Hardest-to-Round Cases – Part 2

Vincent LEFÈVRE

AriC, INRIA Grenoble – Rhône-Alpes / LIP, ENS-Lyon

Journées TaMaDi, Lyon, 2013-10-08

Outline

- Hardest-to-Round Cases in binary64 (Double Precision)
- Functions x^n
- Average Computation Time

Hardest-to-Round Cases in binary64 (Double Precision)

Let's recall...

- Floating-point system in radix 2.
- Double precision ($p = 53$).
- **No subnormals.**
  In input, the exponent range will be extended to include subnormals.
- Exact cases are regarded as hard-to-round cases (stored in the database).
  Exactness is checked by `readres` with GNU MPFR and these cases are not output.
- Algorithm used: L-algorithm (first step).

Hardest-to-Round Cases in binary64 (Double Precision) [2]

After 13 812 778 CPU core hours ($\approx$ 1576 years) for the first step, in summary:

- e^x, 2^x, 10^x, sinh, cosh, $\sin(2\pi x)$, $\cos(2\pi x)$, $\tan(2\pi x)$;
- x^n for $3 \leq n \leq 5188$ and $-180 \leq n \leq -2$;
- sin, cos, tan between $-\pi/2$ and $\pi/2$;
- the corresponding inverse functions.

Hardest-to-Round Cases in binary64 (Double Precision) [3]

The following results are presented in a different way from 2010, separating rounding to nearest and directed rounding.

Only the **hardest-to-round case** in the considered domain is given.

Filtering done manually. Let's hope there are no errors...

Format of the results: `function(hexForm) = hexForm:rf[k]xxxx`
where:

- `hexForm` denotes a binary64 number in the ISO C99 / IEEE 754-2008 hexadecimal format (here, we chose $\pm$`1.hhhhhhhhhhhhhPe`, where `h` is a hexadecimal digit and `e` is the binary exponent written in decimal);
- `r` is the rounding bit;
- `f` is the following bit;
- `k` is the number of times this bit is repeated;
- `xxxx` are the next 4 bits of the exact result.

Functions exp and log

Function exp:

- Rounding to nearest, whole domain:
  `exp(-1.12D31A20FB38BP+5) = 1.5B0BF3244820AP-50:01[58]0010`
- Directed rounding, in $(-\infty, -2^{-37}] \cup [2^{-36}, +\infty)$:
  `exp(-1.ED318EFB627EAP-27) = 1.FFFFFF84B39C4P-1:11[59]0001`
- Directed rounding, in $[-2^{-37}, 2^{-36}]$: (special)
  `exp(1.FFFFFFFFFFFFP-53) = 1.0000000000000P0:11[104]0101`

Function log:

- Rounding to nearest, whole domain:
  `log(1.FD15DAA6CE332P+732) = 1.FC12387D06329P+8:10[61]1111`
- Directed rounding, whole domain:
  `log(1.62A88613629B6P+678) = 1.D6479EBA7C971P+8:00[64]1110`

Functions expm1 and log1p

Function expm1:

- Rounding to nearest, in $(-\infty, -2^{-51}] \cup [2^{-51}, +\infty)$:
  `expm1(1.274BBF1EFB1A2P-10) = 1.2776572C25129P-10:10[58]1000`
- Directed rounding, in $(-\infty, -2^{-35}] \cup [2^{-35}, +\infty)$:
  (except the cases whose image is very close to -1)
  `expm1(1.83D4BCDEBB3F4P+2) = 1.AB50B409C8AEEP+8:00[57]1000`
- Directed rounding, in $[-2^{-35}, -2^{-51}] \cup [-2^{-51}, -2^{-35}]$: (special)
  `expm1(-1.800000000003P-49) = -1.7FFFFFFFFFFAP-49:00[96]1000`

Function log1p:

- Rounding to nearest, in $(-1, -2^{-37}] \cup [2^{-37}, +\infty)$:
  `log1p(1.FD15DAA6CE332P+732) = 1.FC12387D06329P+8:10[61]1111`
- Rounding to nearest, in $[-2^{-37}, -2^{-51}] \cup [2^{-51}, 2^{-37}]$: (special)
  `log1p(1.800000000003P-50) = 1.7FFFFFFFFFFEP-50:10[99]1000`
- Directed rounding, in $(-1, -2^{-37}] \cup [2^{-37}, +\infty)$:
  `log1p(1.62A88613629B6P+678) = 1.D6479EBA7C971P+8:00[64]1110`
- Directed rounding, in $[-2^{-37}, -2^{-51}] \cup [2^{-51}, 2^{-37}]$: (special)
  `log1p(1.800000000006P-49) = 1.7FFFFFFFFFFDP-49:00[96]1000`

Functions sinh and asinh

Function sinh:

- Rounding to nearest, in $[2^{-25}, +\infty)$:
  `sinh(1.897374D74DE2AP-13) = 1.897374FE073E1P-13:10[56]1011`
- Directed rounding, in $[2^{-16}, +\infty)$:
  `sinh(1.E07E71BFCF06FP+5) = 1.91EC4412C344FP+85:00[55]1000`
- Directed rounding, in $[2^{-25}, 2^{-16}]$: (special)
  `sinh(1.DFFFFFFFFFE3EP-20) = 1.E000000000FD1P-20:11[72]0001`

Function asinh:

- Rounding to nearest, in $[2^{-25}, +\infty)$:
  `asinh(1.FD15DAA6CE332P+731) = 1.FC12387D06329P+8:10[61]1111`
- Directed rounding, in $[2^{-18}, +\infty)$:
  `asinh(1.62A88613629B6P+677) = 1.D6479EBA7C971P+8:00[64]1110`
- Directed rounding, in $[2^{-25}, 2^{-18}]$: (special)
  `asinh(1.E000000000FD2P-20) = 1.DFFFFFFFFFE3EP-20:00[72]1110`

Functions cosh and acosh

Function cosh:

- Rounding to nearest, in $[2^{-25}, +\infty)$:
  `cosh(1.EA5F2F2E4B0C5P+1) = 1.710DB0CD0FED5P+4:10[57]1110`
- Directed rounding, in $[2^{-16}, +\infty)$:
  `cosh(1.E07E71BFCF06FP+5) = 1.91EC4412C344FP+85:00[55]1000`
- Directed rounding, in $[2^{-25}, 2^{-16}]$: (special)
  `cosh(1.7FFFFFFFFFF7P-23) = 1.000000000047P0:11[89]0010`

Function acosh:

- Rounding to nearest, in $[1, +\infty)$:
  `acosh(1.297DE35D02E90P+13) = 1.3B562D2651A5DP+3:01[61]0001`
- Directed rounding, in $[1, +\infty)$:
  `acosh(1.62A88613629B6P+677) = 1.D6479EBA7C971P+8:00[64]1110`

Functions sin and asin

Function sin:

