

First Deliverable: Related Work

- Find and read the most relevant information to your work:
 - Find all relevant code/pseudocode for the algorithms you experiment.
 - Find and install all relevant software systems you evaluate.
 - Find and read previous similar experimentation or evaluation work.
- We expect you to read approximately 3-6 article's (50 pages) worth of best previous works.
- Use the library and internet.
- Write a review of the related work you find. This should be approximately a page (~5000 characters) of prose.
- Format yet to be decided, more after a fortnight.

Second Deliverable: Test Plan

The test plan should contain:

1. A list and possibly a brief description of the algorithms or technologies to be studied. If needed, specify here more accurately possible open issues in the topic description.
2. A description of intended input data or benchmark cases and their origin and justification.
3. If the input data is simulated, list of parameters that describes the data.

Which parameters were chosen as *factors* to be varied in the experiments, and why? What range of values will be assigned to the factors?

What fixed values were chosen for other parameters, and why?

4. In some cases interesting factors (or other parameters) can not yet be stated without preliminary experimentation. If so, express the issue and return elaborate the test plan later, for example in discussions 22.-26.3.2004.
- Example: Searching x in a sorted array

- Algorithms: linear scan, bisection search $\text{mid}_b = \frac{\text{lo} + \text{hi}}{2}$, interpolation search

$$\text{mid}_i = \text{lo} + (\text{hi} - \text{lo}) \frac{x - A[\text{lo}]}{A[\text{hi}] - A[\text{lo}]},$$

combination search

$$\text{mid}_c = C_1 \text{mid}_i + (1 - C_1) \text{mid}_b,$$

but linear if $\text{hi} - \text{lo} < C_2$.

- Input data: N 32-bit integers $A[0] \dots A[N - 1]$ where $A[i] - A[i - 1]$ follows geometric distribution with variance σ^2 . Search for keys uniformly distributed in the range of integers.
- Factors: N, σ^2 . For example, let N be $\lceil 10^d \rceil$ for $d \in \{0.5, 0.75, 1, 1.25, \dots, 6.75, 7\}$.
(Concern: will the 32-bit integers be sufficient if d and σ^2 are large?)
- Other parameters: Use $C_1 \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$.
- Open questions: C_2 can not yet be chosen optimally. It is will be chosen to be the N where bisection search becomes faster than linear scan.
- (More specifications.)

5. For some topics there are few or no parameters, but they do have significant work in benchmark implementation. Give here a description and justification for your benchmark programs.
6. Environments (hardware, OS, languages, their implementations) where the tests are performed.

Exercises in experimental algorithmics will practically always have to use C (or C++ or FORTRAN). Otherwise the implementation details of the language may affect the results too much.

7. (Preliminary) description of the test runs.

“Choose 100 keys into an array, use the `rdtsc` instructions to take time before and after testing searches for those 100 keys.”

8. Describe and justify the statistical treatment of measurements.

“Use 1000 test cases for each factor setting. Their average, 5% lowest and highest percentile results are recorded. Each test case is executed 100 times but, in order to cover for load from other processes, at different times of the day and their median is used.”

9. How do you expect to report your results? Give a plan on the tables/graphs/etc. you plan to use. This may, however, change quite a bit as your experiments progress.

10. Checklist:

Estimate the total duration of your experiments and the amount of page space to present your results. Be prepared to prune.

Conversely, will all interesting cases be covered?

Zipfian Distribution

- Applies to words in human or computer languages, operating system calls, colors in images, etc., and is the basis of many compression approaches.
- George Kingsley Zipf, professor in German at Harvard University in the 30's.
- $P_n \sim 1/n^\alpha$, where P_n is the frequency of occurrence of the n th ranked item.
- $\alpha \approx 1$ yields original Zipfian distribution.
- $\alpha = 0$ yields uniform distribution.

Caveat Caches

- Caches improve memory reads significantly in case of
 - Temporal locality: the data has already been read to the cache and hasn't been replaced since.
 - Spatial locality: the data is “near” another data which is cached.
- Another process can sporadically disturb temporal locality.
- Subtle details, such as internals of malloc and free or in programming language implementation, affect spatial locality.
⇒ Should you implement your own memory management in order to factor out such unknowns?

