

# Discussion on WG 2.1 Challenge Problems

Lambert Meertens

# Current situation

- We have been rather successful in developing theory and methods for the crisp derivation of solutions for certain kinds of problems
- For many other kinds of problems we aren't quite as far
- Use common set of *challenge problems* for focus on pushing the envelope

# Example problem: Web cache

We have two basic sorts: **KEY** and **RESULT**. There is a constant of sort **RESULT**, called 'Empty'. A *repository* is a binary relation, being a subset of **KEY** **RESULT**, where the key field is unique.

A repository server receives requests from a client and returns replies. A request has the form  $(k)$  with  $k$ : **KEY**.

If there is no pair  $(k, r)$  in the repository, the server replies with **(Empty)**. If there is a pair  $(k, r)$  in the repository, the reply is  $(r)$ .

# Cache problem (2)

**We want to put a cache between the client and the server. To the client, the cache component behaves exactly like a server. To the server, the cache is indistinguishable from a client. The idea is that the cache maintains its own repository, initially empty. Upon receiving a request ( $k$ ) from the client, if there is no pair ( $k, r$ ) in the cache repository, the cache sends the request ( $k$ ) to the server and relays the reply ( $r$ ) received back to the client. Additionally, the pair ( $k, r$ ) is stored in the cache's repository. If there is a pair ( $k, r$ ) in the cache repository, the cache replies to the client with ( $r$ ).**

# Cache problem (3)

**Derive the cache as sketched above using the requirement that from the point of view of the client there is no difference between client  $\leftrightarrow$  server and client  $\leftrightarrow$  cache  $\leftrightarrow$  server.**

- A more elaborate version could involve time stamps

# Other possible challenges

- In-situ quicksort
- Unification
- Approximate TSP
- Distributed dynamic maximum
- Incremental rescheduling
- Targeting FPGA
- Mixed systems (codesign, hybrid)

# Selection criteria

- Is the problem non-contrived (ideally paradigmatic for actual problems)?
- Does the problem have a clear and crisp formulation?
- Is it doable?
- Does this problem represent some new aspect?

# Operation

- Collect proposed problems
- Small editorial committee ensures good problem mix and clear formulations
- Problems published on web site
- Major part of meetings set aside to presentations of solutions
- Best solutions published on web site



# Solution criteria

- Is the solution clear, concise, precise?
- Can it be taught?
- Is it effective (directly implementable)?
- Is the solution method generalizable?
- Does the solution improve on earlier solutions?
- Or does it bring a new viewpoint?

# Web pages

- The CP web pages are ‘dynamic’: solutions may be improved or merged as time goes on; result is product of the Group while reasonably acknowledging individual contributions
- Open to ‘external’ input

# Questions

- Is this a good idea?
- Are there enough ‘good’ problems?
- Will people actually work on the problems?
- Will this lead to new or improved theory and methods?
- Can we form an editorial committee?