- Rounding to nearest, in $[2^{-25}, (1 + 4675/2^{13}) \cdot 2^1]$:
  `sin(1.598BAE9E632F6P-7) = 1.598A0AEA48996P-7:01[59]0000`
- Directed rounding, in $[2^{-18}, (1 + 4675/2^{13}) \cdot 2^1]$:
  `sin(1.FE767739D0F6DP-2) = 1.E9950730C4695P-2:11[65]0000`
- Directed rounding, in $[2^{-25}, 2^{-18}]$: (special)
  `sin(1.E0000000001C2P-20) = 1.DFFFFFFFFF02EP-20:00[72]1110`

Function asin:

- Rounding to nearest, in $[2^{-25}, 1]$:
  `asin(1.C373FF4AAD79BP-14) = 1.C373FF594D65AP-14:10[57]1010`
- Directed rounding, in $[2^{-18}, 1]$:
  `asin(1.E9950730C4696P-2) = 1.FE767739D0F6DP-2:00[64]1000`
- Directed rounding, in $[2^{-25}, 2^{-18}]$: (special)
  `asin(1.DFFFFFFFFF02EP-20) = 1.E0000000001C1P-20:11[72]0001`

Functions cos and acos

Function cos:

- Rounding to nearest, in $[0, \mathrm{acos}(2^{-26})] \cup [\mathrm{acos}(-2^{-27}), 4]$:
  `cos(1.34EC2F9FC9C00P+1) = -1.7E2A5C30E1D6DP-1:01[58]0110`
- Directed rounding, in $[2^{-17}, \mathrm{acos}(2^{-26})] \cup [\mathrm{acos}(-2^{-27}), 4]$:
  `cos(1.06B505550E6B2P-9) = 1.FFFFBC9A3FBFEP-1:00[58]1100`
- Directed rounding, in $[0, 2^{-17}]$: (special)
  `cos(1.800000000009P-23) = 1.FFFFFFFFFFF70P-1:00[88]1101`

Function acos:

- Rounding to nearest, in $[-1, -2^{-27}] \cup [2^{-26}, 1]$:
  `acos(1.53EA6C7255E88P-4) = 1.7CDACB6BBE707P0:01[57]0101`
- Directed rounding, in $[-1, -2^{-27}] \cup [2^{-26}, 1]$:
  `acos(1.FD737BE914578P-11) = 1.91E006D41D8D8P0:11[62]0010`

Functions tan and atan

Function tan:

- Rounding to nearest, in $[2^{-18}, \pi/2]$:
  `tan(1.50486B2F87014P-5) = 1.5078CEBFF9C72P-5:10[57]1001`
- Rounding to nearest, in $[2^{-25}, 2^{-18}]$: (special)
  `tan(1.DFFFFFFFFFF1FP-22) = 1.E000000000151P-22:01[78]0100`
- Directed rounding, in $[2^{-17}, \pi/2]$:
  `tan(1.A33F32AC5CEB5P-3) = 1.A933FE176B375P-3:00[55]1010`
- Directed rounding, in $[2^{-25}, 2^{-17}]$: (special)
  `tan(1.DFFFFFFFFFC7CP-21) = 1.E000000000545P-21:11[72]0100`

Function atan:

- Rounding to nearest, in $(2^{-25}, +\infty)$:
  `atan(1.6298B5896ED3CP+1) = 1.3970E827504C6P0:10[63]1101`
- Directed rounding, in $(2^{-18}, +\infty)$:
  `atan(1.EB19A7B5C3292P+29) = 1.921FB540173D6P0:11[59]0011`
- Directed rounding, in $[2^{-25}, 2^{-18}]$: (special)
  `atan(1.E000000000546P-21) = 1.DFFFFFFFFFC7CP-21:00[72]1011`

Functions sin2pi and asin2pi

Warning! Results not guaranteed by readres.

Function sin2pi:

- Rounding to nearest, in $[2^{-58}, 1/2]$:
  sin2pi(1.F339AB57731D3P-51) = 1.88173243FB0F4P-48:01[56]0010
- Directed rounding, in $[2^{-58}, 1/2]$:
  sin2pi(1.BC03DF34E902CP-55) = 1.5CBA89AF1F855P-52:00[58]1101

Function asin2pi:

- Rounding to nearest, in $[2^{-57}\pi, 1]$:
  asin2pi(1.7718543A5606AP-29) = 1.DD95F913D2D22P-32:10[58]1011
- Directed rounding, in $[2^{-57}\pi, 1]$:
  asin2pi(1.5CBA89AF1F855P-52) = 1.BC03DF34E902BP-55:11[57]0111

Functions cos2pi and acos2pi

Warning! Results not guaranteed by `readres`.

Function `cos2pi`:

- Rounding to nearest, in $[0, 1/2]$:
  `cos2pi(1.8242846E3D0AFP-25) = 1.FFFFFFFFFFE98P-1:01[57]0101`
- Directed rounding, in $[0, 1/2]$:
  `cos2pi(1.B17C08C8AB938P-14) = 1.FFFFF8ECE1969P-1:00[55]1110`

Function `acos2pi`:

- Rounding to nearest, in $[-1, 1]$:
  `acos2pi( 1.6C6CBC45DC8DEP-49) = 1.FFFFFFFFFFFF1P-3:01[61]0001`
  `acos2pi(-1.6C6CBC45DC8DEP-48) = 1.000000000000EP-2:10[61]1110`
- Directed rounding, in $[-1, 1]$:
  `acos2pi( 1.6C6CBC45DC8DEP-48) = 1.FFFFFFFFFFFE2P-3:11[60]0001`
  `acos2pi(-1.6C6CBC45DC8DEP-47) = 1.00000000001DP-2:00[60]1110`

Functions tan2pi and atan2pi

Warning! Results not guaranteed by `readres`.

Function `tan2pi`:

- Rounding to nearest, in $[2^{-58}, 1/4]$:
  `tan2pi(1.9E2371E233D1BP-35) = 1.45437A2EBE656P-32:10[56]1100`
- Directed rounding, in $[2^{-58}, 1/4]$:
  `tan2pi(1.AC84C88F979A2P-55) = 1.508ECB38F52F9P-52:00[56]1000`

Function `atan2pi`:

- Rounding to nearest, whole domain:
  `atan2pi(1.E1A235BAB7461P+43) = 1.FFFFFFFFFFEA5P-3:10[59]1000`
- Directed rounding, whole domain:
  `atan2pi(1.E1A235BAB7461P+42) = 1.FFFFFFFFFFD4BP-3:00[58]1000`

Functions exp2 and log2

Function exp2:

- Rounding to nearest, in $[-1/2, 1/2]$ ($\rightarrow$ whole domain):
  `exp2(1.E4596526BF94DP-10) = 1.0053FC2EC2B53P0:01[59]0100`
- Directed rounding, in $[-1/2, 1/2]$ ($\rightarrow$ whole domain):
  `exp2(1.BFBBDE44EDFC5P-25) = 1.0000009B2C385P0:00[59]1011`

Function log2:

- Rounding to nearest, in $[1/2, +\infty)$ ($\rightarrow$ whole domain):
  `log2(1.1BA39FF28E3EAP+4) = 1.097767BB6B1E6P+2:10[54]1001`
- Directed rounding, in $[1/2, +\infty)$ ($\rightarrow$ whole domain):
  `log2(1.61555F75885B4P+512) = 1.003B81681E9B9P+9:11[55]0011`

Only HR cases whose exponent is a power of 2 are given.

