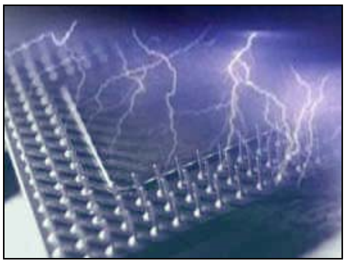
Vatios 1.0

User Manual



Juan Antonio Victorio Ferrer

University of Zaragoza

Table of Contents

1. Description

Vatios is the Spanish word for "*Watts*". In this context, Vatios is a set of tools for computer architects that was developed as a part of a project at the Computer Architecture Group (gaZ) at the University of Zaragoza http://webdiis.unizar.es/gaz/. Our start point was Wattch, developed by David Brooks http://www.eecs.harvard.edu/~dbrooks/

Wattch is a tool based on SimpleScalar that simulates the execution of a program or an execution-trace and, at the end of the simulation, gives power and energy consumption of the different units that make up the processor.

Vatios is a set of tools, written in C that can be used to get power and energy consumption values from super-scalar processors. It's based on Wattch and not only provides the same functionality as, but also gives more flexibility and new interesting features.

Vatios is compounded of `sim-vatios`, which is the simulator similar to Wattch, and `power-vatios`, which is a tool to calculate power/consumption from previous simulations.

2. Differences

Wattch has been a great development because a computer architect can take design-decisions based on power and energy predictions made by Wattch at early development stages. With Wattch, you don't need a detailed circuit level description of the processor to get the energy consumption prediction.

Wattch's main problem is its lack of flexibility. All the technology parameters used during the simulation are directly written in the code, and the user is not able to modify them without re-compiling the tool. We have written the Vatios tool set in order to avoid those problems and create a more flexible and generic tool.

The Vatios' main simulator `sim-vatios` uses a more efficient algorithm that saves some floating-point operations every cycle of the simulation, calculates the Energy prediction as well and allows the user to dump in a file use/access distribution statistics of every processor's unit, and later, reuse those statistics to generate new predictions for different tech points or power models.

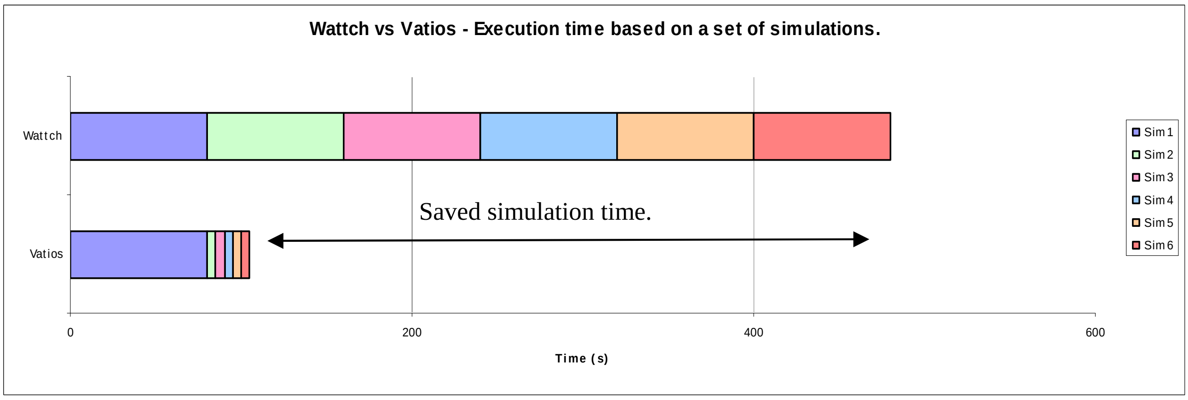
Instead of counting number of access and at the end of the cycle, generating the power/energy prediction every simulation's cycle, `sim-vatios` counts the number of accesses and generates use/access distributions statistics. With those statistics `sim-vatios` calculates the energy/power predictions at the end of the simulation, not at the end of the cycle. Moreover, this allows us to dump those statistics to a file and reuse them with `power-vatios`.

We have developed a program named `power-vatios` that with a dumped file, that includes the unit's architectural parameters and the usage statistics, can generate the same power and energy predictions that Wattch would generate without re-compiling the simulator or re-simulating the temporal execution.

We have also updated the CACTI 1.0 tool to the CACTI 4.2 tool.

3. Simulation Speedup

The main advantage is that this program allows you to change different technology parameters as well as the implementation of the power models (the algorithm used to calculate the peak power of a unit based on architectural parameters) and generate new energy/power predictions without re-simulating the program or execution trace if the unit's timing and use don't change.



