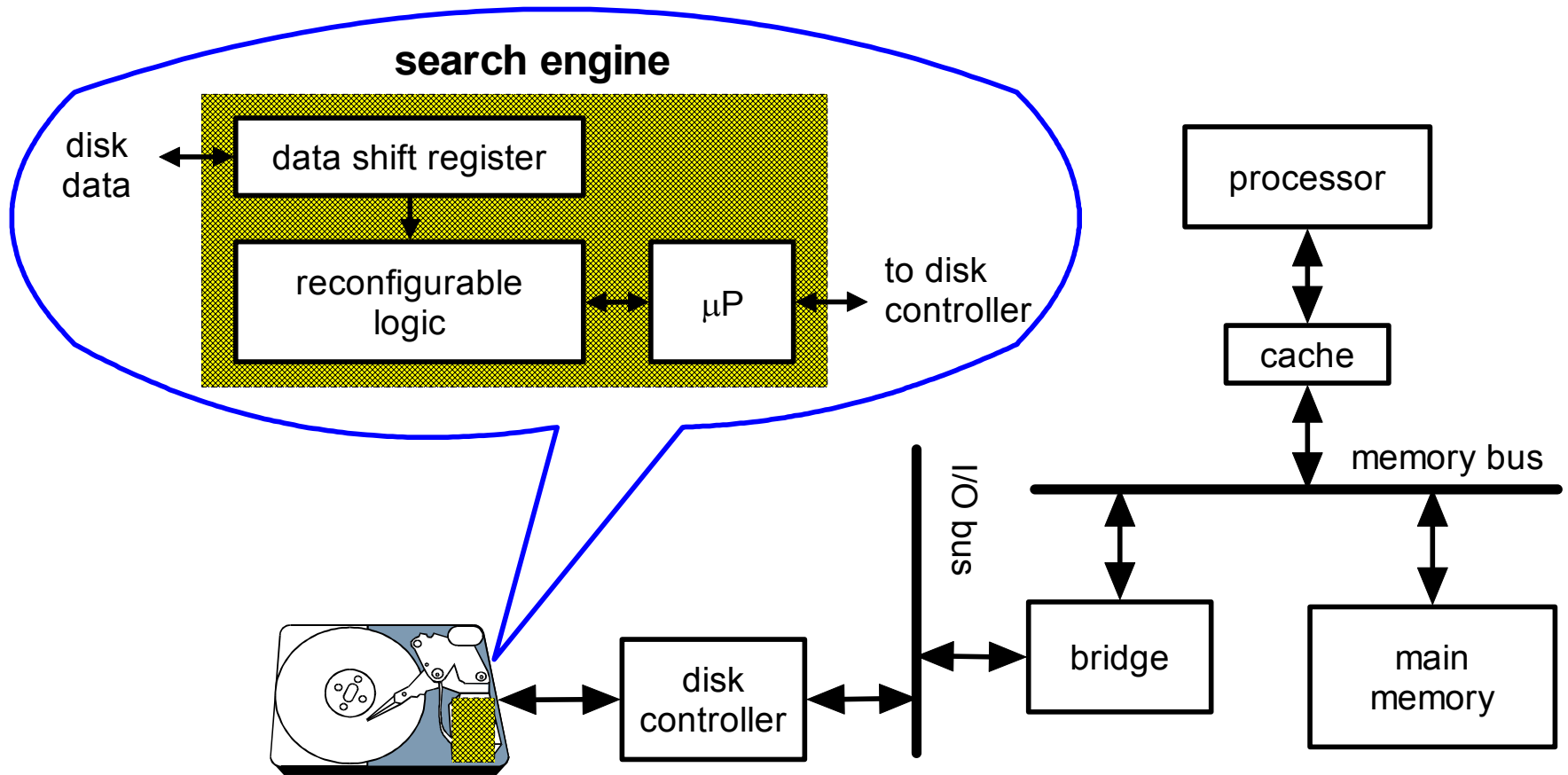
# Streaming Approach



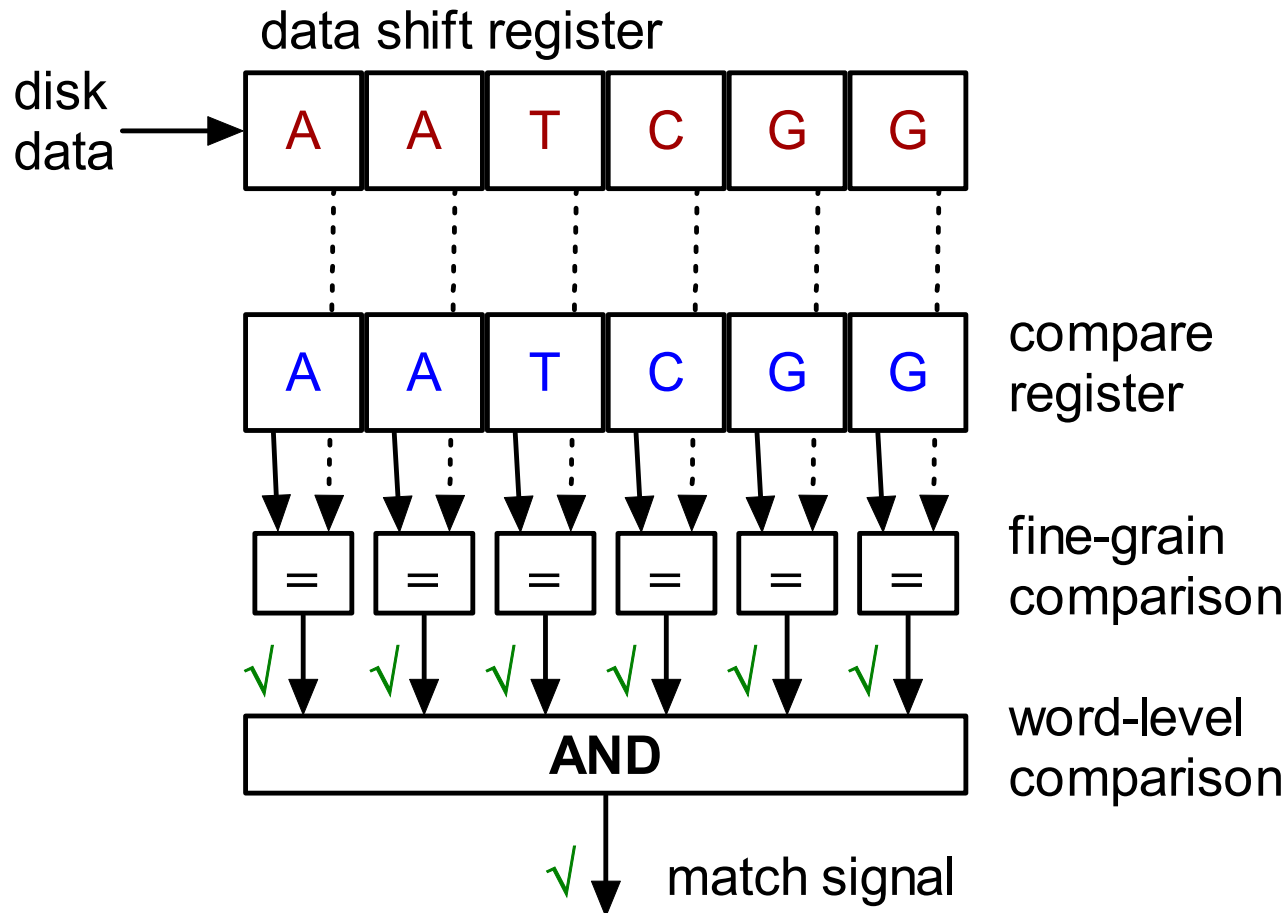
# Streaming Approach



# Search Engine in Context



# Reconfigurable Hardware for Text Searches



# Sources of Performance Gains

---

1. Disk Search Parallelism: Each engine searches in parallel across a disk or disk surface
2. System Parallelism: Searching is off-loaded to search engines and main processor can perform other tasks
3. Reduced data movement overhead: Disk data moves principally to search engine, not successively over system bus, memory bus, to cache, etc.
4. Hardware logic for searching: Searching, matching, and query operations are performed on streaming data in hardware rather than in software
5. Specialized hardware logic tailored to queries: Reconfigurable hardware permits matching the query logic to the search engine logic and preserves flexibility



# Technical Status

---

- Prototype operational
- External to an ATA/100 drive
  - →performance is currently disk-limited
  - SCSI-based RAID system under development
- 3 applications functional
  - Exact text search
  - Approximate text search (agrep)
  - Biosequence search (Smith-Waterman)



# Performance

Speedup relative to 1 GHz processor

<b>Application</b>	<b>Disk-limited speedup</b>	<b>Logic-limited speedup</b>
Exact text search	1.1	14
Approx. text search	12	31
Biosequence search	50	125



# Summary

---

- Fast, inexpensive searches for large and changing databases
- Approximate searches supported
- Up to 100 times faster than standard database searches
- Performance is scalable and uses conventional disk drives
- Data Search Systems, Inc. is actively commercializing the technology