Functions exp10 and log10

Function exp10:

- Rounding to nearest, whole domain:
  `exp10(1.A83B1CF779890P-26) = 1.000000F434FAAP0:01[60]0101`
- Directed rounding, whole domain:
  `exp10(-1.1416C72A588A6P-1) = 1.27D838F22D09FP-2:11[65]0010`

Function log10:

- Rounding to nearest, whole domain:
  `log10(1.E12D66744FF81P+429) = 1.02D4F53729E44P+7:10[68]1001`
- Directed rounding, whole domain:
  `log10(1.CE41D8FA665FAP+4) = 1.75F49C6AD3BADP0:00[66]1010`

Functions x^n

- For each integer n, only one binade to test, and switching from one n to the next one (and temporarily switching back when needed) is done automatically with the current scripts. The code became stable on 2011-11-25.
- The exponent range is assumed to be unbounded.
- For $|n|$ large, the approximation error by a low-degree polynomial is important.
- Since 2013-07-23 (for $n \geq 4981$), the default internal precision of 300 digits configured for Maple is no longer sufficient.

  This problem has been detected automatically. No wrong results!

  The clients are now started with an option setting the internal precision used by Maple to 350 digits.
- In the following slides, several HR cases may be given for each function class and rounding mode.

HR-Cases of x^n

Function pown for $-180 \leq n \leq -2$:

- Rounding to nearest:
  pown(1.EC658072F2432,-83) = 1.98AEB1A202D6EP-79:01[58]0100
  pown(1.AFC6556E8B8BF,-84) = 1.9272905F02088P-64:01[58]0101
  pown(1.C372354062FD0,-127) = 1.0B3A6186E8373P-104:01[58]0100
- Directed rounding:
  pown(1.A338DAE8C33B7,-166) = 1.D6E21F8ED2049P-119:00[58]1001
  pown(1.44C8DBB1C0114,-179) = 1.74CBC6427CF4DP-62:00[58]1010
  pown(1.527C94D2A1264,-179) = 1.D437BFF7CCB87P-73:11[58]0010

Function rootn (denoted rtn) for $-180 \leq n \leq -2$:

- Rounding to nearest:
  rtn(1.B7BDFD5807F33P-114,-180) = 1.8BE6BE4B400C2:01[71]0001
- Directed rounding:
  rtn(1.D6E21F8ED2049P-119,-166) = 1.A338DAE8C33B7:00[66]1100
  rtn(1.74CBC6427CF4DP-62, -179) = 1.44C8DBB1C0114:00[66]1100
  rtn(1.D437BFF7CCB88P-73, -179) = 1.527C94D2A1263:11[66]0001

HR-Cases of x^n [2]

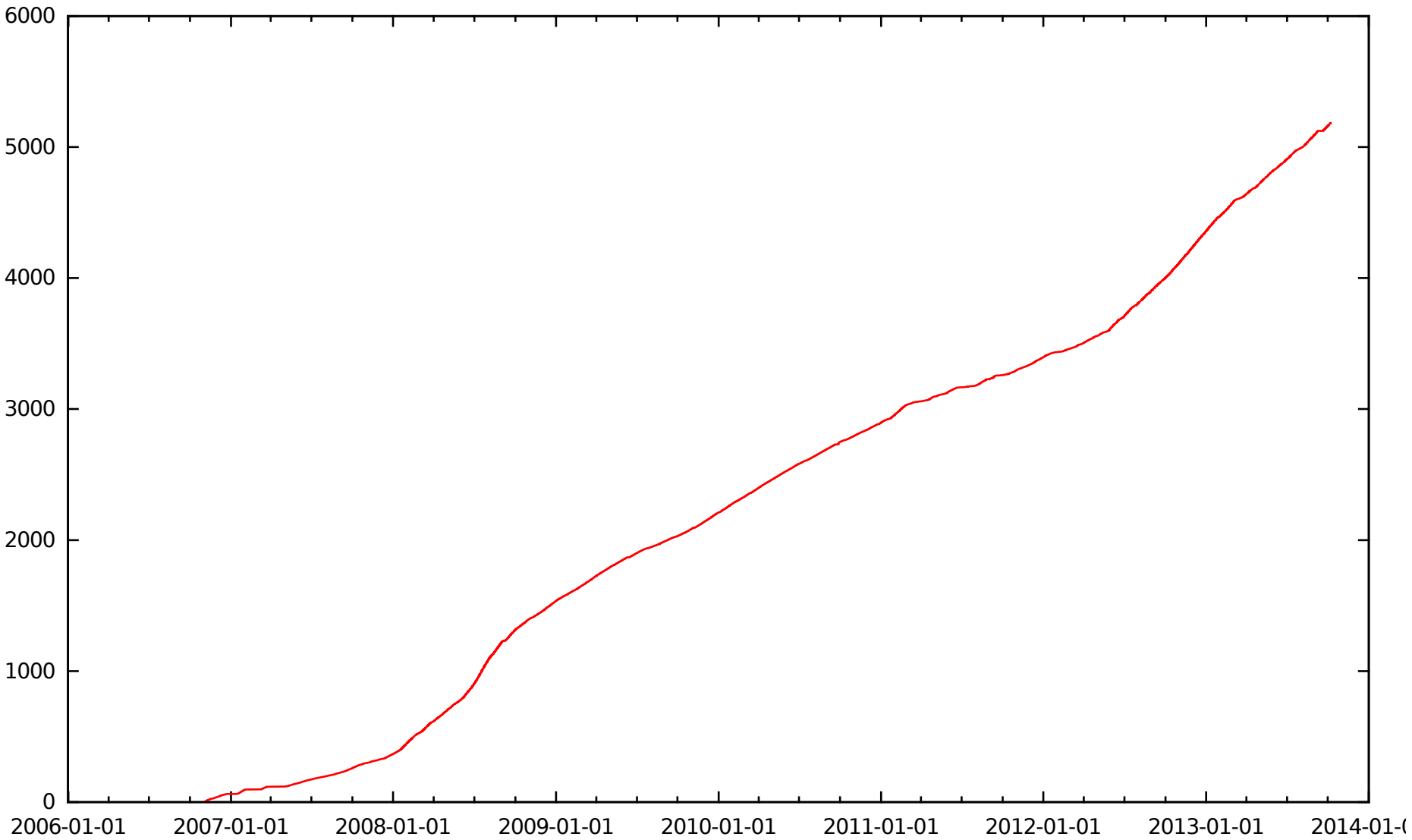
Function pown for $3 \leq n \leq 5188$:

- Rounding to nearest:
  `pown(1.5C69202D46821, 952) = 1.3B993E08AAD26P+423:10[63]1001`
  `pown(1.C72CE7406B3CE,1776) = 1.7D646B1EE4F67P+1474:01[64]0011`
- Directed rounding:
  `pown(1.290EB7BCC6A0E,4025) = 1.B10D94BB8FD98P+863:11[63]0000`

Function rootn (denoted rtn) for $3 \leq n \leq 5188$:

- Rounding to nearest:
  `rtn(1.DCBA0C48B3F29P+253, 1039) = 1.2F4027B25ACDF:01[73]0100`
  `rtn(1.AC171E04B83E0P+137, 1907) = 1.0D24A15B3F0AF:10[73]1101`
- Directed rounding:
  `rtn(1.C3EC89C7763F1P+1493,2309) = 1.90DC35E30BD19:11[73]0001`
  `rtn(1.7587F927AFFD8P+2911,3592) = 1.C0FF5FB7FB24E:00[74]1101`
  `rtn(1.BE49DF2392B0FP+2117,3712) = 1.7C2D624C9A5C5:00[74]1111`
  `rtn(1.B10D94BB8FD99P+863, 4025) = 1.290EB7BCC6A0E:00[75]1011`

Progression of HR-Case Results for x^n



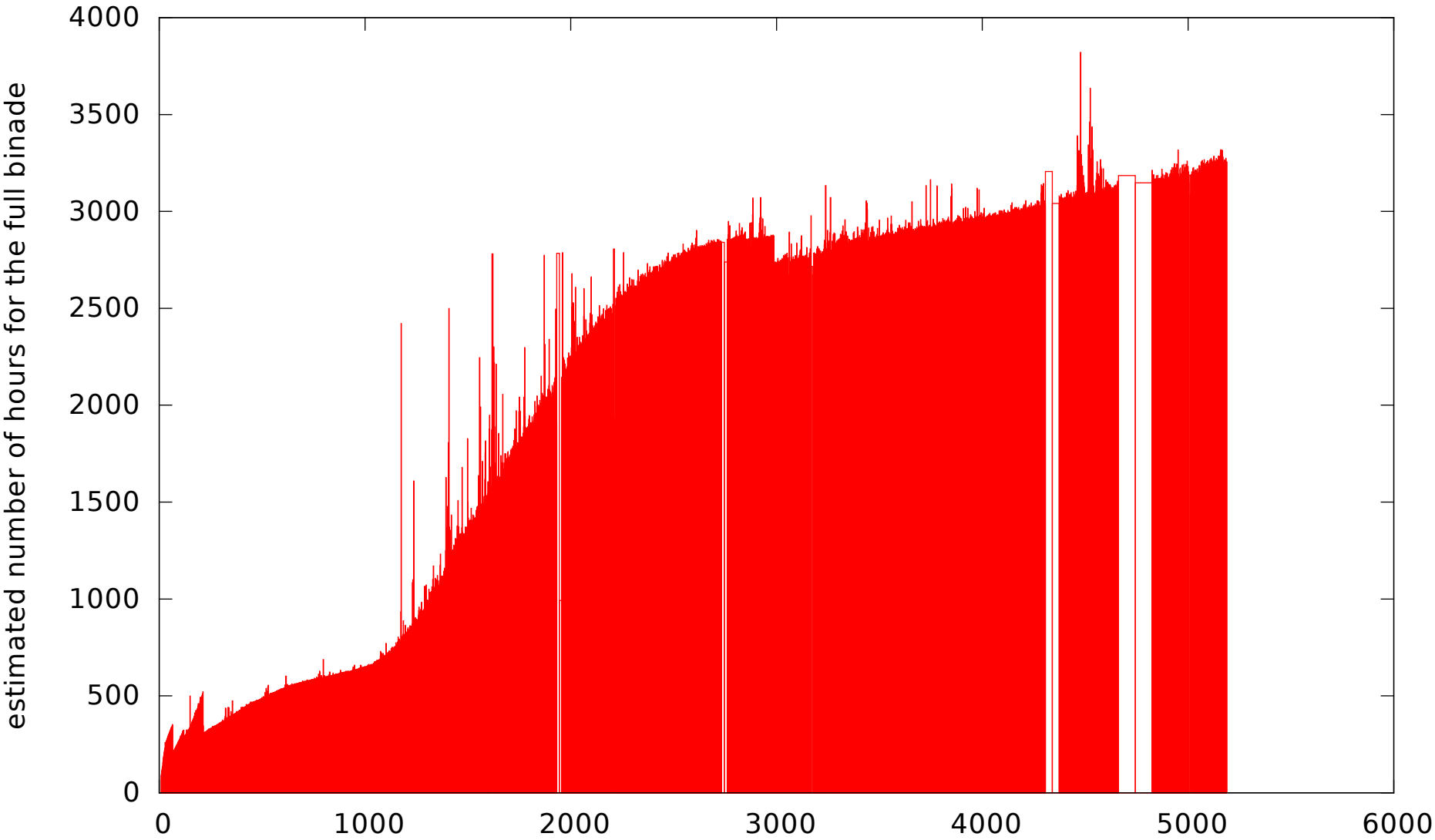
Average Computation Time

The following slides give graphs showing an estimate of the average computation time of each tested x^n (the full binade) for some machines.