Let's consider this example: We have to simulate an execution trace, with 2 different power models and each of these, in 3 different technology points.

With Wattch we have to simulate the execution:

3 technology points * 2 different power models = 6 times.

Moreover, we are not considering the compilation time, because with Wattch, each time that we are changing the technology point, we have to compile the simulator.

With Vatios we have to simulate one full temporal execution with `sim-vatios`. Then, since we have the use/access statistical distribution in a dumped file, and assuming that the timing of the simulation doesn't change if we move to another technology point, we have to execute `power-vatios` 5 extra times to calculate the energy/power prediction without re-simulating the full temporal execution. So, we have saved 5 unnecessary full temporal simulations, what depending of the length of the trace, can mean several hours.

4. Requisites

We have developed the Vatios tool-set in C under Linux on a x86 machine, and we don't use any special library, so the requisites are:

The Vatios tar.gz file
The GCC compiler
The Make tool
The standard library

You can download the Vatios tar.gz file from http://webdiis.unizar.es/gaz/vatios/ and in the case that you don't have any of the other tools, you should install or tell the administrator to install them.

Vatios is based on SimpleScalar, so there would be no difficult to compile it on any system that can use SimpleScalar.

5. Compilation

If you meet the requisites, you should extract the tar.gz file to your selected directory and then access the directory and type `make` in the command like. Everything else should be done automatically. This process will generate two executable files `sim-vatios` and `power-vatios`.

6. Use `sim-vatios`

First of all, you should simulate the executable or the execution trace. You have to do this the same way you would do it with SimpleScalar or with Wattch. This time the name of the executable will be `sim-vatios`. But there are some parameters that you have to take care of.

6.1. The `–dumpconfig` parameter

The first parameter you should take care of is `-dumpconfig myDumpedFile`. This option already appears in SimpleScalar, but, if you are using it with `sim-vatios`, it will dump not only the config (architectural parameters used by the simulation like cache sizes, pipe-stages widths, etc…), but also the statistical information (use/activity of ech unit) needed to calculate the energy/power prediction.

If you don't specify the `–dumpconfig` option, the execution of `sim-vatios` will be similar to a normal Wattch execution.

6.2. The `–tech` parameter

To calculate the energy/power prediction, the program needs the technology point you are simulating at, and the energy model you are using for each unit.

Therefore, we have defined the parameter `-tech` which defines the technology point. Available tech points are: 800, 400, 350, 250, 180 and 100 in nm. If you specify an unavailable tech point, the simulator will report an error.

6.3. The `–freq` parameter

Wattch has been developed simulating a 600Mhz CPU. All the power is calculated at this clock-speed. We have programmed Vatios considering the CPU speed as a parameter. By default, if you don't specify this parameter, it will be 600Mhz, as in Wattch, but you can specify the speed as an integer number in Mhz. For example, for a 2Ghz CPU you have to add this option: `-freq 2000`

6.4. Power Models

An energy model is an algorithm that calculates the peak power of a unit. Wattch defines only one power model for each unit. With Vatios there are several power models for each unit, and they are identified by numbers. By default, this version of Vatios runs with the same power models as Wattch (power model 1)

6.4.1. The power model #0.

but we have added the 0 power model, that means that this unit consumes 0 power/energy. You can use the 0 power model in the case that you want to ignore some units in your predictions. Units modelled with the 0 power model will consume no energy, and therefore, they will have no influence in the total energy/power predictions.

To specify the power model you are going to use for a unit (only if it's not the default), you have to invoke `sim-vatios` with and option that looks like:

```
-unitName:model powerModelNumber
```

For example:

```
-rename:model 0 -bpred:model 0
```

6.4.2. The power model #2 (User defined peak power of the unit).

In Vatios, if you want, you can't directly specify the peak power of a Unit. This can be done using the power model 2. In this case you will have to add an additional parameter (the power this unit consumes).

As an example:

```
-rename:model 2 –rename:power 2.5
```

This line means that want to directly specify the peak power for the rename unit (model 2) and that this unit consumes 2.5 Watts.

6.4.3. The power model #3

In sim-vatios, we have updated the CACTI 1.0 tool to the CACTI 4.2 tool. This version of CACTI is able to calculate not only the subbanking, but also the power of a cache-like unit.

We have added a new power model (model 3) that uses CACTI 4.2 to calculate the peak power of the Instruction Cache (and Intruction TLB), Data Cache (and Data TLB), Level 2 Data Cache and Regfile.