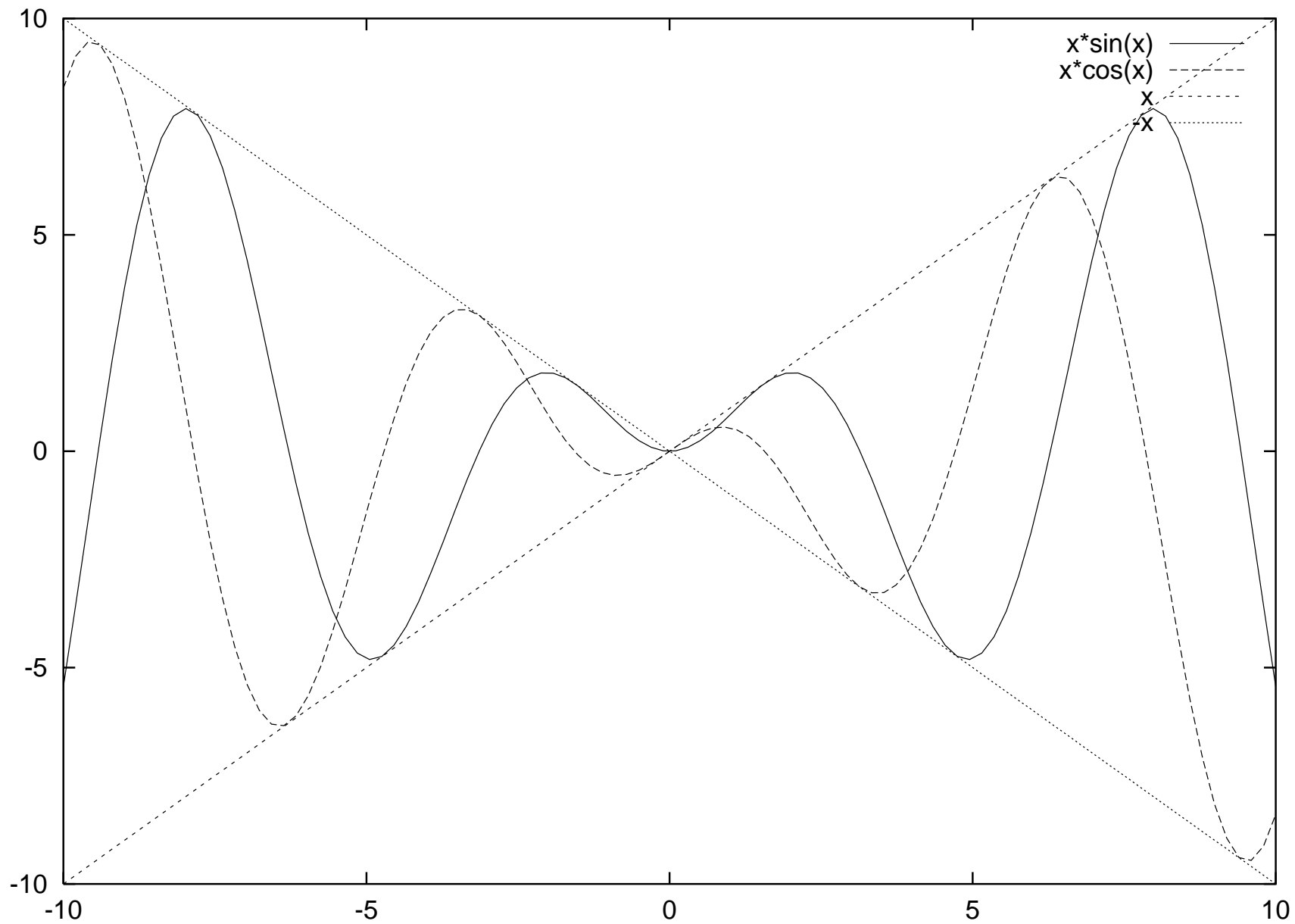
- Cost for a hit in

L1 Cache	$< 1ns$
L2 Cache (on-chip)	$3 - 10ns$
Main Memory	$100ns$
Main Memory + TLB miss	$200ns$
Disk	$\approx 10ms$

- How does this affect your tests???