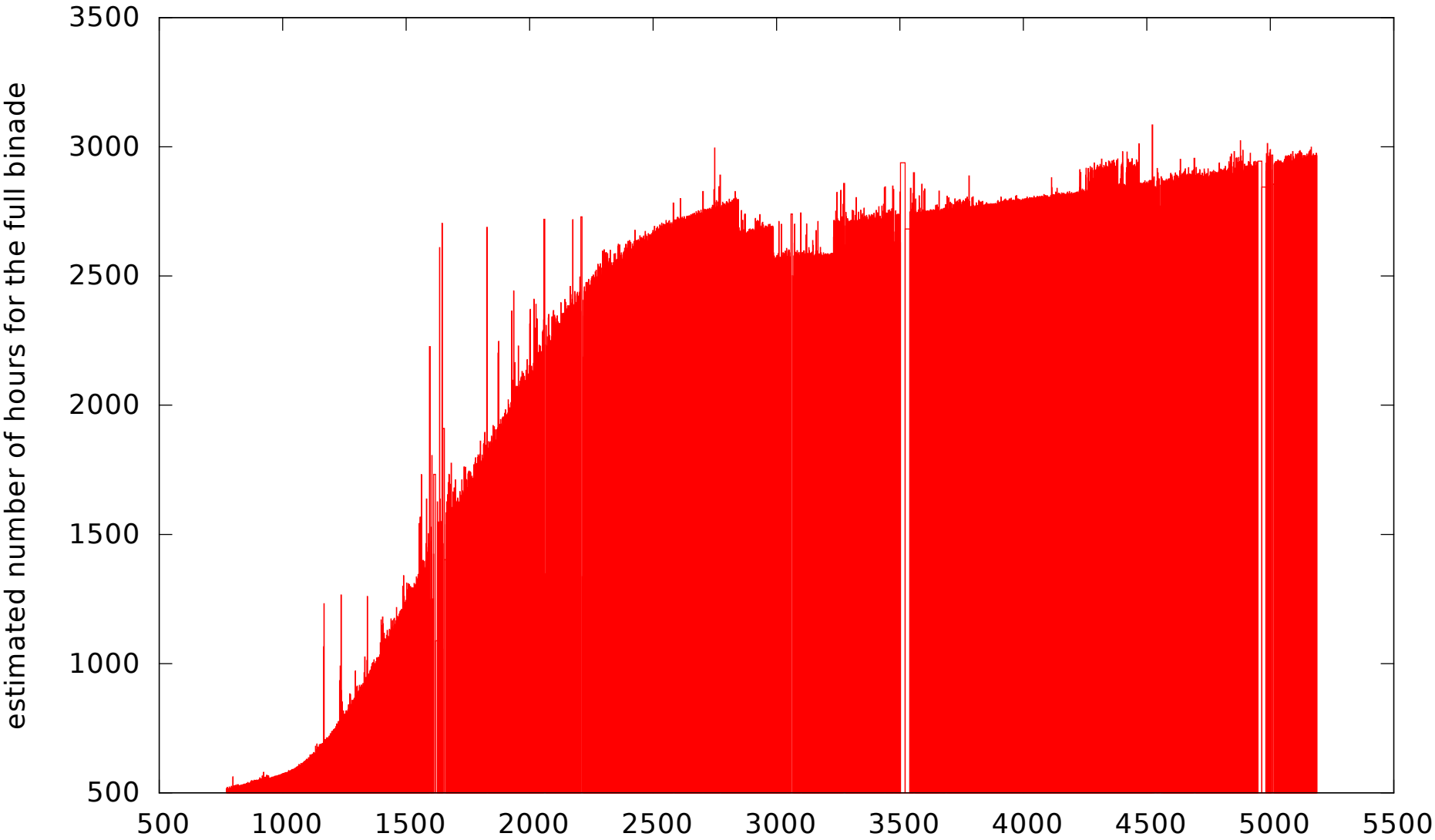
Based on the actual tests of x^n (see previous slides), assuming that the total time is proportional to the time taken by the interval chunks that have been tested on the machine.

Some timings are surprising... Some of them can be explained, e.g. a change of the parameters of the algorithms before $n = 500$ (?).

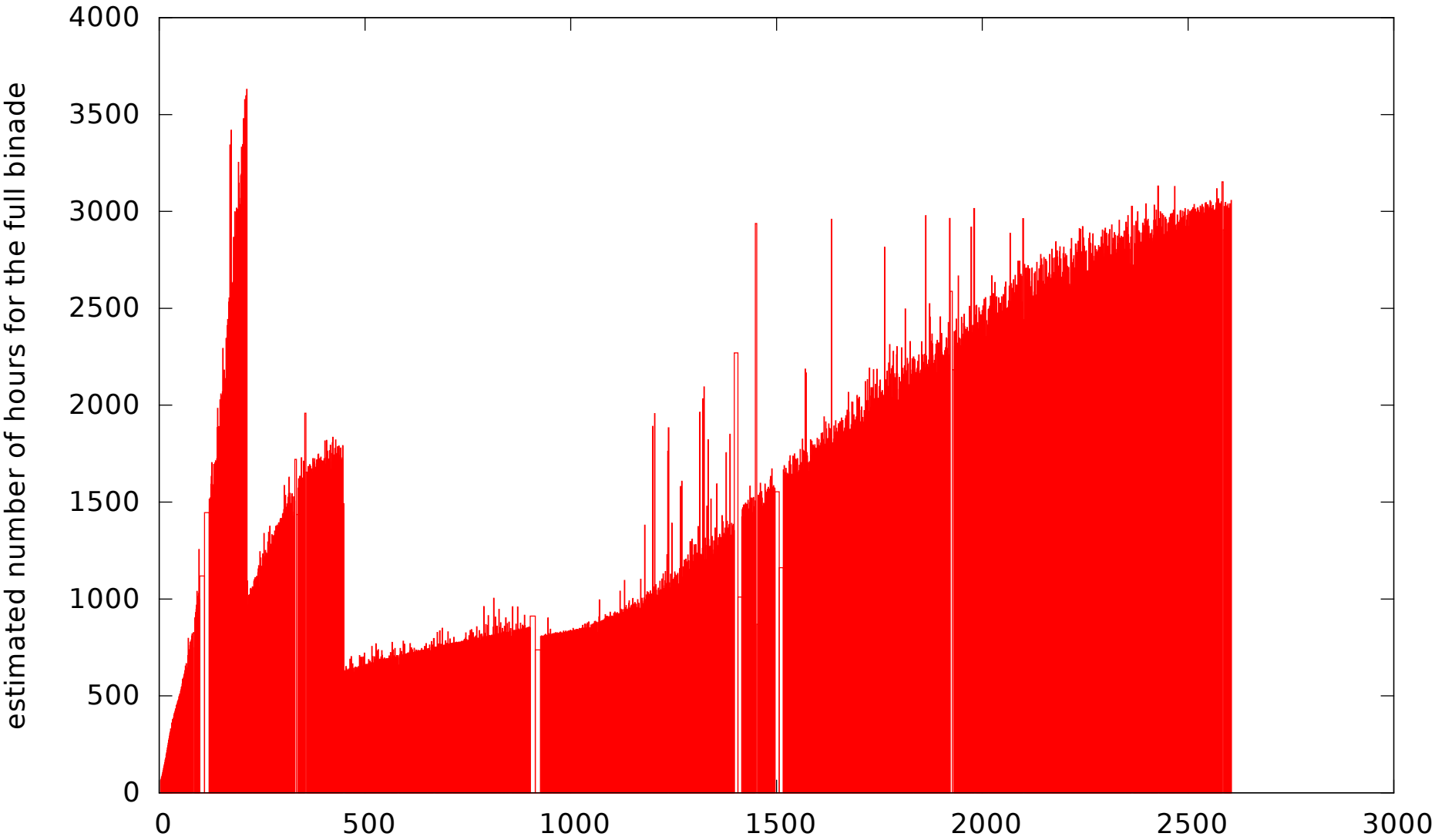
Host courge


estimated number of hours for the full binade
0
500
1000
1500
2000
2500
3000
3500
4000
0
1000
2000
3000
4000
5000
6000