6.5. Names of the Units

The names of the units are specified at `vatios_strings.h` and by default are these:

```
-rename
-bpred
-regfile
-icache
-dcache
-dcache2
-ialu
-falu
-resultbus
-window_preg
-window_selection
-window_wakeup
-lsq_wakeup
-lsq_preg
```

By default, if you don’t specify the power model, the power model 1 will be used (the same as Wattch)

If you need to simulate with other tech points, or you need to add new power models to Vatios, please refer to the “Vatios Modification Manual”.

7. Use `power-vatios`

Imagine that you want to simulate the energy consumption of one processor in several technologies or using several power models. At first glance, it seems unnecessary to simulate the execution every time, if the timing is going to be the same.

If you have already simulated one execution and you have dumped a file with the use/access information, you can use this file to calculate power/energy predictions without re-simulating the entire execution again. Starting from a use/activity file dumped with `sim-vatios`, you only have to invoke `power-vatios` with the appropriate arguments.

The first argument you have to specify is the name of the dumped file. This time with the option `-config myDumpedFile`

You also have to specify the tech point, frequency and the power models. This is done the same way as with `sim-vatios`.

8. Results

If you are not familiar with Wattch, it displays 4 kinds of results, depending on the clock gating that simulates. It displays the peak power of the units as well.

The first one (NCC) simulates no clock gating, the second one (CC1) simulates simple clock gating, the third one (CC2) simulates aggressive ideal clock gating, ant the last one (CC3) simulates aggressive non-ideal clock gating. If you need a more detailed explanation of these results, please read the original Wattch paper.

With Vatios we calculate the same results as with Wattch, the only difference is that we cannot detect the cycle when the energy consumption is maximum. We have also changed the way that we calculate the clock energy. It's important to note that results with Vatios will be different from results with Wattch, since the Cacti tool has been updated to version 4.2 and this tool is used to get the results.

With Wattch:

Clock energy CCX = Clock energy NCC * Total Energy CCX / Total Energy NCC

Where Total Energy is the energy dissipated by the rest of the units that cycle.

With Vatios, since with our system we cannot calculate the energy in a specific cycle, we use this formula but taking the total energy of the whole simulation.

Here can be seen some examples of the results generated by Vatios:

```
#=========================================#
#VATIOS - Power/energy calculator tool V1.0#
#=========================================#
Rename Accesses Power AF Independent: 0
Rename Accesses Energy NCC: 0
Rename Accesses Energy CC1: 0
Rename Accesses Energy CC2: 0
Rename Accesses Energy CC3: 0
Bpred Accesses Power AF Independent: 18.6886
Bpred Accesses Energy NCC: 973957
Bpred Accesses Energy CC1: 122625
Bpred Accesses Energy CC2: 69007.7
Bpred Accesses Energy CC3: 154168
Regfile Accesses Power AF Independent: 0.677213
Regfile Accesses Power AF Dependent: 7.49981
Regfile Accesses Energy NCC: 426145
Regfile Accesses Energy CC1: 181133
Regfile Accesses Energy CC2: 43714.1
Regfile Accesses Energy CC3: 65263.8
```

9. Examples

In this section we are going to comment a pair of invocations to Vatios in order to help understanding how Vatios works.

9.1. Example 1

With this invocation

```
./sim-vatios –tech 400 tests-alpha/eio/test-math.eio
```

we are simulating the test-math trace, without dumping the access statistics to a file, calculating and displaying the energy/power predictions at the end of the simulation, the prediction is calculated for a tech point of 400 nm , with the same power models as Wattch (default power models).

9.2. Example 2

With this invocation

```
./sim-vatios -dumpconfig myFile.txt tests-alpha/eio/test-
math.eio
```

We simulate the test-math trace, dumping the access statistics to “myFile.txt”, and calculating the energy/power prediction for a tech point of 350 nm (default tech point) and the same power models as Wattch (default power models).

With this invocation

```
./power-vatios -config myFile.txt
```

we are calculating the power/energy predictions for the architecture and access statistics saved “in myFile.txt” for a tech point of 350 nm (default tech point) and the same power models as Wattch (default power models).

With this invocation

```
./power-vatios -config myFile.txt -tech 400
```

We are specifying that the tech point we want to simulate is 400, not the default 350.

With this invocation

```
./power-vatios -config myFile.txt -rename:model 0
```

We are specifying that we want to ignore (power model 0) the rename energy/power in this simulation.

10. Further development

We hope to further develop Vatios, add new power models and technology points. However, as it’s free software, you can improve it as you want. You can also send your request by email.