Gnuplot

- Free, command-driven, interactive, function and data plotting program.
- <http://www.gnuplot.info/>
- <http://t16web.lanl.gov/Kawano/gnuplot/index-e.html>
- Interactive hierarchial help system:
gnuplot> help
- First example:
gnuplot> plot [x=-10:10] x*sin(x), x*cos(x), x, -x
- Replot the same into a postscript file foo.eps:
gnuplot> set output "ex-1.eps"
gnuplot> set terminal postscript
gnuplot> replot



- Second example, configuring the plotting in various ways:

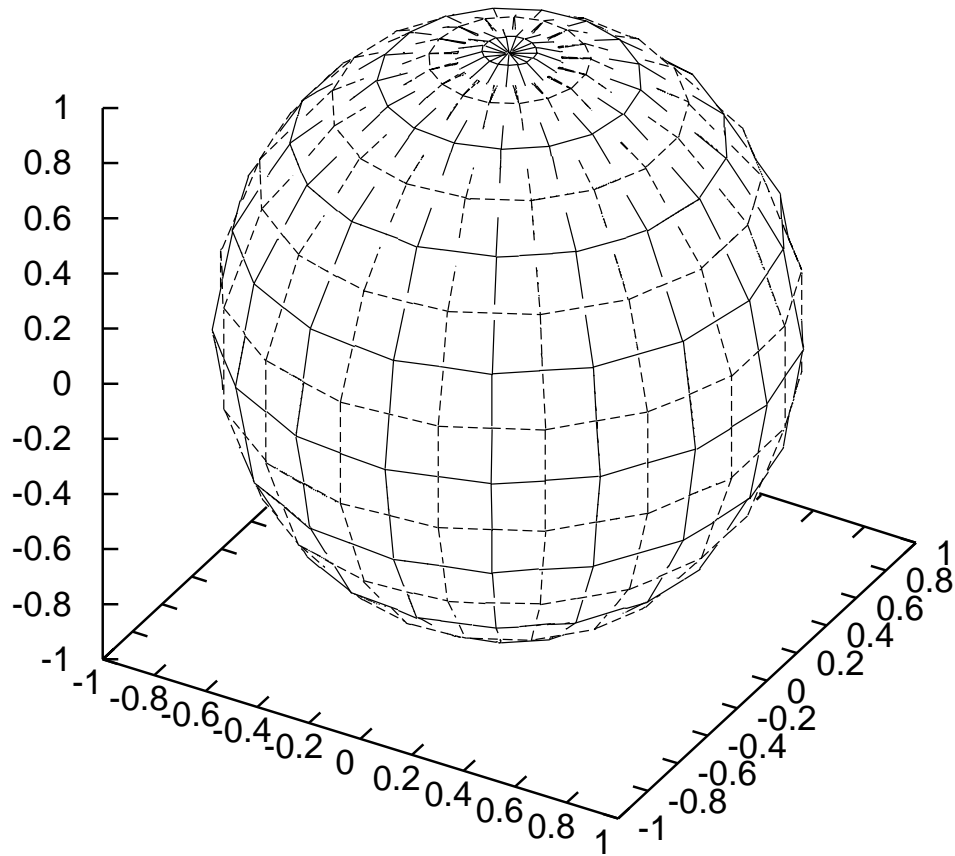
```
set logscale x 2
set logscale y
set xlabel 'key size [bytes]'
set xrange [8:4096]
set ylabel 'Probability of need to read the full key'
set data style lines
set key 2200,0.005
plot "ex-2.in" using 1:8 t 'key caching twice better', \
      "ex-2.in" using 1:7 t 'key caching 50% better', \
      ..
      "ex-2.in" using 1:2 t 'key inclusion twice better'
```


- Other useful features:
 - plot ...with errorbars for showing confidence intervals
 - fit for fitting/smoothing

- Last demo:

```
set parametric
set angle degree
set urange [0:360]
set vrange [0:360]
set isosample 18,18
set ticslevel 0
set size 0.65,1.0
set hidden
splot cos(u)*cos(v), sin(u)*cos(v), sin(v)
```

$\cos(u)\cos(v), \sin(u)\cos(v), \sin(v)$ ———



Next

- Next Week: L^AT_EX and structure of the report.
- Fortnight: Mechanisms for returning of deliverables