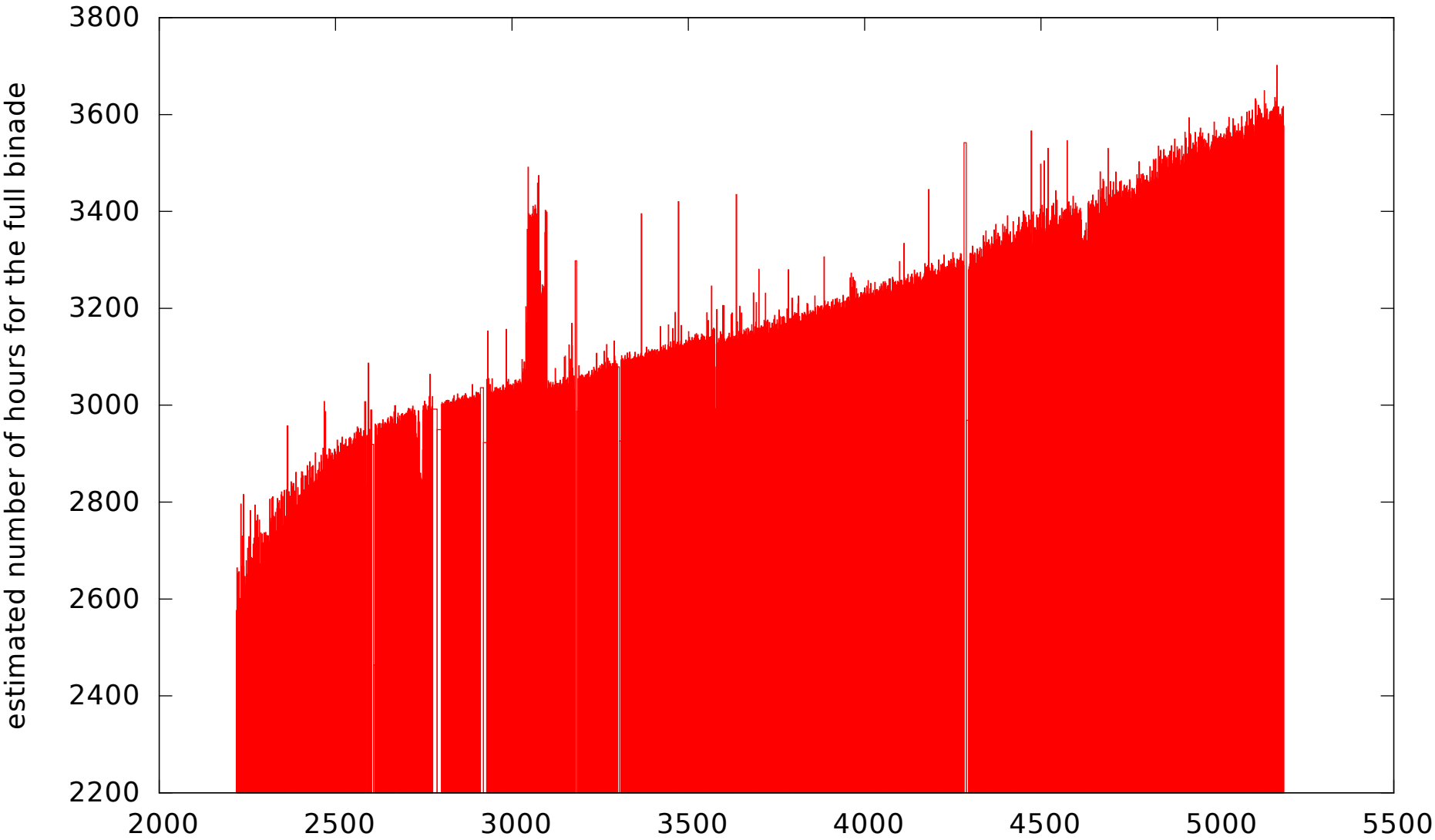
Host patate


estimated number of hours for the full binade
3500
3000
2500
2000
1500
1000
500
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
5500

