**Filippo**

* All wrapped up in a reasonable way
* When is the simulator introduced? Slide “Average simulation results, 64” seems to mix results from real machine and the simulator.
* How is the instantaneous power consumption measured?
  + Interference between the power deamon and the application?
  + The additional overhead introduced by the deamon is not quantified.
* Task creation overhead?
  + Say which application is represented in Slide 69?
* GTime
  + Does GTime depend on N? Slide 74.
  + R should be a function of N
  + Nested parallelism? Just divide r by the levels of nested parallelism that you may have.
  + Value of r = 16x on slide 78. Where does it come from?
* Last contribution on the PhD document
  + Why isn’t it present in the presentation?

**Daniel Jiménez**

* Comment the last contribution just at the very high level
* Task creation (slide 69)
  + Task finalization is not directly accounted
  + How about lock contention?
  + Task creation in the thesis is not exactly the same as in nanos
  + Picos considers task creation, dependence and scheduling as task creation
* Slide 36 is a bit misleading in terms of what are ready tasks and what are not
* Combining CATS and TaskGenX?
  + Task Priorization is accelerated or not?
* Cost of migrating tasks from little to big?
  + **Possible answer: There is no migration from little to big. Cores get tasks from ready queues but tasks do not migrate from big to little or the opposite.**
* Concept of critical tasks. When using CATS, tasks might not be critical?
  + **Possible answer: Both CATS and CPATH estimate which tasks are critical. It is impossible to figure out the real critical path on execution time.**
* More details on canneal?

**Xavier Martorell**

* Slide number 8
  + Example code is not the best one.
  + Unify the code examples.
* Plot of Cholesky’s TDG using CATS and CPATH.
  + How different are they? Are they the same in the case of Cholesky?
* CPATH
  + Slide 64, why is CPATH that bad? Additional overhead.
* Comment on Task execution time differences across different task types.
  + Is there a relationship between these differences and CPATH efficiency?
* Combine priorities given by the user with dynamically computed ones.
* TaskGenX
  + Nomenclature is slightly change from DAS THREAD.
  + Finalization is mixed with dependence analysis?
    - Done is different from delete?
  + Slide 68: change the order of runtime activity
* Related work: **Add DAS THREAD plus PICOS**.
* How would the HW accelerator would look like:
  + Which exact part would you accelerate and how?
  + Specific memory region to avoid the impact of memory misses?
    - Using a scratchpad?
  + Add a high-level description of the HW.
    - Graph analysis HW able to traverse and update the TDG data structure.