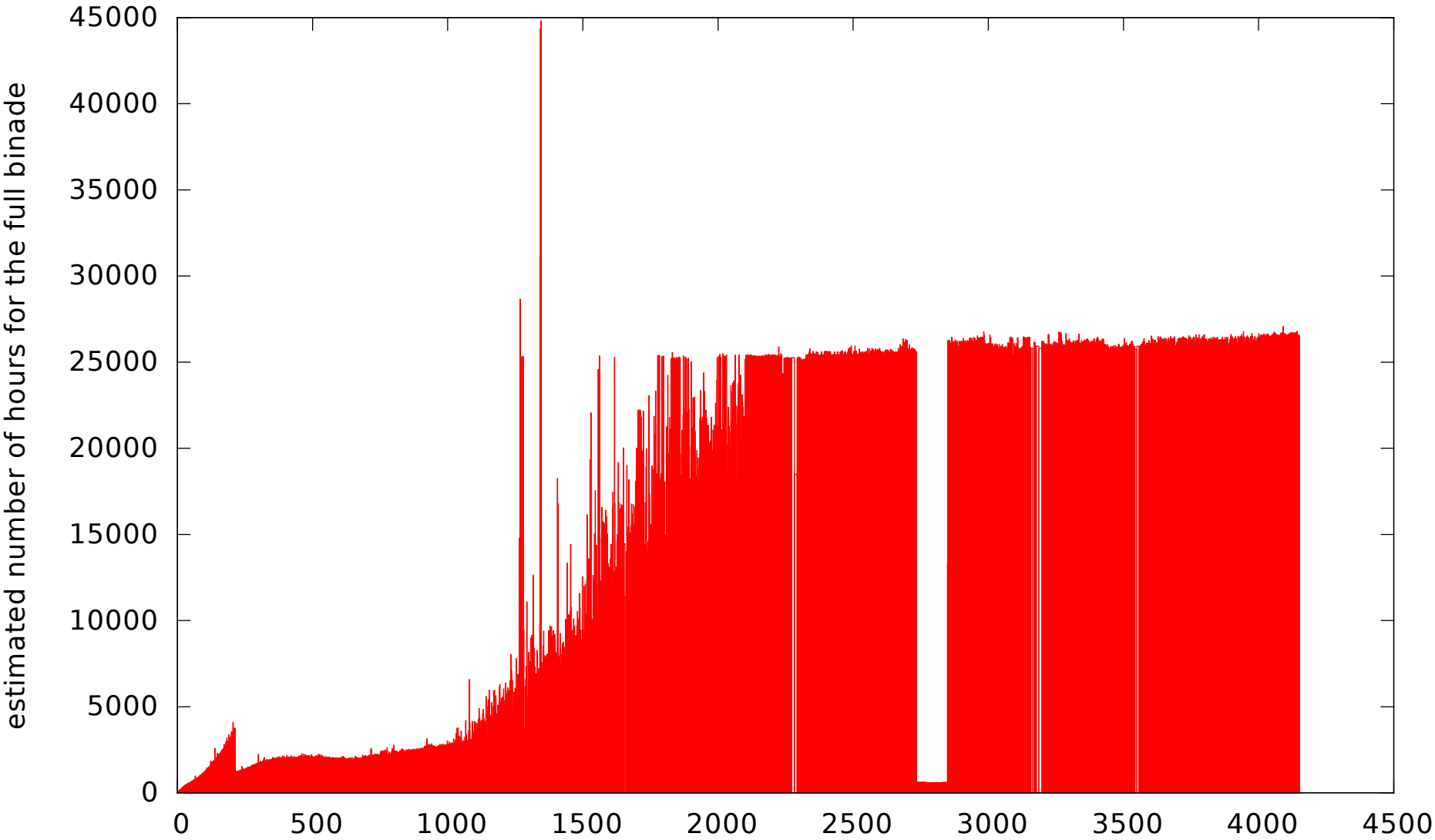
Host vin


estimated number of hours for the full binade
0
500
1000
1500
2000
2500
3000
3500
4000
0
500
1000
1500
2000
2500
3000

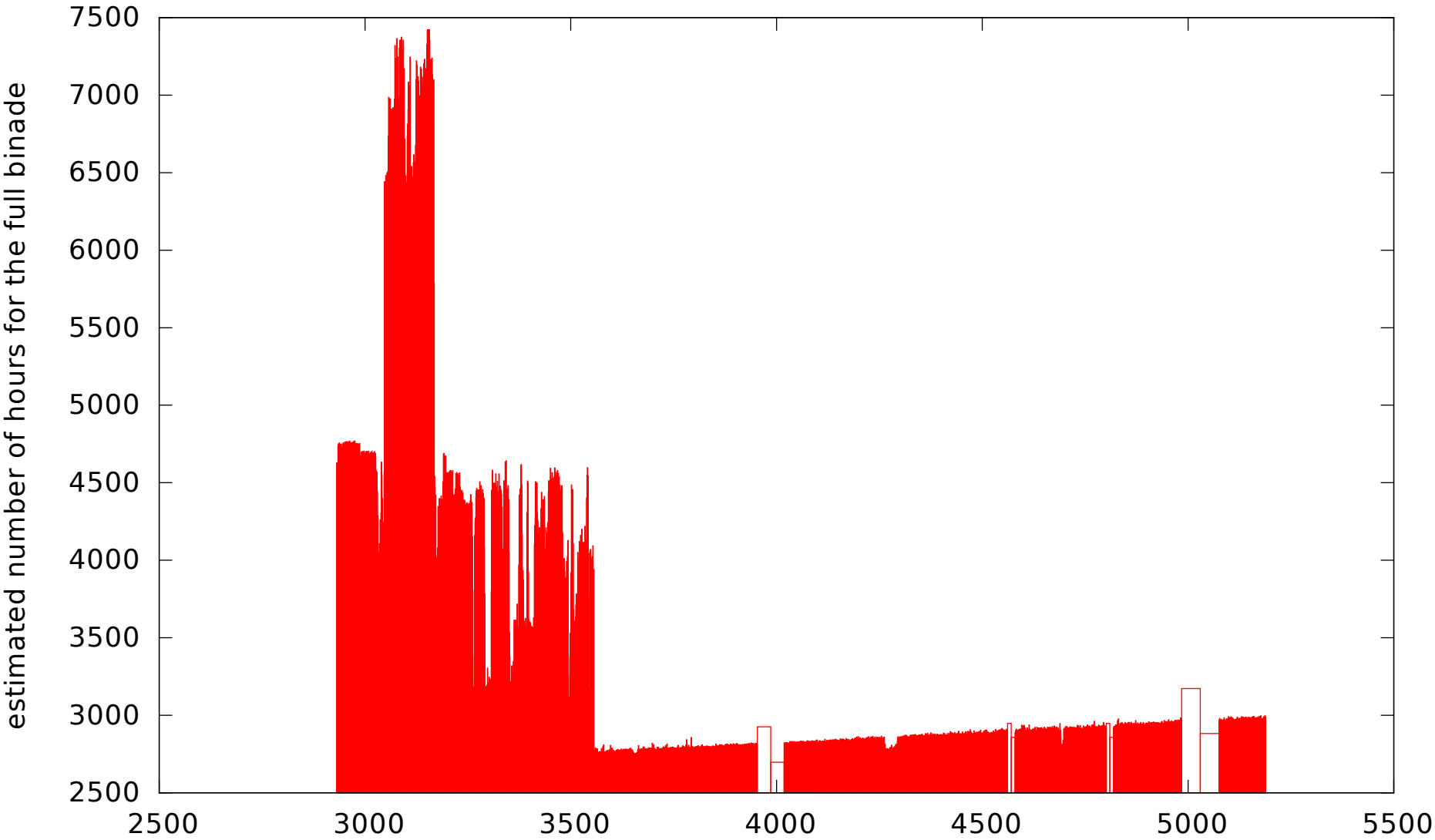
Host ypig


estimated number of hours for the full binade
3800
3600
3400
3200
3000
2800
2600
2400
2200
2000
2500
3000
3500
4000
4500
5000
5500

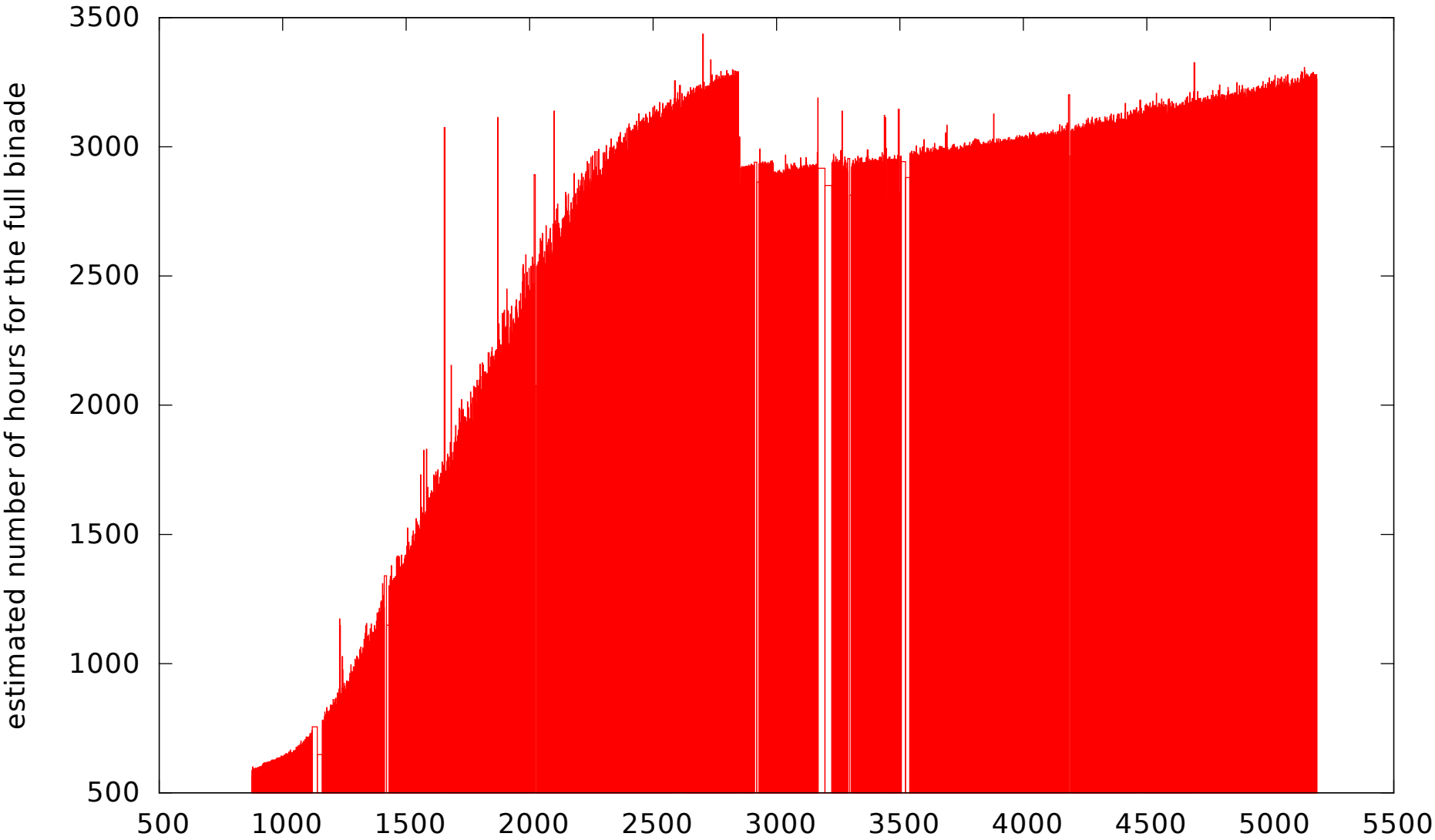
Host ble


estimated number of hours for the full binade
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
0
500
1000
1500
2000
2500
3000
3500
4000
4500

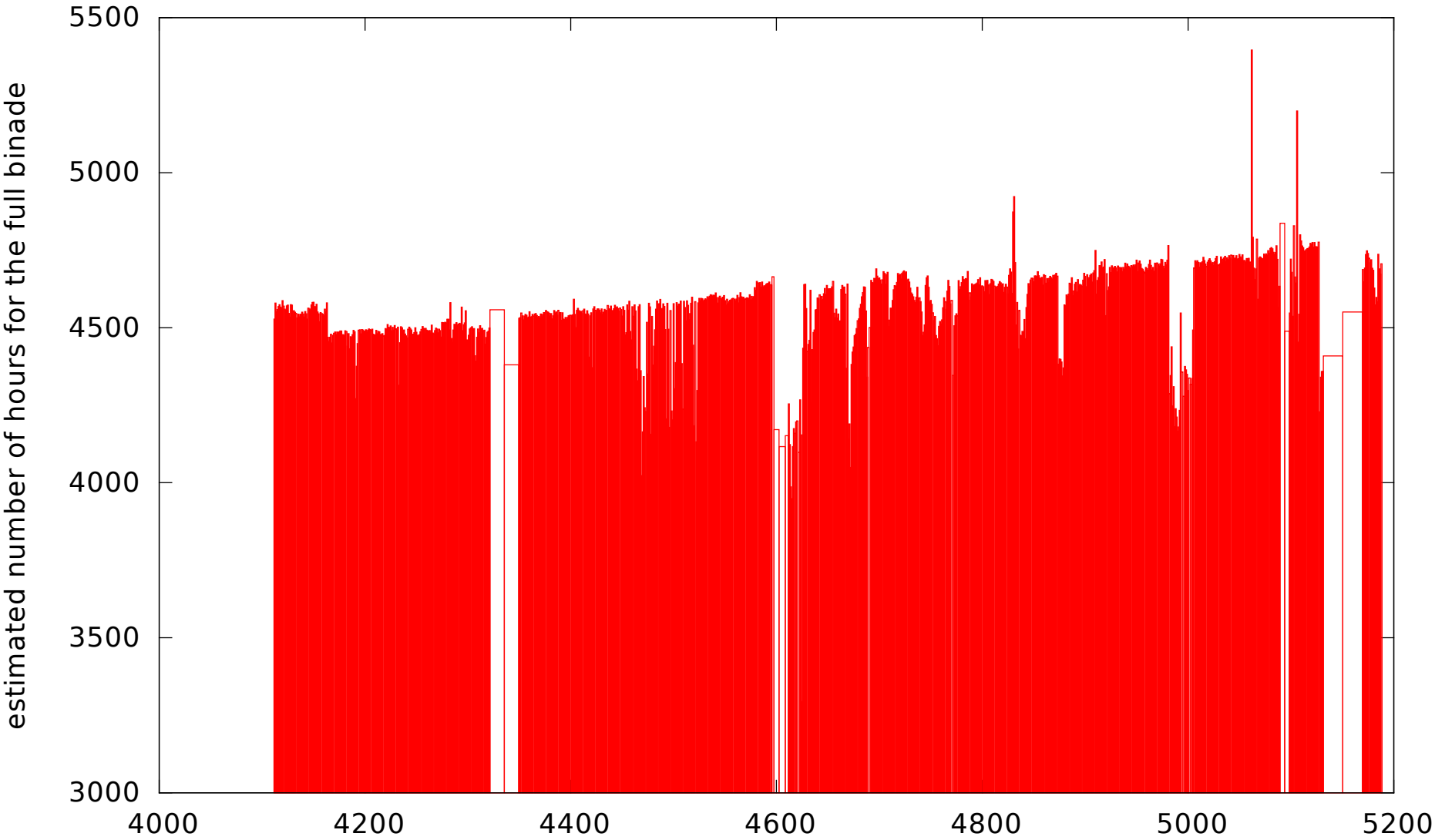
Host claustar



Host acalou


estimated number of hours for the full binade
3500
3000
2500
2000
1500
1000
500
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
5500

Host cassis


estimated number of hours for the full binade
5500
5000
4500
4000
3500
3000
4000
4200
4400
4600
4800
5000
5200